HPSDR - USB Data Protocol

Revisions

Rev	Date	Changes	By
0.1	29 Jan 06	Original draft	VK6APH
0.2	30 Jan 06	Added I/Q designation to packet diagram. Added BPF and LPF to be on I2C	VK6APH
0.3	7 Feb 06	Modified number of bytes in sync/command bytes to be the same in both directions	VK6APH
0.4	9 Feb 06	Added explanation regarding choice of sync bytes	VK6APH
0.5	20 Feb 06	Changed protocol so that Microphone/Line data is sent at all times. Removed I2C data from control packets since now sent via FX2	VK6APH
1.0	25 Feb 06	Added note that V1.0 of Janus code runs at 48kHz	VK6APH
1.4	25 Feb 06	First version for public comment	VK6APH
1.5	1 May 06	Added MOX from PC, dot and dash inputs and A/D speed control	VK6APH
1.6	1 Aug 06	Updated Sync characters	VK6APH
1.7	28 May 07	Revised C&C data format	VK6APH
1.8	10 Sep 07	Revised C&C data format to include Penny and Mercury	VK6APH
1.9	17 Sept 07	Added number of bytes in FIFO to Tx protocol	VK6APH
1.10	24 Feb 08	Changed 125MHz clock reference to 122.88MHz. Changed Alex Attenuator options to 0/10/20/30dB. Added Class E or Normal mode. Added LT2208 Preamp gain Added LT2208 Overflow Added LT2208 Dither	VK6APH
1.11	25 May 08	Added LT2208 Random	VK6APH
1.12	2 June 08	Correct Left Right data	VK6APH
1.13	14 June 08	Added Alex antenna switching data	VK6APH
1.14	31 Jan 09	Changed LT2208 Preamp to Preamp Added Software serial numbers for Mercury and Penny Added ADC samples on EP4	VK6APH
1.15	3 Feb 09	Added Software serial number for Ozy	VK6APH
1.16	17 Feb 09	Added Penelope Forward Power	VK6APH
1.17	28 Mar 09	Added note regarding sampling rates to Ozy + EP4 notes	VK6APH
1.18	11Apr 09	Added note regarding initial clock selection	VK6APH
1.19	21Apr 09	Added support for Excalibur 10MHz clock	VK6APH
1.20	25 June 09	Explained Penny mic data	VK6APH
1.21	10 Aug 09	Split dot & PTT into separate signals (from Ozy V1.6)	VK6APH
1.22	14 Aug 09	Added fully Duplex capability (from Ozy V1.6) and multiple Mercury receivers (incomplete)	VK6APH
1.23	9 Sept 09	Completed multiple Mercury receiver support. Changed to fixed width font (Courier New – 8 pt)	VK6APH
1.24	13 Nov 09	Added support for Hermes and Penny Lane	VK6APH

Protocol Overview:

- The USB data consists of 512 byte packets
- The sample rate from the receiver A/D converter to the PC is selectable between 48/96/192kHz at 24 bits
- The sample rate from the microphone to the PC is 48kHz at 16 bits
- The sample rate from the PC to the speakers/headphones is 48kHz at 16 bits
- The sample rate from the PC to the I/Q transmit audio is 48kHz at 16 bits
- Control signals that are high priority are sent each 512 block, lower priority data is sent less frequently

Functions required:

- PTT
- Dot/dash key active
- A/D sampling speed 192/96/48k
- NCO Frequencies
- Penelope Open Collector outputs
- Mercury Pre-amps and attenuator

Protocol

The protocol consists of a 512 byte frame consisting of a sync sequence, Command & Control data and ADC or DAC data.

A frame length of 512 bytes is used since this is the maximum number of bytes that the FIFO in the FX2 USB interface can hold.

High priority control data is sent as part of each frame e.g. PTT command/request. Lower priority data is sent as available on a predefined schedule e.g. NCO frequency.

Sync Sequence

This consists of a three byte sync sequence comprising <0x7F><0x7F><0x7F> . The sync sequence is sent at the start of every 512byte frame and appears at the start of the frame.

Protocol – From HPSDR to PC

HPSDR sends data to the PC over USB using End Points (EP) 4 and 6.

EP4.

HPSDR sends to EP4 a block of 4096 x 16 bit raw ADC samples. These are intended to be used to create a separate bandscope or 'scope display of the RF input.
Data can be read from EP4 in 4k blocks. The start of each block will always be the start of the block of samples; hence no sync or start of block signal is required.

Data can be read at any time, it is not necessary to read at any particular data rate. At present the sample rate is 48kHz so a 4k buffer is available 10.7 time per second.

EP6.

The protocol consists of a frame of 512 bytes. Each frame starts with three sync bytes (0x7F, 0x7F, 0x7F) followed by five Command and Control (C&C) bytes (C0..C4). The first C&C byte(C0) bits [7:3] are used as an address that indicates what the next four bytes contain.

The balance of the frame consists of I, Q and microphone/line in samples (or left, right and microphone/line samples if a Janus card is being used).

For a **single** Mercury board/receiver, Hermes or Janus the protocol is as follows:

```
0                                                                   16
<Sync><Sync><Sync><C0><C1><C2><C3><C4><I2><I1><I0><Q2><Q1><Q0><M1><M0>

17                                                                  32
<I2><I1><I0><Q2><Q1><Q0><M1><M0><I2><I1><I0><Q2><Q1><Q0><M1><M0>

etc

504                         511
<I2><I1><I0><Q2><Q1><Q0><M1><M0>
```

Where:

```
Sync    - 0x7F
Cn      - Command/Control Byte
I2      - Bits 23-16 of I sample (Mercury/Hermes) or Left sample (Janus)
I1      - Bits  15-8 of I sample (Mercury/Hermes) or Left sample (Janus)
I0      - Bits   7-0 of I sample (Mercury/Hermes) or Left sample (Janus)
Q2      - Bits 23-16 of Q sample (Mercury/Hermes) or Left sample (Janus)
Q1      - Bits  15-8 of Q sample (Mercury/Hermes) or Left sample (Janus)
Q0      - Bits   7-0 of Q sample (Mercury/Hermes) or Left sample (Janus)
M1      - Bits  15-9 of Mic/Line sample
M0      - Bits   7-0 of Mic/Line sample
```

For **multiple** Mercury boards, or multiple receivers in the one Mercury/Hermes board, the protocol is extended as follows:

E.g. with 3 receivers

```
0                                   7
<Sync><Sync><Sync><C0><C1><C2><C3><C4>

8                                                                                                   27
<I12><I11><I10><Q12><Q11><Q10><I22><I21><I20><Q22><Q21><Q20><I32><I31><I30><Q32><Q31><Q30><M1><M0>

28                                                                                                  47
<I12><I11><I10><Q12><Q11><Q10><I22><I21><I20><Q22><Q21><Q20><I32><I31><I30><Q32><Q31><Q30><M1><M0>

Etc

492                                                             511
<Q22><Q21><Q20><I32><I31><I30><Q32><Q31><Q30><M1><M0><0><0><0><0>
```

Where:

I_n2 - Bits 23-16 of I sample for receiver n etc.

NOTE 1: where there are insufficient samples to exactly fill a 512 byte frame then the end of the frame is padded with 0s. The number of padded 0s is as follows:

Number of receivers	Padding
1	0
2	0
3	4
4	10
5	24
6	10
7	20
8	4

NOTE 2: The sample rate of the Microphone data is always 48kHz irrespective of the L/R (I&Q) sample rates. At 96/192kHz sample rates the microphone data is just duplicated and additional samples can be discarded as required.

Command & Control

NOTE: Bits **7-3** 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
          | | |
          | | + --------- PTT  (1 = active, 0 = inactive), GPIO[23]= Ozy J8-8, Hermes J11-1
          | +------------ DASH (1 = active, 0 = inactive), GPIO[21]= Ozy J8-6, Hermes J11-4
          +-------------- DOT  (1 = active, 0 = inactive), GPIO[22]= Ozy J8-7, Hermes J11-5
```

C1

```
0 0 0 0 0 0 0 0
        | | | |
        | | | +---------- LT2208 Overflow (1 = active, 0 = inactive)
        | | +------------ Hermes I01 (0 = active, 1 = inactive)
        | +-------------- Hermes I02 (0 = active, 1 = inactive)
        +---------------- Hermes I03 (0 = active, 1 = inactive)
```

C2 - Mercury or Hermes software serial number (0 to 255)
C3 - Penelope software serial number (0 to 255) - set to 0 when Hermes
C4 - Ozy software serial number (0 to 255) - set to 0 when Hermes

C0
0 0 0 0 **1** x x x

C1 - Bits 15-8 of Forward Power from Penelope or Hermes* (AIN5)
C2 - Bits 7-0 of Forward Power from Penelope or Hermes (AIN5)
C3 - Bits 15-8 of Forward Power from Alex or Apollo (AIN1)
C4 - Bits 7-0 of Forward Power from Alex or Apollo (AIN1)

C0
0 0 0 **1** 0 x x x

C1 - Bits 15-8 of Reverse Power from Alex or Apollo (AIN2)
C2 - Bits 7-0 of Reverse Power from Alex or Apollo (AIN2)
C3 - Bits 15-8 of AIN3 from Penny or Hermes
C4 - Bits 7-0 of AIN3 from Penny or Hermes

C0
0 0 0 **1 1** x x x

C1 - Bits 15-8 of AIN4 from Penny or Hermes
C2 - Bits 7-0 of AIN4 from Penny or Hermes
C3 - Bits 15-8 of AIN6 from Penny or Hermes (13.8v supply on Hermes)
C4 - Bits 7-0 of AIN6 from Penny or Hermes (13.8v supply on Hermes)

*Note: All analog levels are 12 bits.

IMPORTANT: From V1.3 of the Ozy FPGA code, at reset the clock selection in Ozy *DEFAULTS TO MERCURY*. If a Mercury board is fitted then frames from Ozy will be sent immediately. If a Mercury board is not present then it will be necessary to send a few C&C frames to select the desired clock source before data can be received from Ozy.

Protocol – From PC to HPSDR

The PC sends Command and Control plus two audio streams to the HPSDR on End Point 2 (EP2). The audio signals are:

1. 48kHz 16 bit Left/Right received audio
2. 48kHz 16 bit I/Q

Since the received audio is also used to monitor the transmitted audio then these two streams must be available simultaneously.

NOTE: The sampling rate, and hence data rate, is *ALWAYS* 48kHz and is independent of the sampling rate (e.g. 192/96/48kHz) set on the HPSDR to PC link.

IMPORTANT: From V1.3 of the Ozy FPGA code, at reset the clock selection in Ozy *DEFAULTS TO MERCURY*. If a Mercury board is fitted then frames from Ozy will be sent immediately. If a Mercury board is not present then it will be necessary to send a few C&C frames to select the desired clock source before data can be received from Ozy.

Since the DACs use 16 bits per sample then, in order that an integer number of Left/Right and I/Q samples will be included in the 512 byte packed, the maximum number of samples is

(512 – 8) = 63 x 4 x 2 bytes i.e. 63 Receiver L/R samples and 63 I/Q L/R samples

This provides 8 bytes to transfer status data from the PC to the FPGA. The first characters in the 512 byte packet will be sync which is 0x7F7F7F. Since 3 bytes are required for the sync character 5 bytes are used to send Command & Control data.

```
0                                                                   15
<Sync><Sync><Sync><C0><C1><C2><C3><C4><L1><L0><R1><R0><I1><I0><Q1><Q0>

 16                                                                 31
<L1><L0><R1><R0><I1><I0><Q1><Q0><L1><L0><R1><R0><I1><I0><Q1><Q0>

Etc

496                                                                511
<L1><L0><R1><R0><I1><I0><Q1><Q0><L1><L0><R1><R0><I1><I0><Q1><Q0>
```

Where:

```
Sync    – 0x7F
Cn      – Command/Control Byte
L1      – Bits  15-8 of Left audio sample
L0      – Bits   7-0 of Left audio sample
R1      – Bits  15-8 of Right audio sample
R0      – Bits   7-0 of Right audio sample
I1      – Bits  15-8 of I sample
I0      – Bits   7-0 of I sample
Q1      – Bits  15-8 of Q sample
Q0      – Bits   7-0 of Q sample
```

Note: When using Hermes the Transmitter output level is set by the drive level value (C0 = 0b0001001x, C1 = 0x00 to 0xFF) and not by the amplitude of the I&Q signals. These are held to a peak value of +/-1.0 by AGC action in the PC DSP code. When using Penelope the amplitude of the I&Q signals controls the output level.

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

C2

```
0 0 0 0 0 0 0 0
|           | |
|           | +------------ Mode (1 = Class E, 0 = All other modes)
+---------- +-------------- Open Collector Outputs on Penelope or Hermes (bit 6…..bit 0)
```

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)
```

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)
```

C0
0 0 0 0 0 0 **1** x **C1, C2, C3, C4** NCO Frequency in Hz for Transmitter (and Receiver if C4[2] is **not** set) (32 bit binary representation - MSB in C1)

C0
0 0 0 0 0 **1** 0 x **C1, C2, C3, C4** NCO Frequency in Hz for Receiver_1 if C4[2] is set

C0
0 0 0 0 0 **1 1** x **C1, C2, C3, C4** NCO Frequency in Hz for Receiver _2 if C4[2] is set

C0
0 0 0 0 **1** 0 0 x **C1, C2, C3, C4** NCO Frequency in Hz for Receiver _3 if C4[2] is set

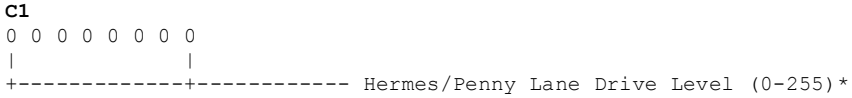
C0
0 0 0 0 **1** 0 **1** x **C1, C2, C3, C4** NCO Frequency in Hz for Receiver _4 if C4[2] is set

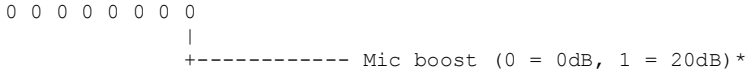
C0
0 0 0 0 **1 1** 0 x **C1, C2, C3, C4** NCO Frequency in Hz for Receiver _5 if C4[2] is set

C0
0 0 0 0 **1 1 1** x **C1, C2, C3, C4** NCO Frequency in Hz for Receiver _6 if C4[2] is set

C0
0 0 0 **1** 0 0 0 x **C1, C2, C3, C4** NCO Frequency in Hz for Receiver _7 if C4[2] is set

C0
0 0 0 **1** 0 0 **1** x

C1


C2


* Ignored by Penelope