Accessing the N6GN Reference, Frequency Extender and Transverters

The N6GN WiFi-enabled boards have three ways for the user to access them. While to some degree they are set-and-forget with most operation continuing unaided and unsupervised, particularly during initial installation it will be necessary to ‘localize’ and possibly customize settings. These operations require access as does updating the firmware, as described in another document.

There are three ways that these units may be accessed

1. By a USB cable directly to the IoT33
2. By way of a web browser connecting to the web server on the units
3. By way of a command line interface, e.g. ‘curl’ to the web server

The initial interface must be by way of (1) the USB serial port on the IoT33 controller which is mounted on a socket on the PCB. As described in the firmware update document, although it is possible to do this with only a serial USB cable between a host computer running a terminal emulator and the IoT33, there are advantages to installing the Arduino IDE first, as described in Step 1 of the FW updating document. Doing this not only provides the proper environment for uploading new .bin files but the IDE also provides a Serial Monitor which can be used to initially configure the WiFi access, particularly the local SSID and Password. Be aware that the Port connection from the IDE can be a little quirky such that after reset or power cycle it may be necessary to re-select the appropriate Port.

Once WiFi access has been configured and saved to NVRAM on the IoT33, web interface, either by way of a (2) web browser or a (3) command line becomes possible.

Upon powering the unit the USB port will provide an initial report of the startup process followed by a very simple user access to modify some of the default settings.

At that point, dimply sending an ‘Enter’ to the unit should produce a response that describes these options with something like the following:

Active State:
web page is http://0.0.0.0:8078
=================================================
01) SSID = changeToYourSSID
02) password = changeToYourPassword
03) Clock1 = 25000000
04) User Clock2 = 100000000
05) Discipline = GPS
06) autoPromoteGPS = false
07) cpi = 3
08) rN_div = 64
09) dacnom = 2640
10) Tnominal = 27.00
11) xtaltc = 0.00

============

Use 'A' to save Active State to Power-On state
Use 'D' to save Default State to Power-On state
Use 'Mxx' to modify a parameter e.g. 'M09' to modify dacnom
Use 'S' to show Active State

No Changes

[Note] Presently, only the parameters in **bold** are settable by the user

Following this, sending ‘M01’ and ‘M02’ will result in prompts for the user to enter the SSID and password of a local WiFi access point, respectively. These new settings are then written and saved to EEPROM storage on the IoT33 by sending an ‘A’ which should result in a verification

**Active State written to EEPROM/Power-On State**

so that they will be used upon the next reset or power-up of the unit. If the selected WiFi access point can be successfully connected and proves DHCP support, typing ‘A’, as before, will provide the IP address of the web page to be used by a local browser, e.g.

**Active State:**
web page is [http://10.0.0.246:8078](http://10.0.0.246:8078)

From this point on, typing the web page address provided into the address field of a local web browser such as Chrome, Firefox, Safari etc should provide user access to the unit’s web interface.

An initial home page might look something like this:
Particular details may depend upon the unit type, discipline and lock settings and other parameters.

It is also possible to access a unit without using a web browser through a command line utility such as ‘curl’. This can allow computer interaction to provide automated and remote control apart from a user. As an example:

```
curl --get --data 'Status=1' http://10.0.0.246:8078
```

returns the Status & Utility Page information.

As of the time of writing there are a number of key:value pairs that may be sent and responded to by a unit. To understand this it can be noted that selecting “Status & Utility Page” in the above example actually POSTs a command to the unit which can be seen in the browser’s address line:

```
http://10.0.0.246:8078/?Status=1
```

Here ‘Status’ is a key and ‘1’ is a value. This results in the unit returning the Status & Utility page, however possibilities for setting parameters exist as well. Here is a current list of recognized key:value pairs, not all of which may apply to any given unit type since control of frequency converting units is included.
Caution, these may change with firmware revision so don’t count on any of these always being present.

<table>
<thead>
<tr>
<th>KeyWord</th>
<th>Values</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>allClocks</td>
<td>0,1</td>
<td>enable all Los</td>
</tr>
<tr>
<td>autoPromoteGNSS</td>
<td>0,1</td>
<td>set to change unlocked condition to GNSS lock</td>
</tr>
<tr>
<td>d_2 mult</td>
<td>Float</td>
<td>change user Clock</td>
</tr>
<tr>
<td>CCurrent</td>
<td>0-7</td>
<td>ADF4001 charge pump current, see data adf4001 sheet</td>
</tr>
<tr>
<td>discipline</td>
<td>GNSS,Ext10,TCXO XO</td>
<td>set discipline type</td>
</tr>
<tr>
<td>ADF4001 mult</td>
<td>0-7</td>
<td>set disciplinary type</td>
</tr>
<tr>
<td>IFreq</td>
<td>Float</td>
<td>set Ftune to ham band, MHz, shifted to 20 MHz IF</td>
</tr>
<tr>
<td>sock_div</td>
<td>2**(1-7)</td>
<td>User Clock output divider</td>
</tr>
<tr>
<td>ph_div</td>
<td>Float</td>
<td>additional scaling of ADF4001 R&amp;N to change pfd comparison frequency</td>
</tr>
<tr>
<td>ph_vco</td>
<td>Float</td>
<td>set PllB (user clock) only</td>
</tr>
<tr>
<td>requestedBand</td>
<td>0-7</td>
<td>ADF4001 R&amp;N divider, see data adf4001 sheet</td>
</tr>
<tr>
<td>requestedUserClock</td>
<td>10,20,50,80,100,440,700,1290,2200,440,33910,231290</td>
<td>Set Ftune to ham band, MHz, shifted to 20 MHz IF</td>
</tr>
<tr>
<td>m0</td>
<td>0-7</td>
<td>ADF4001 R&amp;N divider, see data adf4001 sheet</td>
</tr>
<tr>
<td>SetIF3</td>
<td>0,1</td>
<td>Move LO3 Frequency up to Synth2 (can’t turn it off without turning off Synth2)</td>
</tr>
<tr>
<td>setPreamp</td>
<td>0,1</td>
<td>Aux Output Enable</td>
</tr>
<tr>
<td>setSynth1</td>
<td>0,1</td>
<td>Aux Output Enable</td>
</tr>
<tr>
<td>setSynth2</td>
<td>0,1</td>
<td>Sleep for 16 seconds</td>
</tr>
<tr>
<td>sleep</td>
<td>0,1</td>
<td>sets page to return</td>
</tr>
<tr>
<td>tx_en</td>
<td>0,1</td>
<td>enable upconverter [PTT controls]</td>
</tr>
</tbody>
</table>