Using the Gnuradio Hermes/Metis Module

Version: January 19, 2013 Tom McDermott, N5EG

Scope

This document describes usage and some considerations in the module that implements the Gnuradio interface to the Hermes/Metis Gnuradio block. The module is called HermesNB, where NB stands for NarrowBand. The NB modules implements one 48k transmit channel, and one or two receive channels, where the receive channels can be 48k, 96k, or 192k. A future HermesWB module may provide a receive-only wideband sampled data stream.

Be sure the read the document “How to Build Gnuradio Hermes-Metis.pdf” first, and download and install the module. The HermesNB module needs to be installed in order for Gnuradio to find and recognize it.

The module has been tested with Hermes, and should work with a Metis-based system, although at this time such operation has not been confirmed.

Note: the unit is FULL DUPLEX. It is capable of transmitting and receiving simultaneously. Thus, some parameters have been added to control transmitter and receiver muting. It means that a Gnuradio source of symbols must always be connected to the ‘in’ connector, even if you are not planning to transmit. Gnuradio provides a CONSTANT SOURCE, which you can set to put out zero valued (complex) samples which effectively disables the transmitter.

Usage

The HermesNB module can be selected from the Gnuradio Add panel, and a copy dragged onto the GnuRadio Companion (GRC) design surface. Double-click that block which will open the module for parameter editing. The module will look like figure 1. There are two types of parameters, those which are only used when the GRC flow graph is started, and those that have interactive capabilities whilst the GRC flow graph is running. The interactive parameters are underlined, while the start parameters are not underlined.

If you scroll down the block, the brief documentation for the block becomes visible (figure 2).

Start Parameters

The following parameters are referenced only when the flow graph is started. Their value does not interactively change the flow graph while it is running:

- ID

The Id is arbitrary. Gnuradio uses it to distinguish a block from all other blocks, it must have a unique value in the flow graph. Gnuradio will generate a default value for you when you drag the block onto the design surface, and usually there's no reason for you to need to change it (but you can).

- Num Receivers (1 or 2).

The Num Receivers parameter determines if the block will implement one or two receivers. If the value 1 is selected, the block will have one output connector 'out0'. If the 2 is entered, the block will have two output connectors, 'out0' and 'out1'. Receive Frequency 0 controls the Receiver 0, whose signals are emitted on the 'out0' connector. The Receiver Frequency 1 control similarly tunes Receiver 1, whose signals are emitted on the 'out1' connector.

- Ethernet Interface

This is a text field indicating which Ethernet interface you want to use to connect to the Hermes or Metis module. The default interface in Linux is eth0 (all lower case). Gnuradio requires the parameter to be enclosed in double-quote marks: for example "eth0". When the flow graph is started, the hermesNB module will print a list of all interfaces it found on the host. Usually these are the loopback interface "lo" and the Ethernet interface "eth0" but your system may have additional or different choices.

Interactive Parameters

The interactive parameters can be changed while the flow graph is running. However you cannot change them by typing values into the block – they must be changed by other flow graph blocks (such as variables, sliders, etc.). This may be sort of confusing, but it's the way the Gnuradio works.

- RxSamp = 48000, 96000, or 192000

The receive sample rate sets the output sample rate of the receivers in samples per second. There is one 24-bit I and one 24-bit Q value for each sample. The allowed values are 48000, 96000, and 192000.

- RxPreamp = 0 (Off), or 1 (On)

The value 0 inserts 20 dB of attenuation into the receiver. This is listed as preamp off by the Hermes documentation. The value 1 removes the 20 dB attenuation from the receiver input.

- PTTMode = 0 (TxOff), 1 (Vox on Txframe contents), or 2 (TxOn)

This controls the PTT control output bit, which for example you might have wired to an external T/R relay. IT DOES NOT TURN THE TRANSMITTER ON OR OFF. The transmitter is ALWAYS ON, however you can cause the transmitter to produce no output while ON.

The value 0 means that the PTT control bit on the output connector is set to off (de-asserted).
A value of 1 causes the PTT control bit to become asserted when the transmit data stream contains non-zero characters. If the transmit data stream contains all-zero symbols, the PTT output control bit will be de-asserted.
A value of 2 asserts the PTT control output bit (on).

- PTTOffMutesTx = 0 (Tx always emits), 1 (Tx IQ is zeroed when PTT is Off)

This parameter allows muting the transmit stream when PTT is Off. The value 0 means that the transmitter stream is unaltered internal to the module (whatever symbols that you place into the transmit stream are sent to the transmitter). The value 1 means that if PTT is OFF, that any symbols that you have placed into the transmit stream are replaced by all zeros. This effectively mutes the transmitter causing it not to put out any RF power.

- PTTOnMutesRx = 0 (Rx always receives), 1 (Rx IQ is zeroed when PTT is On)

This parameter allows muting the receive stream when PTT is On. The value 0 means that the receive stream is unaltered and it will receive even while you are transmitting. The value 1 means that if PTT is On, that all the received symbols will be zeroed out. This effectively mutes the receiver causing it not to make sound while you are transmitting.

- TxDrive = 0..255 (0 is minimum (but not zero) drive, 255 is maximum drive)

The Transmit Drive level has a value of 0 to 255. The value 255 provides maximum transmit output (when the I and Q transit symbols reach peak value) and the value 0 provides the minimum output power. Note that a value of zero does not completely disable the transmitter; the transmitter will still emit some small value of signal. In order to completely kill all transmit output signal, the transmit I and Q symbols must be zero.

- Clock Source

This field allows control of the HPSDR Clock and Mic selectors. The Hermes module ignores this byte (since all the selectors are fixed). The Atlas bus based HPSDR systems have controllable clock sources. The value of this parameter mirrors value of the control byte C1 when the register index (C0) has a value of zero. The lower 2 bits of this register are ignored, while the upper 6 bits mirror the control register (shown below).

The Default value selects Mercury as the clock source, and it has a value of “0xfc” {including quote marks} . See the "HPSDR - USB Data Protocol" document (portion reproduced below) for details of the control register and the resultant actions.

Command & Control

NOTE: Bits **7-1** of C0 form an address that determines how C1-C4 should be decoded. C0 is varied round-robin fashion so that all addresses are sent in sequence.

```
C0
0 0 0 0 0 0 0 0
              |
              +------------ MOX (1 = active, 0 = inactive)
C1
0 0 0 0 0 0 0 0
| | | | | | | |
| | | | | | + +------------ Speed (00 = 48kHz, 01 = 96kHz, 10 = 192kHz)
| | | | + +---------------- 10MHz Ref. (00 = Atlas/Excalibur, 01 = Penelope, 10 = Mercury)*
| | | +-------------------- 122.88MHz source (0 = Penelope, 1 = Mercury)*
| + +---------------------- Config (00 = nil, 01 = Penelope, 10 = Mercury, 11 = both)*
+-------------------------- Mic source (0 = Janus, 1 = Penelope)*
```

* Ignored by Hermes

- Alex Control

This provides 4 bytes to control an Alex module attached to either Hermes or an Atlas bus based HPSDR system. Various different registers in HPSDR have been concatenated into one 32-bit value for this parameter.

The high-order byte of the parameter (bits 24-31) are sent to control register C3 (when C0=0).
The second-order byte of the parameter (bits 16-23) are sent to control register C4 (when C0=0).
The third-order byte of the parameter (bits 8-15) are sent to control register C3 (when C0=0x12).
The low-order byte of the parameter (bits 0-7) are sent to control register C4 (when C0=0x12).
Example: "0x00000000" {including quote marks}. Please see the "HPSDR - USB Data Protocol" document (portion shown below) for definitions of the control registers and the resultant actions.

Command & Control

NOTE: Bits **7-1** of C0 form an address that determines how C1-C4 should be decoded. C0 is varied round-robin fashion so that all addresses are sent in sequence.

```
C0
0 0 0 0 0 0 0 0

              |
              +------------ MOX (1 = active, 0 = inactive)

C3
0 0 0 0 0 0 0 0
| | | | | | | |
| | | | | | + +------------ Alex Attenuator (00 = 0dB, 01 = 10dB, 10 = 20dB, 11 = 30dB)
| | | | | +---------------- Preamp On/Off (0 = Off, 1 = On)
| | | | +------------------ LT2208 Dither (0 = Off, 1 = On)
| | | + ------------------- LT2208 Random (0= Off, 1 = On)
| + + --------------------- Alex Rx Antenna (00 = none, 01 = Rx1, 10 = Rx2, 11 = XV)
+ ------------------------- Alex Rx out (0 = off, 1 = on). Set if Alex Rx Antenna > 0.


C4
0 0 0 0 0 0 0 0
| | | | | | | |
| | | | | | + + ----------- Alex Tx relay (00 = Tx1, 01= Tx2, 10 = Tx3)
| | | | | + --------------- Duplex (0 = off, 1 = on)
| | + + +------------------ Number of Receivers (000 = 1, 111 = 8)
| +------------------------ Time stamp – 1PPS on LSB of Mic data (0 = off, 1 = on)
+-------------------------- Common Mercury Frequency (0 = independent frequencies to Mercury
                             Boards, 1 = same frequency to all Mercury boards)


-------------
C0
0 0 0 1 0 0 1 x

C3
0 0 0 0 0 0 0 0
  | | | | | | |
  | | | | | | +------------ Alex   -  select 13MHz  HPF (0 = disable, 1 = enable)2
  | | | | | +-------------- Alex   -  select 20MHz  HPF (0 = disable, 1 = enable)2
  | | | | +---------------- Alex   -  select 9.5MHz HPF (0 = disable, 1 = enable)2
  | | | +------------------ Alex   -  select 6.5MHz HPF (0 = disable, 1 = enable)2
  | | +-------------------- Alex   -  select 1.5MHz HPF (0 = disable, 1 = enable)2
  | +---------------------- Alex   -  Bypass all HPFs   (0 = disable, 1 = enable)2
  +------------------------ Alex   -  6M low noise amplifier (0 = disable, 1 = enable)2

C4
0 0 0 0 0 0 0 0
  | | | | | | |
  | | | | | | +------------ Alex   -  select 30/20m LPF (0 = disable, 1 = enable)2
  | | | | | +-------------- Alex   -  select 60/40m LPF (0 = disable, 1 = enable)2
  | | | | +---------------- Alex   -  select 80m    LPF (0 = disable, 1 = enable)2
  | | | +------------------ Alex   -  select 160m   LPF (0 = disable, 1 = enable)2
  | | +-------------------- Alex   -  select 6m     LPF (0 = disable, 1 = enable)2
  | +---------------------- Alex   -  select 12/10m LPF (0 = disable, 1 = enable)2
  +------------------------ Alex   -  select 17/15m LPF (0 = disable, 1 = enable)2

2 Only valid when Alex - manual HPF/LPF filter select is enabled
```

Miscellaneous

Throttle

Sometimes Gnuradio will give you a warning that there is no 'throttle' control, and that you should insert one unless you are using the USRP module. DO NOT USE A THROTTLE if you are using the hermesNB block. The purpose of the throttle is to prevent CPU starvation for Gnuradio flow graphs that have no physical source of timing to limit the computation rate. The hermesNB block provides actual timing to the flow graph and thus a throttle should not be used.

Audio Sink

The Gnuradio Audio Sink Block can be used to produce sound via the computer soundcard. At this time, HermesNB does not support the audio amplifier or Mic input on the HermesNB or Metis based Atlas bus systems, so the audio sink is the only way to produce sound. The Gnuradio audio sink block has selectable sample rates with a drop down menu, but sample rates not on the drop-down are also supported (for example 8000 samples per second). Not all audio cards support all rates however. The audio card drive may issue an error message or it may not if it does not support a chosen sample rate. The Gnuradio audio sink block can be set for one (mono) or two (stereo) inputs.

The audio sink defaults to BLOCKING mode which causes severe problems with HermesNB. This is because HermesNB provides the pacing of samples, not the audio soundcard. Thus the audio Blocking on the audio sink block must be turned off, i.e. set to the NON BLOCKING mode.

There will be a mismatch between the exact speed of the HermesNB timing and the sound card sample timing. An underflow for example will cause a text display of aUaUaUaUaUaU… on the text control panel. It is only informational and does not impact the audio quality.

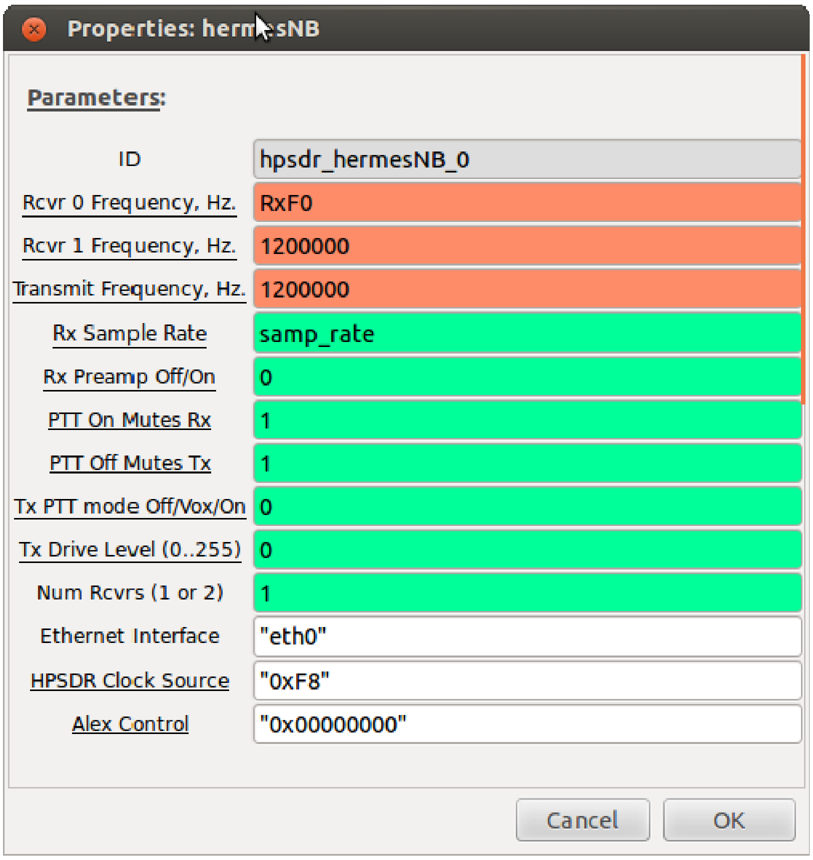
Network Addressing & Status

The HermesNB application will print the addresses of the host and the Hermes/metis board that it discovers on the text panel. The text below shows an example. The buffer counts may or may not appear depending on when Gnuradio shuts down the HermesNB modules compared to the text buffer thread timing.

```
Executing: "/home/tom/Desktop/top_block.py"
>>> gr_fir_ccf: using SSE
Looking for Metis/Hermes card on interface eth0
```

```
Interface[0]:lo  Interface[1]:eth0
eth0 IP Address: 192.168.0.2
eth0 MAC Address: 00:80:ad:74:5f:5e
Metis MAC address 00:04:A3:63:F9:8E
Metis IP address 192.168.0.4
Using Volk machine: sse2_32_mmx
aUaUaU
LostRxBufCount = 0  TotalRxBufCount = 1657  LostTxBufCount = 1
TotalTxBufCount = 1657  CorruptRxCount = 0  LostEthernetRx = 0

>>> Done
```

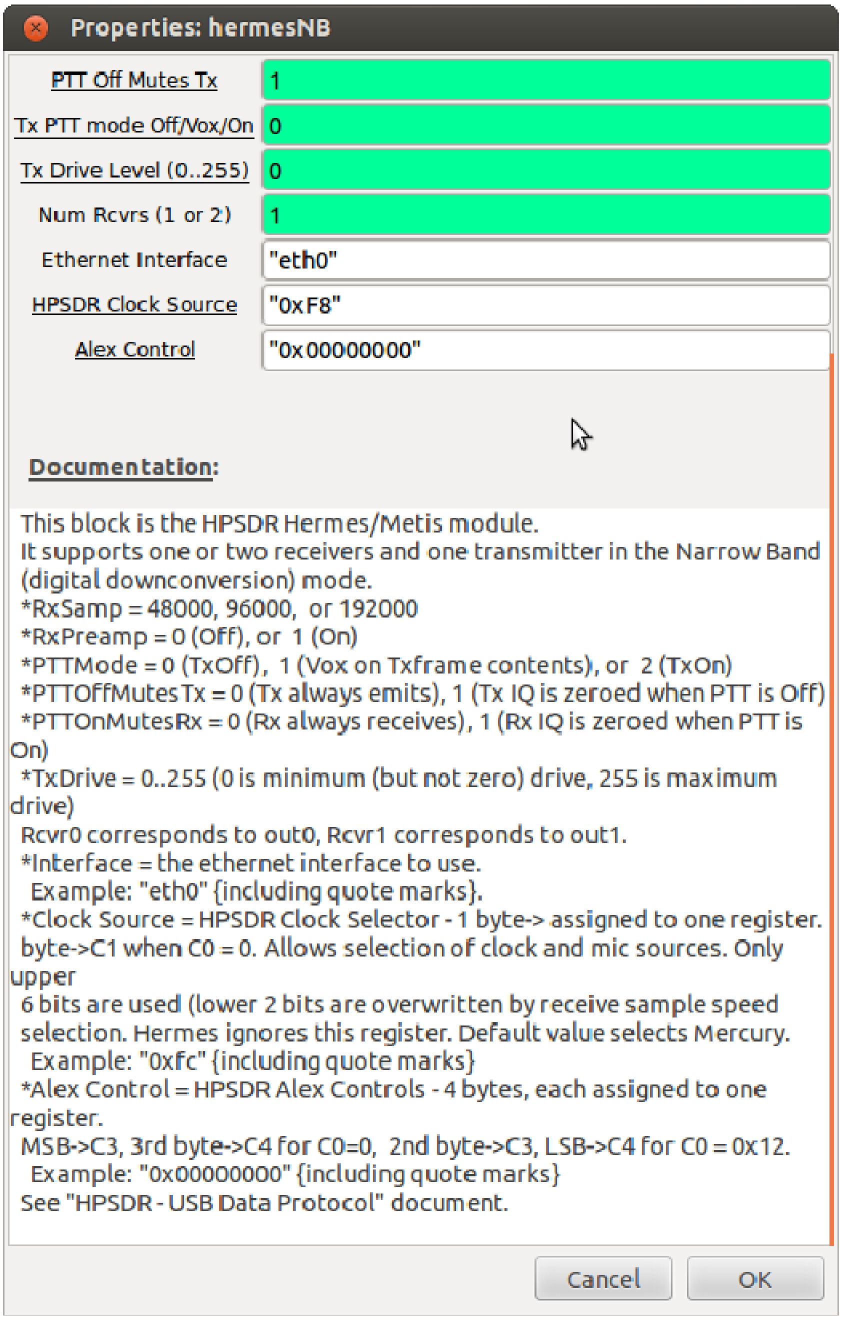


Properties: hermesNB

Parameters:

ID	hpsdr_hermesNB_0
Rcvr 0 Frequency, Hz.	RxF0
Rcvr 1 Frequency, Hz.	1200000
Transmit Frequency, Hz.	1200000
Rx Sample Rate	samp_rate
Rx Preamp Off/On	0
PTT On Mutes Rx	1
PTT Off Mutes Tx	1
Tx PTT mode Off/Vox/On	0
Tx Drive Level (0..255)	0
Num Rcvrs (1 or 2)	1
Ethernet Interface	"eth0"
HPSDR Clock Source	"0xF8"
Alex Control	"0x00000000"

Figure 1 – HermesNB block parameters



PTT Off Mutes Tx	1
Tx PTT mode Off/Vox/On	0
Tx Drive Level (0..255)	0
Num Rcvrs (1 or 2)	1
Ethernet Interface	"eth0"
HPSDR Clock Source	"0xF8"
Alex Control	"0x00000000"

Documentation:

This block is the HPSDR Hermes/Metis module.
It supports one or two receivers and one transmitter in the Narrow Band (digital downconversion) mode.
*RxSamp = 48000, 96000, or 192000
*RxPreamp = 0 (Off), or 1 (On)
*PTTMode = 0 (TxOff), 1 (Vox on Txframe contents), or 2 (TxOn)
*PTTOffMutesTx = 0 (Tx always emits), 1 (Tx IQ is zeroed when PTT is Off)
*PTTOnMutesRx = 0 (Rx always receives), 1 (Rx IQ is zeroed when PTT is On)
*TxDrive = 0..255 (0 is minimum (but not zero) drive, 255 is maximum drive)
Rcvr0 corresponds to out0, Rcvr1 corresponds to out1.
*Interface = the ethernet interface to use.
Example: "eth0" {including quote marks}.
*Clock Source = HPSDR Clock Selector - 1 byte-> assigned to one register. byte->C1 when C0 = 0. Allows selection of clock and mic sources. Only upper
6 bits are used (lower 2 bits are overwritten by receive sample speed selection. Hermes ignores this register. Default value selects Mercury.
Example: "0xfc" {including quote marks}
*Alex Control = HPSDR Alex Controls - 4 bytes, each assigned to one register.
MSB->C3, 3rd byte->C4 for C0=0, 2nd byte->C3, LSB->C4 for C0 = 0x12.
Example: "0x00000000" {including quote marks}
See "HPSDR - USB Data Protocol" document.

Figure 2 – HermesNB block scrolled down to the documentation section