iKonvert NMEA2000 Gateway

Developer's Guide

Document Version: 1.03

October 3rd, 2018

Approvals:

Signature: _____ Date: _____

Paul Sumpner of Digital Yacht

Document Revision History

Date	Version	Description of Changes	Author(s)
03/10/2018	1.00	First Release	Paul Sumpner
20/10/2018	1.01	First round of feedback from developers	Paul Sumpner
24/10/2018	1.02	Review of Document against V1.0.0 Release Candidate	Paul Sumpner
12/12/2018	1.03	TX PGN Sentence now encoded in Base64	Paul Sumpner

Table of Contents

(Right-click and update field to refresh Table of Contents)

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Overview 4

2. Hardware Information 4

2.1 PCB Information 4

2.2 General Information 5

2.3 LED Information 6

3. Operation 6

3.1 Power Up 6

3.2 Normal Operation 7

3.3 Application Closes or Restarts 7

4. iKonvert Serial Data Protocol 8

5. Gateway Test Tool Reference Application 12

1. Introduction

1.1 Purpose

This document provides a full Developer's Guide to supporting the iKonvert NMEA2000 Gateway in a Desktop type application or as part of an embedded hardware design.

1.2 Scope

The scope of this document is to give a developer all hardware and software information they need to integrate and support the iKonvert Gateway in their applications or products.

1.3 Overview

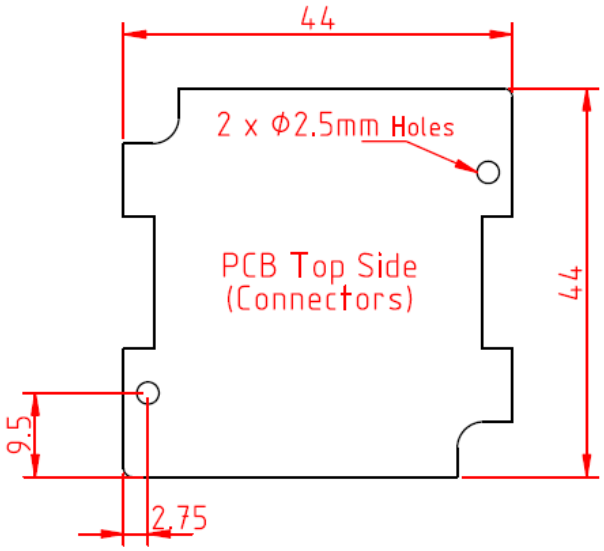
The iKonvert NMEA2000 Gateway is designed to allow an Application Developer or Equipment Manufacturer to support bi-directional communication with an NMEA2000 network, through simple serial messages. Any individual or company that wishes to integrate iKonvert will need as a minimum the NMEA2000 Appendix A+B which is available from http://nmea.org for both NMEA members and non-members.

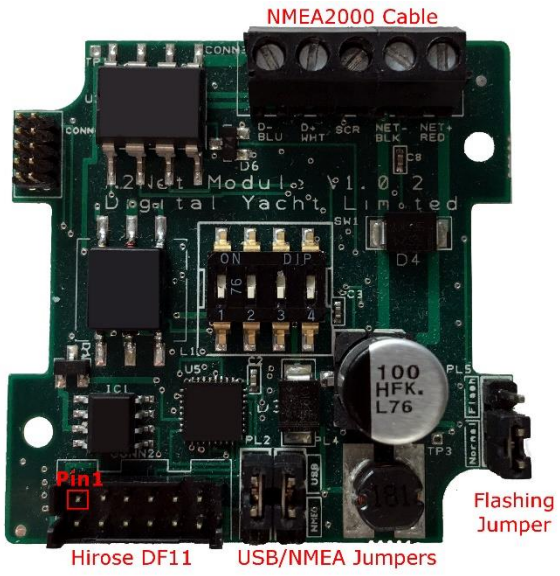
No special libraries are required to integrate iKonvert and the USB version uses an FTDI chipset that has good driver support for all common operating systems.

2. Hardware Information

2.1 PCB Information

The iKonvert Gateway is available in two versions; a USB version or ISO (NMEA0183) version. The only difference between the two versions are the cable and the position of two "jumpers" on the PCB.





The small PCB has been designed for easy integration in to other products with two 2.5mm fixing holes. It features a 5 way terminal block for connection to an NMEA2000 cable and an Hirose DF11 connector for IO and USB.

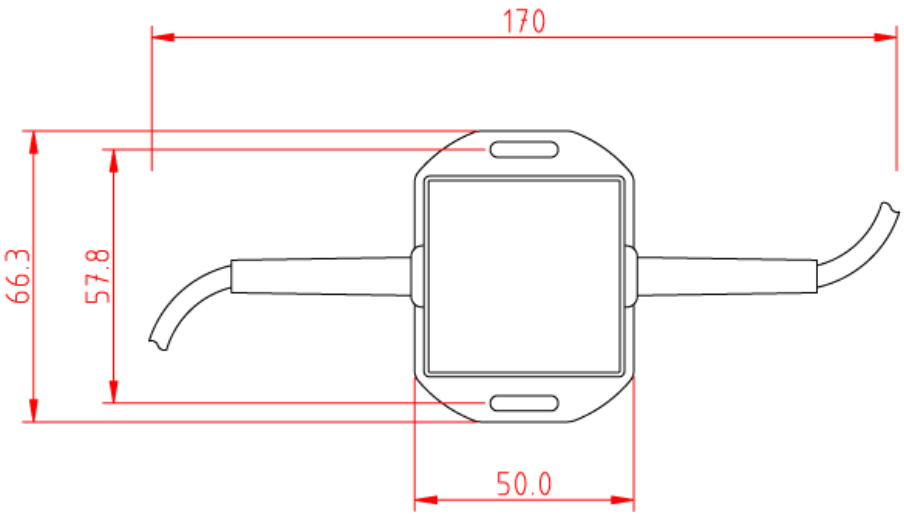
Customisation/re-branding is very easy, just a simple silver product label which we can provide the artwork of in Illustrator or EPS format. If you wish to have your own brand on the product label, we simply ask you to pay for a batch of product labels which you then free-issue to us for inclusion on your units.

Our standard NMEA2000 drop cable (Male), USB cable or ISO Data cable can be supplied or for customers that wish to use their own cable, the DF11 connector (DF11-12DP-2DSA) pin out is shown.

PL1 pin	Function
1	Serial TX+
2	Serial RX+
3	Serial TX –
4	Analogue Input
5	USB GND
6	Digital Output (OC)
7	USP DP
8	NC
9	USB DM
10	Serial RX-
11	USB VBUS
12	NC

2.2 General Information

The enclosure can be mounted using its two slotted 4mm wide fixing holes (see dimensions below). Use suitable screws/bolts (not supplied) to fix the converter to a flat surface or cable tie it to a cable loom or other solid object. Note - the unit may be installed in any orientation.



To access the PCB, simply unscrew the two cross-head self-tapping screws on the rear of the enclosure. There is a 4 way DIP switch in the center of the PCB that will be used in the future for mode selection. For now the gateway has just one mode, which we refer to as the “RAW Data” mode, the protocol of which is covered in this document. **In “RAW Data” mode the gateway’s baud rate is 230400**.

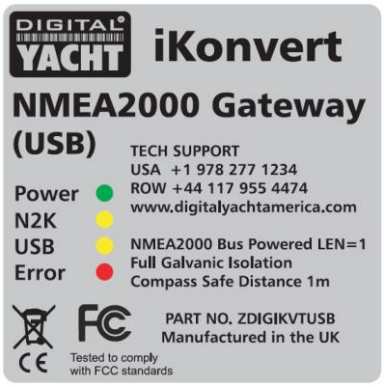
The gateway has one Analogue Sensor Input (Red Wire) and one “Open Collector” digital output (Black Wire). These are currently not utilized and only accessible using the ISO Data cable, but will be enabled in the future.

A 5-way terminal block is used for connecting the NMEA2000 drop cable. This means that longer drop cables can easily be fitted or even a proprietary NMEA2000 cable like a SeaTalkNG cable from Raymarine – simply cut one end off and fit the bare wires.

The NMEA2000 interface is fully isolated (power and data) and the board takes its power from the NMEA2000 network (LEN=1).

The USB interface is powered from the USB connection and will register as a virtual COM port on the computer it is connected to, even if the gateway is not powered up from the NMEA2000 network.

2.3 LED Information



The gateway has four indicator LEDs that show the status of the gateway.

When the unit powers up the Power LED will illuminate. If the gateway is initialised and on the NMEA2000 network, then the N2K LED will flash as N2K data is received and the USB LED will flash as application data is received from the USB device.

The Error LED will flash whenever error conditions occur such as; the application sends wrong commands to the gateway or if frame errors occur on the NMEA2000 networks.

3. Operation

The following description of operation, is only applicable when the gateway is in “RAW data” mode, future new modes will have different operation.

3.1 Power Up

When the gateway powers up for the first time, or after a factory reset, it will send out a Boot Loader text string….

$PDGY,TEXT,Digital_Yacht_iKonvert_v1.00_#AB-181018-12345_Address:XXX CR LF

Which identifies the model, firmware version, serial number and last CAN address.

Then it will wait for the application to configure and initialise it. At this stage even though the gateway is physically connected to the NMEA2000 network, it is “Off Bus” (inactive) and no NMEA2000 communication is taking place.

To tell the application that it is powered up but “Off Bus”, the gateway outputs an “I’m Alive” message every second, which is the Network Status Message (see 4.3) but with null data values…

$PDGY,000000,,,,,,, CR LF

By default just the mandatory TX and RX PGNs are enabled and the first task for the application is to send the required TX and RX PGN lists to the gateway (see 4.5 and 4.6), which will store them in Non-Volatile memory.

The TX and RX PGN lists are retained by the gateway after a power cycle, so it is not necessary for the application to set these every time it is run.

Before any NMEA2000 communication can take place, the gateway must be initialised, which causes it to go “On Bus” i.e. negotiate a CAN Address.

The gateway is designed to operate as autonomously from the application as possible and once initialised it will take carry out all the normal NMEA2000 network management tasks automatically. Negotiating a CAN Address, maintaining a list of active devices on the network, responding to requests from other devices, etc. are all handled with no input from the application.

3.2 Normal Operation

Once the TX and RX PGN lists are configured, the application must initialise the gateway by sending it the Initialisation Sentence (see 4.7). This causes the gateway to go “On Bus” and the N2K LED will now illuminate and flash as data is received.

All PGNs received that are in the gateway’s currently configured RX PGN list will be passed through to the application using the RX PGN Sentence (see 4.1). The PGN number, priority, source and destination Addresses, time stamp and the PGN payload (encoded in base64) are all included in the sentence, so that the application has full visibility of what the data is, where it has come from, it’s priority and relative timing.

Every second, the gateway will also send a Network Status Sentence that provides the application with additional information about the status of the NMEA2000 network (see 4.3).

If the application wishes to transmit a PGN on to the network, it must send a TX PGN sentence (see 4.2) to the gateway. The gateway will check that the requested PGN to be transmitted is in its TX PGN List, and if it is, it will place the PGN in its transmit queue and send at the first opportunity. If the PGN is not in the TX PGN list or the gateway is unable to complete this transmission it will send a NACK sentence back to the application with the reason for non-transmission (see 4.9).

3.3 Application Closes or Restarts

If the application closes for any reason and communication with the iKonvert gateway is lost, the gateway will continue to stay in whatever state it was in before i.e. “On Bus” or Off Bus”.

If it is “On Bus” then it continues to send out the Heartbeat PGN 126993 and responds to any network management PGNs.

When the application runs again and opens the virtual serial port connection to the iKonvert gateway, it will immediately start to receive the Network Status Sentence from the gateway, with null data if “Off Bus” or with the status of the NMEA2000 network if “On Bus”. Also, if the gateway is “On Bus” it will immediately start to send the application RX PGN Sentences, without the application having to re-initialise it.

4. iKonvert Serial Data Protocol

The gateway will be connected to a computer via USB and in order for the gateway to communicate with an application running on that computer, a number of Proprietary NMEA0183 sentences have been defined.

Sentences that only contain ASCII characters start with a **$** symbol and sentences that include the binary PGN data start with a **!** symbol.

To reduce serial data bandwidth, all received binary PGN data is encoded in base64, whilst all PGN data to be sent to the gateway by the app must be in HEX.

All sentences sent over USB **do not have** an NMEA0183 checksum, as the USB interface has its own CRC data integrity checks. In the future a Checksum ON/OFF setting will be implemented to allow ISO versions of the gateway to output a checksum.

4.1 RX PGN Sentence

For each RX PGN received from the NMEA2000 network, the gateway will generate a mixed ASCII and Base64 (PGN payload) sentence in the following format…

!PDGY,<pgn#>,p,src,dst,timer,<pgn_data> CR LF

Key

<pgn#>	=	NMEA2000 PGN number between 0 and 999999
p	=	Priority 0-7 with 0 being highest and 7 lowest
src	=	Source Address of the device sending the PGN between 0-251
dst	=	Destination Address of the device receiving the PGN between 0-255 (255 = global)
timer	=	internal timer of the gateway in milliseconds 0-999999
<pgn_data>	=	The binary payload of the PGN encoded in Base64

4.2 TX PGN Sentence

For each TX PGN the application wishes to send on the NMEA2000 network, the app will send a mixed ASCII and Base64 (PGN payload) sentence in the following format…

!PDGY,<pgn#>,dst,<pgn_data> CR LF

Key

<pgn#>	=	NMEA2000 PGN number between 0 and 999999
dst	=	Destination Address of the device sending the PGN between 0-255
<pgn_data>	=	The binary payload of the PGN encoded in Base64

4.3 Network Status Sentence

To provide the application with useful data about the NMEA2000 network, the gateway generates the following ASCII sentence (at 1Hz)….

$PDGY,000000,29,12,7,12345678,12,31 CR LF

```
                ^  ^ ^    ^       ^   ^
                1  2 3    4       5   6
```

Key

Field 1	=	Bus Load between 0-99%
Field 2	=	Frame Errors (not currently supported)
Field 3	=	Number of Devices Live on network
Field 4	=	Network Uptime in seconds
Field 5	=	Gateway's CAN address
Field 6	=	Number of Rejected TX PGN Requests

NOTE - A PGN number 000000 is used to identify this special network status data, which under normal circumstances would never be seen on an NMEA2000 network

The number of devices on the network is useful to monitor for changes in the network status, although all applications would benefit from maintaining their own NMEA2000 Device List. A change in the number of devices can be the trigger for your application to query the devices on the network.

You will need the Gateway's CAN Address so that your application can know which addressed PGNs are being sent to it and ignore addressed PGNs with a different destination address.

"Rejected TX PGN Requests" counter increments by 1 each time the application tries to transmit a PGN at a rate faster than the default/recommended update rate as defined by the NMEA2000 Appendix B. For instance, if the application tries to send compass heading at 12Hz the count will increase by two every second as the default/recommended update rate is 10Hz.

The Network Status Sentence is output at 1Hz and the gateway always outputs this Status message even when not on the bus. In this "Off Bus" situation the gateway outputs the sentence below, as an "I'm Alive" type message so the application knows the gateway is powered up but not initialised…

$PDGY,000000,,,,,,, CR LF

4.4 Factory Reset Sentence

To reset the gateway to factory defaults, the app will send an ASCII sentence in the following format…

$PDGY,N2NET_RESET CR LF

A factory reset clears all settings, configurations and the TX/RX Lists.

4.5 Set RX PGN List Sentence

To set the RX PGN List that the gateway will store and use, the app will send an ASCII sentence in the following format…

$PDGY,RX_LIST,xxxxxx,xxxxxx,xxxxxx,xxxxxx…….xxxxxx CR LF

4.6 Set TX PGN List Sentence

To set the TX PGN List that the gateway will store and use, the app will send an ASCII sentence in the following format…

$PDGY,TX_LIST,xxxxxx,xxxxxx,xxxxxx,xxxxxx…….xxxxxx CR LF

4.7 Initialisation Sentence

To Initialise the gateway (must be done before gateway goes on bus), the app will send an ASCII sentence in the following format…

$PDGY,N2NET_INIT,<mode> CR LF

Initialises the gateway and enables it to go on to the NMEA2000 network. Any setup/configuration of the Gateway, including TX and RX PGN lists must be done before the initialisation sentence is sent.

If the gateway's TX/RX PGN list is blank (default) just the mandatory TX and RX PGNs are enabled but if the TX/RX lists have PGNs in them, the gateway will use these lists when queried by another network device.

The <mode> field is used to tell the gateway to go in to a particular operating mode. Currently only two modes are defined "NORMAL" and "ALL".

4.8 Offline Sentence

To make the gateway go "Off Bus" so that the application can change its settings, the app will send an ASCII sentence in the following format…

$PDGY,N2NET_OFFLINE CR LF

4.9 Gateway Setup Sentence

To setup the gateway with any required configuration settings, the app will send an ASCII sentence in the following format…

$PDGY,N2NET_SETUP,instance,address,baud,checksum, tbd,tbd CR LF

Key		
Instance	=	Integer 0-9
Address	=	Integer 001-255
Baud	=	Integer 4800-230400
Checksum	=	A = enabled V = disabled and default = disabled

4.10 ACK Sentence

To tell the application that the last received sentence was successfully actioned, the N2Net gateway will send an ASCII sentence in the following format…

$PDGY,ACK,message CR LF

Where "message" is the command that was successfully actioned. Below is a list of currently generated ACK messages;

$PDGY,ACK,N2NET_RESET

$PDGY,ACK,RX_LIST

$PDGY,ACK,TX_LIST

$PDGY,ACK,N2NET_INIT,ALL

$PDGY,ACK,N2NET_INIT,NORMAL

$PDGY,ACK,PROD_CONFIG_SAVED

$PDGY,ACK,<pgn#>,<type>,<length>,<rate>

The last message is the one generated when the app sends a TX PGN command to the gateway. The ACK message repeats the PGN Number of the PGN sent, the type of transmission SF = Single Frame and FP = Fast Packet, length of PGN data payload in bytes and the Update Rate of the PGN in milliseconds i.e. time since App last sent this PGN.

4.11 NAK Sentence

To tell the application that the last received sentence was not actioned, the N2Net gateway will send an ASCII sentence in the following format…

$PDGY,NAK,error#,error_text CR LF

The NAK sentence includes a numeric Error Code and the associated Error Text that the application will display i.e. “Error 2 PGN not in TX List”. The currently generated NAK messages are;

$PDGY,NAK,1,ALREADY_INITIALISED

$PDGY,NAK,2,PGN_NOT_IN_TX_LIST

$PDGY,NAK,3,NOT_INITIALISED_YET

$PDGY,NAK,4,SERIAL_NUMBER_NOT_PROGRAMMED

$PDGY,NAK,5,SERIAL_NUMBER_IS_INCORRECT

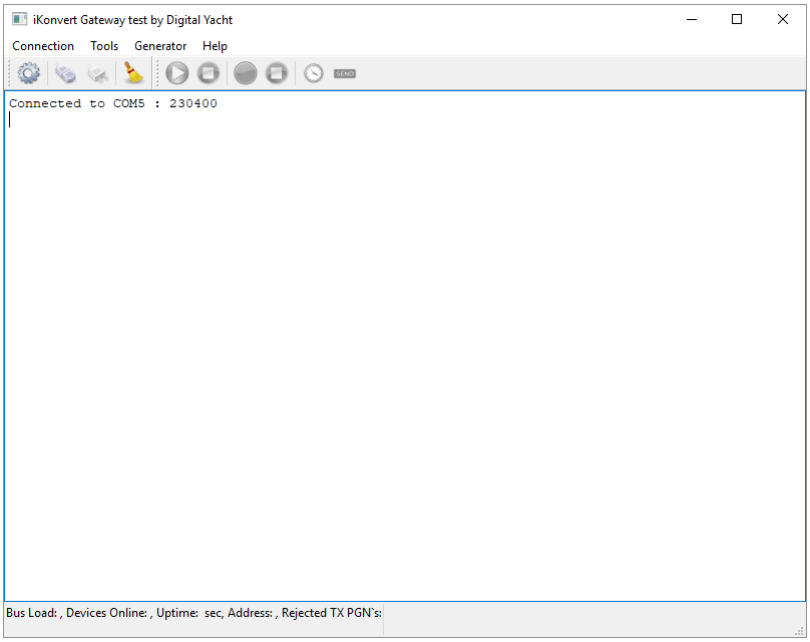
The last two NAK messages are only generated in response to special commands used during production and should never be seen in normal operation.

If the application tries to send a PGN at an update rate faster than the rate recommended in the NMEA2000 specification, the gateway will not NAK the command, but it will reject that PGN transmission and increment its PGN rejection count by one. It is recommended that during app development, developers check the Network Status Message (see section 4.3) and ensure that their app is not creating any rejected PGNs.

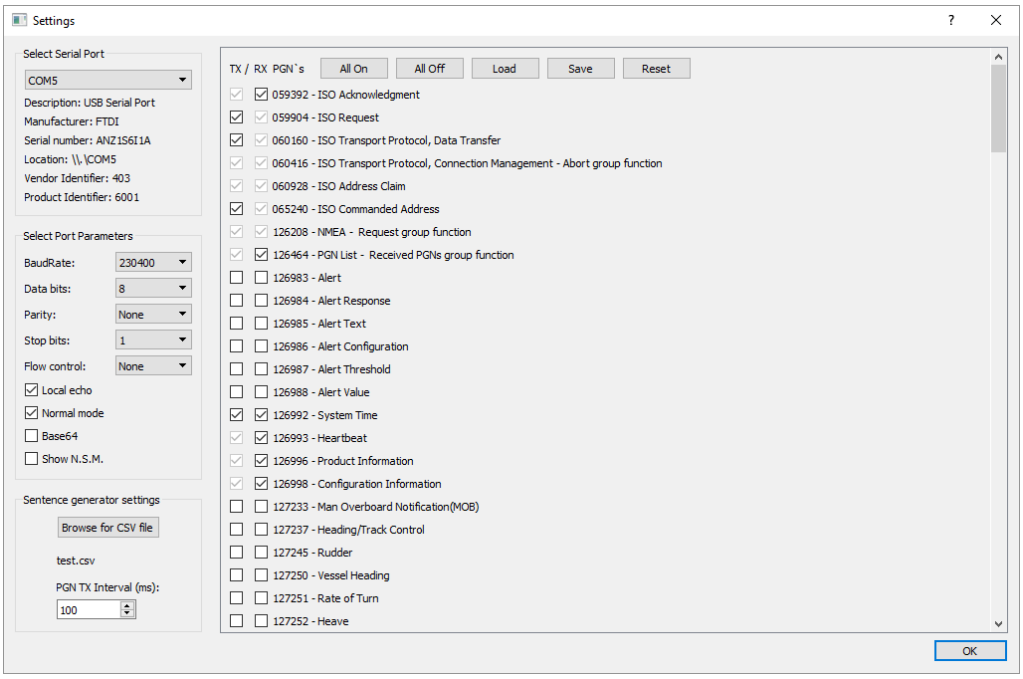
5. Gateway Test Tool Reference Application

To help developers support the iKonvert gateway in the most efficient and cost effective way possible, Digital Yacht have created a QT based Application that demonstrates and provides source code examples of how different commands can be used to control and read the gateway.

The QT App does not need installing just run the terminal.exe file and you will see this window…



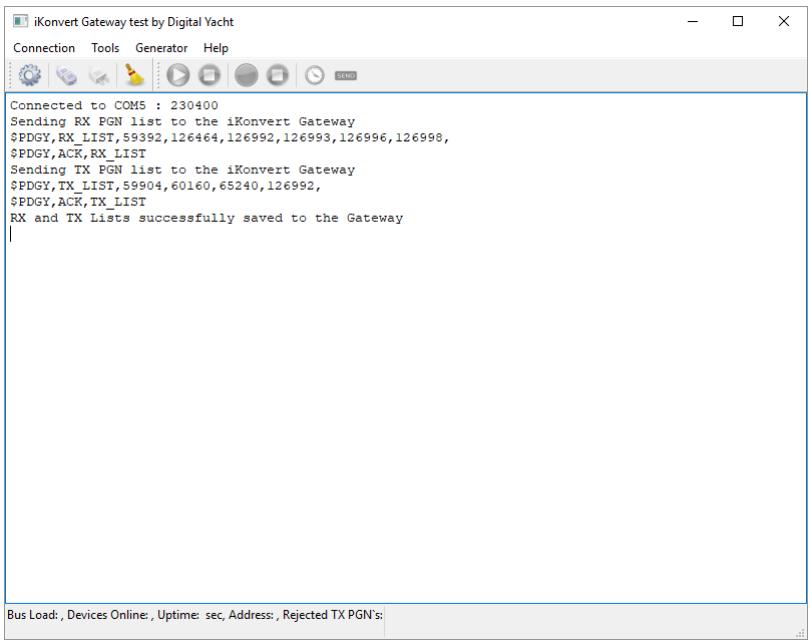
If the gateway is the only USB Serial Port device connected then it will automatically select it and set the baud rate to 230400. If it shows a different COM port, then click the Settings cog-wheel and this window will open…



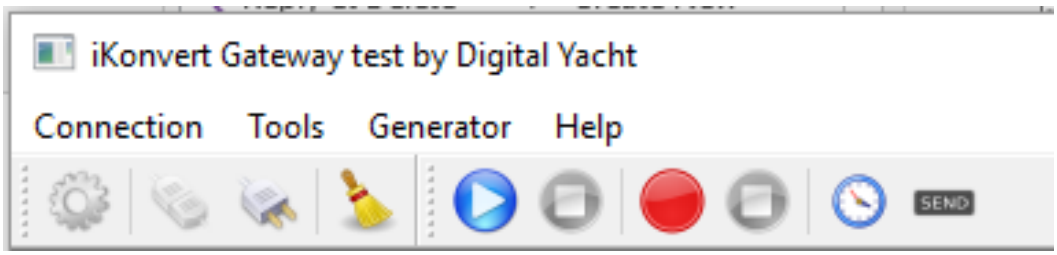
In the settings window you can select…

- The COM port, baud rate and other serial port settings
- “Local echo” mode so that commands you send are displayed in the terminal
- “Normal” initialisation mode or if un-ticked “All” mode will be selected
- Base64 encoding display (default is HEX for ease of reading)
- Whether the Network Status Messages are displayed (can create too much clutter)
- The CSV log file you want to replay and the PGN TX rate it replays at
- Which RX and TX PGNs are included in the gateway’s TX/RX Lists
    - All ON - ticks all PGNs
    - All OFF - unticks all PGNs
    - Load - displays the current lists stored in the gateway
    - Save - updates the gateway’s TX/RX Lists with the currently display PGNs
    - Reset – does a factory reset

Once you have configured the settings you want, click the OK button and the commands will be sent to the gateway and you will be able to see them in the main Terminal Window….

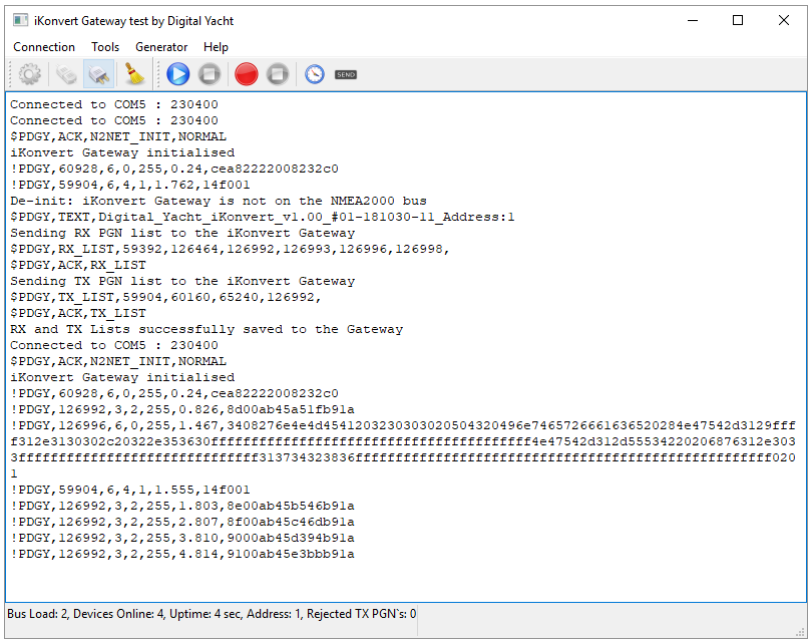


From Left to Right, the icons are;



Settings – Connect – Disconnect – Clear – Play – Stop – Record – Stop – Time – Send

Now you can initialise the gateway and make it go “On Bus” by clicking the **Connect** icon and you will see lots of RX PGN Sentences appear, along with the Network Status Messages (N.S.M), if you have set these to display in the Settings…



You can make the gateway go “Off Bus” by clicking the **Disconnect** button and you can clear the terminal window by clicking the **Clear** icon.

You can record all of the received PGNs to a CSV file for future playback by clicking the **Record** button and then stop the recording by clicking the **Stop** button to the right of the **Record** button.

You can playback the CSV file by selecting the file in the Settings window and also the TX rate in milliseconds and then clicking the **Play** button. To stop playback click the **Stop** button to the right of the **Play** button.

You can transmit a System Time PGN (126992) by clicking the **Time** button – it takes the date and time from the computers real time clock.

You can send any TX PGN that you construct and copy to the clipboard by clicking the **Send** button.

The QT Test Tool App has been compiled for Windows and the executable will run on any Win7/8/10 PC. Full source code is provided if you wish to compile the QT app for LINUX or Mac.