Windfreak Technologies, LLC
SynthNV Serial Communications

Talking to the SynthNV unit is done through USB via a virtual serial / com port. The drivers supplied by WFT must be installed on your PC before communication can happen in Windows. Linux usually works without installing drivers. After plugging in the hardware the com port will need to be identified, then used for any subsequent communication.

The SynthNV comes with software that uses these commands to easily control and program the device through its intuitive GUI, but some advanced users may want to make their own software and/or run on a different OS than Windows. These commands can be tested through serial communications terminals such as this free program: https://www.compuphase.com/software_termite.htm for Windows and other possible solutions for Linux and OSX. Also, the SynthNV has been tested and works well with an Android based app from Google Play called "Android USB Serial".

The first character of any communication to the SynthNV unit is the command. (It is case sensitive.) What this character tells the unit to do is detailed below. Ideally a "package" is sent all at once. For example, a communication for programming the frequency of the LO to 1GHz would be sent as "f1000.0" (without the quotes).

For commands that return information from the SynthNV unit, such as reading the firmware version, it is advisable to send the command and then read the bytes returned fairly quickly to get them out of the USB buffer in your PC.

f) RF Frequency Now (MHz) 1000.0
o) set RF On(1) or Off(0) 1
h) set RF High(1) or Low(0) Power 1
a) set RF Power (0=mimimum, 63=maximum) 63
D) Read Power Detector A/D (0-1023 output)
w) Read RF power in dBm
V) Set A/D voltage reference 1.950
Q) Power Measurment Offset 83.500
v) show firmware version
e) write all settings to eeprom
x) set internal reference (external=0 / internal=1) 1
l) set lower frequency for sweep (MHz) 50.0
u) set upper frequency for sweep (Mhz) 4000.0
s) set step size for sweep (MHz) 50.0
t) set step time is 0.600 ms
g) run sweep (on=1 / off=0) 0
r) set reading while sweeping (on=1 / off=0) 0
d) set display of freq and power during sweep 0
m) show maximum then minimum of a sweep
c) set continuous sweep mode 0
F) AM step time in microseconds 1
q) AM # of samples in a burst 100
%) AM gain in percent 100
@) AM offset value 0

B) Run one AM Burst
A) Run Continuous AM Burst (on=1 / off=0) 0
P) Pulse On time is 1 us
O) Pulse Off time is 10 us
R) # of pulse repetitions is 10000
M) Pulse Off amplitude 127
G) Run one Pulse Burst
j) continuous pulse mode 0
p) get phase lock status (lock=1 / unlock=0) 1
C1) General Purpose AD read J8 P1 (0-1023) 0
C2) General Purpose AD read J8 P2 (0-1023) 0
#) set GP Dig Out J8 Pin 3 (on=1 / off=0) 0
$) set GP Dig Out J8 Pin 5 (on=1 / off=0) 0
H0) PLL Register 0 3E80000
H1) PLL Register 1 8008FA1
H2) PLL Register 2 18015E42
H3) PLL Register 3 4B3
H4) PLL Register 4 A1043C
H5) PLL Register 5 580005
*) PLL phase comparator frequency MHz 2.0
+) Model Type
-) Serial Number 99
?) help

Please keep in mind that the device expects the format shown. For example, if you send simply just an "f" the processor will sit there and wait for the rest of the data and may appear locked up. If you dont send the decimal point and at least one digit afterward, it will have unexpected results.

Send data without hidden termination characters such as a carriage return at the end. All responses will have a \n termination character except the response to "?". After sending a "?" listen to USB until the response is completely finished.

The frequency sweep and modulation routines are loops which will run for the amount of time you set them to run and may not respond to USB during this time. Please keep the run time under 1 second. If you make a mistake and have a very long run time and then save the settings in run mode you could cause the device to not respond for a long time after boot. At this point there isnt much to do but plug in the device and send a "c0, A0, or j0" and wait for the process to stop.

After programming a particular setting, you may enter "?" to get the above list of commands returned to you. The numbers that follow each setting are the actual values programmed. You may also type the command and then a "?" to get a single value back. For example, an "a?" as taken from the above set would return a value of "63\n".

During development if the device seems to stop working properly, come back and always send a "?" and check all of the values carefully. For example, a value of "0" returned for x means the SynthNV is expecting an external reference and it has nothing to lock to.

Sweeping functions explained:

Sweeping is started by sending a "g1". At the end of the sweep the SynthNV will send out 12 characters: endofsweep.\n . It does not send out endofsweep.\n if continuous sweeping mode is active (c=1). If c=1 the SynthNV just continues sweeping with no USB comms. It is not recommended to have r or d enabled(1) while c=1.

"r=1" will set the SynthNV to read its power detector after every new frequency is set in the generator side. No data will come out, other than the endofsweep.\n at the end of the sweep. But the maximum and minimum levels, in dBm, and their associated frequencies will be stored in the SynthNV RAM after the sweep. At this point sending a command of "m" will make the SynthNV respond with the maximum and minimums similar to this (maximum power first, then minimum power – and their associated frequencies in KHz):
4000000
-22.217
50000
-28.312

"d=1" will set the SynthNV to display the frequency and power after every new frequency is set in the generator side. Maximums and minimum power points are still stored as above. After sending "g1" the SynthNV will sweep and report the data as it sweeps. For the example, below it starts with the lowest frequency "l950.0" and ends with the highest frequency "u1050.0" and steps as per the setting of "s20.0". Frequency is reported in KHz and power in dBm.
950000
-10.304
970000
-10.494
990000
-10.589
1010000
-10.685
1030000
-10.780
1050000
-10.875
endofsweep.

AM functions explained:

AM is controlled by a for-next loop in the SynthNV processor that continuously reprograms a 6-bit value from a look up table to the digital RF attenuator inside the SynthNV with a sample delay time. This delay time is in microseconds and controlled via "F) AM step time in microseconds" as in the list of commands above.

Programming the look up table (LUT) is not something that can be done from a terminal emulator like the other variables. To program the LUT you send a byte value of 9 (not a character 9), then you send a byte value for the LUT size. A LUT size of 32 is typical, 255 is maximum. The smaller the LUT size, the faster and higher resolution the AM modulation will go (per cycle). 6-bit values of each element in the LUT are between 0 and 63. A typical LUT programming sequence for a sine wave from Windfreak

software will look like this: 9 32 38 44 49 54 58 61 62 63 62 61 58 54 49 44 38 32 25 19 14 9 5 2 1 0 1 2 5 9 14 19 25
Remember these values are a single byte, not characters. There are no termination characters, and no other spaces. All bytes need to go in one continuous packet with no delays between bytes. At the time, this was done to make an AM LUT transfer through USB as quickly as possible.

After the LUT is uploaded it only goes to RAM. In order to save the LUT permanently in the device an "e" (write all settings to eeprom) command needs to be sent. Be careful, as ALL settings for the device are saved at this time.

The other two AM commands perform operations on the LUT in volatile memory:
%) AM gain in percent 100
@) AM offset value 0
They scale or offset the values in the LUT. Therefore, it is best to load the waveform shape into the LUT so it takes up the full 6 bit range.