

DRAFT

# OpenHPSDR Ethernet Protocol

V1.1

©June 2015 Phil Harman VK6PH

## Revisions

Rev	Date	Changes	By
1.0	20 June 2015	First release	VK6PH
1.1	27 June 2015	Changed Hermes Lite reference number from 5 to 6. Added DSP clock frequency and number of receivers implemented to Discovery reply. Enable either receiver and transmitter frequency or phase word to be sent to hardware. Added memory mapped registers.	VK6PH KF7O

## **Acknowledgements**

My thanks to Jeremy, NH6Z, for providing initial the suggestions regarding a possible protocol and for the suggested document format.

Thanks also to Alex, VE3NEA, for the original suggestion of using UDP ports to send and receive data from.

Special thanks to Scotty, WA2DFI, for his detail analysis of the early drafts of this protocol and numerous significant contributions.

The follow have provided input and review of the protocol during its development:

Warren - NR0V, Doug - W5WC, Joe - K5SO, Chen - W7AY, George - K9TRV, Leif – SM5BSZ, Simon - G4ELI, Vasiliy - K3IT, Tom - N5EG, Edson - PY2SDR, Dave - KV0S, Alex - VE3NEA, John – G0ORX, Francis - ON5RF, Alberto – I2PHD, David - WA8YWQ, Roger – W3SZ, Bob - G3UKB, Steve-K7FO.

Thanks also to those contributors who wish to remain anonymous.

## TABLE OF CONTENTS

Purpose.....	1
Architecture.....	1
Detailed operation.....	3
OpenHPSDR Host to Hardware Protocol .....	4
Discovery Packet.....	5
Erase Packet.....	6
Program Packet.....	7
Set IP Address Packet .....	9
General Packet to SDR .....	11
RECEIVER SPECIFIC packet .....	15
Transmitter SPECIFIC packet .....	19
Transmitter Synchronisation Packet.....	22
High Priority DATA Packet .....	23
Receiver Audio Packet .....	28
Transmitter I&Q Data Packet.....	29
Memory mapped Registers From PC.....	31
OpenHPSDR Hardware to Host Protocol .....	33
Discovery Reply Packet .....	34
Command Reply Packet.....	36
high priority Status Packet.....	37
Microphone Data Packet .....	40
Wideband Data Packet .....	41
Receiver Packet.....	43
Memory mapped registers from hardware.....	45
Figure 1 UDP Data from Host.....	47
Figure 2 UDP Data to Host.....	48

Figure 3 Receiver Architecture .....	49
Figure 4 Transmitter Architecture .....	50
Appendix A.....	51

## PURPOSE

This specification describes the protocol used to communicate with current and future openHPSR hardware. Its intended audience is hardware and software developers who wish to develop, or modify, hardware and software to use this architecture and protocol.

## ARCHITECTURE

The basic architecture is built on the concept of using UDP ports to send and receive signals and Command & Control data.

Some configuration settings will change infrequently and can be applied to all receivers and transmitters. These 'General' settings need only be sent when the SDR hardware is first turned on or when a setting changes during operation.

Settings that need to change more frequently are sent to control registers associated with receivers or transmitters.

Settings that require near real-time responses are sent as priority packets to specific UDP ports.

Where possible the protocol does not restrict the number of configurable resources. The number of fully independent receivers, associated with a particular analogue to digital converter (ADC), can be configured using the General settings. Similarly, the number of receivers or transmitters that may be combined synchronously or multiplexed is configurable.

The basic architecture for UDP data from the Host (i.e. PC, Tablet etc) is shown in Figure 1. The UDP ports here configure General Registers (where data changes infrequently after initialisation), Transmitter and Receiver Specific Registers and High Priority Registers (where, for example, register settings change between receive and transmit).

Figure 2 illustrates the architecture for UDP data from the SDR hardware to the Host. The ports here include the Microphone data and both wide and narrow band receiver data ports.

In both diagrams some UDP ports are predetermined. Others are set by the user and/or depending on the actual number of ADCs and DACs available on the specific SDR hardware and the number of synchronous and non-synchronous (i.e. multiplexed) receivers and transmitters the user wishes to configure.

Figure 3 illustrates how multiple receivers can be configured on a specific ADC. It also shows how a specific receiver can be configured to be synchronous or multiplexed with other receivers configured on other ADCs.

In general the input of a specific Digital Down Converter (DDC) can be fed from a selected ADC or, for PureSignal requirements, from the data being sent to a specific DAC. Each DDC can operate independently from all others i.e. its sampling rate is set independently.

Where no synchronous or multiplexed receivers are selected then the output of the DDC directly feeds a FIFO and subsequently, using its designated port, the Ethernet PHY.

Where synchronous or multiplexed receivers are selected then the data from these additional receivers are sequentially passed to the FIFO associated with the base receiver DDC. All synchronous or multiplexed receivers must operate at the same sampling rate.

Synchronous receivers are phase coherent whilst multiplexed receivers can be set to different frequencies.

The maximum number of synchronous receivers that can be selected per base receiver is equal to the number of ADCs. The maximum number of multiplexed receivers per base receiver is equal to the number of receivers implemented.

The DDC follows the conventional architecture of CORDIC, CIC filters and sinc compensating FIR filter.

Figure 4 illustrates how multiple transmitters can be configured in either synchronous or multiplexed modes. Presently, no openHPSDR SDR hardware is able to support more than one DAC. In which case data from the FIFO connect to port 1029 (default) is connected directly to the Digital Up Converter (DUC) and the selection of synchronous or multiplexed data is not supported.

The transmitter uses a DUC configured as a 5 stage CIC filter and CORDIC. Note that the Host I&Q data (24 bit I&Q data at 192ksps) directly feeds the CIC filter. In which case the transmitted signal will be -1dB at +/- 22kHz. For most modes this is acceptable but should wider bandwidths be required the designer may wish to include a sinc compensation filter in the Host software.

In order to overcome any potential latency issues between the Host and SDR Hardware, RF and sidetone generation for CW is done in the SDR Hardware. This includes applying a raised cosine profile to the leading and trailing edges of the CW RF and sidetone waveforms. An Iambic Keyer is also implemented and operates in Straight, simulated Bug and Iambic A or B.

Variable frequency and amplitude sidetone is also generated in the SDR Hardware.

Time stamping of receiver I & Q packets is in accordance with the VITA-49 specification for Fractional-seconds Timestamps.

For the current hardware implementation an arbitrary limit of 80 receivers and 8 ADCs has been applied. These limits will be removed as hardware that is capable of exceeding these settings becomes available.

Compared with the previous openHPSDR Ethernet/USB protocol, numerous new features and facilities have been added. However, we should bear in mind the 'Second-system effect' - see

[http://en.wikipedia.org/wiki/Second-system\\_effect](http://en.wikipedia.org/wiki/Second-system_effect)

## DETAILED OPERATION

Consider a network made up of multiple SDRs and multiple Hosts (e.g. PCs). Each SDR on the network has a unique IP address, and listens on fixed port 1024 for Discovery packets.

The following scenario outlines how SDRs and Hosts interact within a given network segment. It is important that multiple SDRs and multiple Hosts each running multiple applications be able to interact on one segment. Note that the term “SDR” applies to a single network connection (IP address); there may be many hardware ports associated with each single IP address.

To establish communication, one network Host will broadcast a Discovery to address <255.255.255.255:1024> from its own IP address and a source port number. In this example, assume it is from <192.168.1.10:8000>.

Every device listening on port 1024 will respond to this broadcast with a response to the Hosts IP address and source port with its IP address and port number. In this example, the SDR responds to <192.168.1.10:8000> from its own fixed address of <192.168.1.30:1024>. Now a command/response channel has been established between the SDR and the Host. This channel will be used for all SDR-to-Host and Host-to-SDR communications until one or more other streams are established.

Once a channel has been established then the General Registers and Receiver and Transmitter Specific Registers should be set up. A High Priority packet should then be sent with the ‘run’ bit set in order to start data from the Hardware.

The format of the Ethernet packets is generally consistent in that they all commence with a 32 bit sequence number. This enables the Host and Hardware to determine if packets have been lost.

In general, configuration data that is typically required by all SDR hardware is sent at the head of a packet. Settings that are specific to openHPSDR hardware are sent at the end of a packet. This leaves room in the middle of the packet for additional register settings to be included in the future without disturbing existing settings.

## OPENHPSDR HOST TO HARDWARE PROTOCOL

The Host will communicate with the Hardware using standard UDP protocol. Byte order shall be MSB first (little Endian) and all values are interpreted as unsigned integers unless otherwise noted.

IP and UDP headers are as per UDP/IP standards.

Key:

IP Header (24 bytes)
UDP Header (8 bytes)
UDP Data (variable bytes)

<div>Bits</div> <div>Bytes</div>	0-7	8-15	16-23	24-31	32- 39	40-47	48-55	56-63
0	Version/IHL	Type of Service	Total Length		Identification		Flags/Fragment Offset	
8	TTL	Protocol	Header Checksum		Source IP Address			
16	Destination IP Address				IP Options			Pad
24	Source Port		Destination Port		UDP Length		UDP Checksum	
32... 1443	UDP Data							

Future diagrams will only show the UDP Data with a starting byte reference of zero.



## DISCOVERY PACKET

A Discovery packet is broadcast from a Host in order to determine what SDRs are present on the network, and if so, how many. The format is as follows:

	Discovery - PC to Hardware	
Byte	Broadcast to Port 1024	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	0x02	Command - Discovery
5		Zero
6		Zero
7		Zero
8		Zero
.....		
59		Zero

### *Destination Address*

This will be Ethernet address 255.255.255.255.

### *Destination Port*

This will be 1024.

### *Sequence Number*

A 4-byte integer set to 0x00000000.

The hardware will respond with the relevant Command Reply Packet.

## ERASE PACKET

An Erase packet is sent from a Host in order to erase the EEPROM on the hardware prior to programming it. The format is as follows:

	Erase Command - PC to Hardware	
Byte	To IP Address of Hardware and Port 1024	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	0x04	Command - Erase
5	Zero	
6	Zero	
7	Zero	
8	Zero	
...		
59	Zero	

### *Destination Address*

This will be the Ethernet address assigned to the hardware.

### *Destination Port*

This will be 1024.

### *Sequence Number*

A 4-byte integer set to 0x00000000.

The hardware will respond with a Command Reply Packet confirming receipt of the Erase command and a subsequent reply when the erase has completed.

NOTE: Larger EEPROMS can take up to 15 seconds to erase. The Host program should include a timer such that if a reply has not been received within this time the user should be prompted to re-try.

## PROGRAM PACKET

A Program packet is sent from a Host in order to program the EEPROM on the hardware. The format is as follows:

	Program Command - PC to Hardware	
Byte	To IP Address of Hardware and Port 1024	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	0x05	Command - Program
5	# of blocks [31:24]	
6	# of blocks [23:16]	
7	# of blocks [15:8]	
8	# of blocks [7:0]	
9	Program data 0	
10	Program data 1	
11	Program data 2	
12	Program data 3	
13	Program data 4	
14	Program data 5	
15	Program data 6	
....		
264	Program data 255	

### *Destination Address*

This will be the Ethernet address assigned to the hardware.

### *Destination Port*

This will be 1024.

### *Sequence Number*

A 4-byte unsigned integer representing a sequence of packets from this port. The sequence starts at 0x00000000, increments for each new packet, and rolls over after exceeding 0xFFFFFFFF.

Each port decoder keeps a separate sequence count such that the hardware can determine if a decoder is missing samples in its stream. The sequence number is set to zero when initiating a Programming sequence.

### *Number of Blocks*

A 4 byte integer that indicates the total number of 256 byte blocks that will be sent from the Host to the SDR hardware.

The hardware will respond with a Command Reply Packet that requests the next block of 256 bytes be sent.

NOTE: Should the Sequence Number in the Command Reply Packet not be consistent with the sent Sequence Number then the Program process should be aborted and the user given the option to restart. An affirmative response should initiate an Erase sequence followed by a Program sequence.

## SET IP ADDRESS PACKET

A Set IP Address packet is sent from a Host in order to set the IP address of the hardware. Prior to setting an IP address a Discovery Packet should be sent and the MAC address of the hardware that's IP address is to be updated recorded.

The format is as follows:

Set IP Address - PC to Hardware		
Byte	Broadcast to Port 1024	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	0x03	Command - Set IP Address
5	ToMAC	MSB
6	ToMAC	
7	ToMAC	
8	ToMAC	
9	ToMAC	
10	ToMAC	LSB
11	AssignIP	MSB
12	AssignIP	
13	AssignIP	
14	AssignIP	LSB
15	Zero	
16	Zero	
17	Zero	
18	Zero	
....		
59	Zero	

### *Destination Address*

This will be Ethernet address 255.255.255.255.

### *Destination Port*

This will be 1024.

### *Sequence Number*

A 4-byte integer set to 0x00000000.

### *MAC Address*

A 6-byte value that contains the MAC address of the hardware that's IP address requires to be set.

### *IP Address*

A 4-byte value that contains the IP address that is required to be set in the hardware's EEPROM. This may be increased in the future to enable IPv6 addressing. Setting the IP address to 0.0.0.0 will force the hardware to use DHCP addressing.

NOTE: After the IP address has been saved in EEPROM, which will take a few seconds, the successful setting of the IP address can be verified by sending a Discovery Command. The reply from the hardware will include the new IP address.

## GENERAL PACKET TO SDR

The General packet contains data that sets infrequently changed settings of the transceiver subsystems. This data is sent when the UDP data changes and, optionally, periodically.

A General Packet should be sent following a successful Discovery exchange.

The format is as follows:

	Control Elements - PC to Hardware	
Byte	To IP Address of Hardware and Port 1024	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	0x00	Command
5	Rx Specific port [15:8]	Default Port # 1025
6	Rx Specific port [7:0]	
7	Tx Specific port [15:8]	Default Port # 1026
8	Tx Specific port [7:0]	
9	High Priority from PC port [15:8]	Default Port # 1027
10	High Priority from PC port [7:0]	
11	High Priority to PC port [15:8]	Default Port # 1025
12	High Priority to PC port [7:0]	
13	Receiver Audio port [15:8]	Default Port # 1028
14	Receiver Audio port [7:0]	
15	Tx0 I&Q port [15:8]	Base Port (Default Port # 1029)
16	Tx0 I&Q port [7:0]	
17	Rx0 port [15:8]	Base Port (Default Port # 1035), Rx1 = Base Port # + 1....Rx79 = Base Port # + 79
18	Rx0 port [7:0]	
19	Mic samples port [15:8]	Default Port # 1026
20	Mic samples port [7:0]	
21	Wideband ADC0 port [15:8]	Base Port (Default Port # 1027), ADC1 = Base Port # + 1....ADC7 = Base Port # + 7
22	Wideband ADC0 port [7:0]	
23	Wideband Enable [7:0]	WB0 = [0], WB1 = 1.....WB7 = [7]
24	Wideband Samples per packet [15:8]	Default 512
25	Wideband Samples per packet [7:0]	
26	Wideband sample size	Default 16 bits
27	Wideband update rate	Default 20 Frames/second
28	Memory mapped from PC port [15:8]	Default Port # xxxx
29	Memory mapped from PC port [7:0]	
30	Memory mapped to PC port [15:8]	Default Port # xxxx

31	Memory mapped to PC port [7:0]	
32	Envelope PWM_max	[15:8]
33	Envelope PWM_max	[7:0]
34	Envelope PWM_min	[15:8]
35	Envelope PWM_min	[7:0]
36	Bits - [0]Time stamp, [1]VITA-49, [2]VNA mode, [3] frequency or phase word	
37		Reserved for future use
..		Reserved for future use
56	Bits - Atlas bus configuration	[2:0] Configuration - see below
57	Bits - 10MHz ref source	[1:0] 10MHz reference source - see below
58	Bits - PA, Apollo, Mercury, clock source	[0] = PA, 1 = Apollo, [2] = Mercury Common Frequency, [3] Clock Source - see below
59	Bits - Alex(n) enable, 1= enable, 0 = disable	[0] = Alex 0....[7] = Alex7

#### *Source Port*

This will be set to 1024.

#### *Destination Port*

This will be set to the Source Port of the Host that initiated the Discovery Packet.

#### *Sequence Number*

A 4-byte integer set to 0x00000000.

#### *Bytes 5 & 6*

These two bytes form a 16 bit number that specifies the port that Receiver Specific commands will be sent to. If set to zero the default port 1025 will be used.

#### *Bytes 7 & 8*

These two bytes form a 16 bit number that specifies the port that Transmitter Specific commands will be sent to. If set to zero the default port 1026 will be used.

#### *Bytes 9 & 10*

These two bytes form a 16 bit number that specifies the port that High Priority commands will be sent to. If set to zero the default port 1027 will be used.

#### *Bytes 11 & 12*

These two bytes form a 16 bit number that specifies the port that High Priority commands from the hardware will be sent from. If set to zero the default port 1025 will be used.

#### *Bytes 13 & 14*

These two bytes form a 16 bit number that specifies the port that Receiver Audio will be sent to. If set to zero the default port 1028 will be used.



#### *Bytes 15 & 16*

These two bytes form a 16 bit number that specifies the port that transmitter I&Q data will be sent to. If set to zero the default port 1029 will be used.

#### *Bytes 17 & 18*

These two bytes form a 16 bit number that specifies the port that Receiver 0 I&Q data will originate from. If set to zero the default port 1035 will be used.. Each subsequent receiver will increment the port number respectively e.g. Receiver 1 will originate from Port + 1, Receiver 2 from Port +2.....Receiver 79 from Port + 79

#### *Bytes 19 & 20*

These two bytes form a 16 bit number that specifies the port that microphone or line in data will originate from. If set to zero the default port 1026 will be used.

#### *Bytes 21& 22*

These two bytes form a 16 bit number that specifies the port that wideband data from ADC0 will originate from. If set to zero the default port 1027 will be used.. Each subsequent wideband data will increment the port number respectively e.g. ADC1 1 will originate from Port + 1, ADC 2 from Port +2.....ADC7 from Port + 7

#### *Byte 23*

Enable wideband data. A set bit enables Wideband data from an associated ADC to be sent e.g. bit 0 enables ADC0, bit1 enables ADC1 etc.

#### *Bytes 24 & 25*

These two bytes form a 16 bit number that specifies the number of wideband samples to use per packet. The default is 512 by 16 bits samples.

#### *Byte 26*

Sets the size of a wideband sample in bits. If set to zero the default of 16 bits will be used.

#### *Byte 27*

Sets the update rate of the wideband data in mS. If set to zero the default of 20mS will be used.

#### *Bytes 28 & 29*

These two bytes form a 16 bit number that specifies the port that memory mapped data from the PC will be sent to. If set to zero the default port xxxx will be used.

#### *Bytes 30 & 31*

These two bytes form a 16 bit number that specifies the port that memory mapped data to the PC will be sent to. If set to zero the default port xxxx will be used.

#### *Bytes 32 & 33*

These two bytes form a 16 bit number that specifies the minimum pulse width for the Envelope Tracking PWM.

#### *Bytes 34 & 35*

These two bytes form a 16 bit number that specifies the maximum pulse width for the Envelope Tracking PWM.

#### *Byte 37*

Bits when set activate the following functions; Bit[0] – Enable time stamping of Receiver I&Q packets, Bit[1] - send data using VITA-49 format, Bit[2] – select VNA mode, Bit[3] selects if the receiver and transmitter

frequency data is sent as frequency ([3] = 0) (in Hz) or phase word (if [3] = 1). A phase word is calculated as follows:

$$\text{phase\_word}[31:0] = 2^{32} * \text{frequency(Hz)} / \text{DSP clock frequency (Hz)}$$

DSP clock frequency is returned in the Discovery reply.

**NOTE: Currently ALL openHPSDR FGPA code requires transmit and receive phase words.**

#### Byte 56

This selects the Atlas bus Mercury receiver configuration as follows:

Configuration (Mercury)
000 - single receiver
001 - two receivers
010 - three receivers
011 - four receivers

#### Byte 57

For Atlas based systems this selects the source of the 10MHz reference clock as follows:

10MHz reference source
00 = 10MHz reference from Atlas bus
01 = 10MHz reference from Penelope
10 = 10MHz reference from Mercury

#### Byte 58

For Atlas based system this selects the following items:

Bit[0] = PA, Bit[1] = Apollo, Bit[2] = Mercury Common Frequency, Bit [3] Clock Source - see below

Clock Source
0 = 122.88MHz source from Penelope
1 = 122.88MHz source from Mercury

#### Byte 59

Each bit enables the respective Alex filter board e.g. Bit[0] set enables Alex0, Bit[1] set enables Alex 1 etc.

## RECEIVER SPECIFIC PACKET

This sets the number of ADCs and the number of receivers associated with each ADC. It also sets receiver parameters that change infrequently. This packet is sent following a successful Discovery command and prior to a Run command or when a parameter changes and, optionally, periodically.

The format of the packet is as follows:

Control Elements - PC to Hardware		
Byte	Rx Specific	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Number of ADCs	Max of 8 ADCs
5	Bits - Dither ADC0...7	[0] = ADC0, [1] = ADC1....[7] = ADC7
6	Bits - Random ADC0..7	[0] = ADC0, [1] = ADC1....[7] = ADC7
7	Rx Enable Rx0....Rx7	[0] = Rx0, [1]= Rx1.....[7] = Rx7
8	Rx Enable Rx8....Rx15	
9	Rx Enable Rx16....Rx23	
10	Rx Enable Rx24....Rx31	
11	Rx Enable Rx32....Rx39	
12	Rx Enable Rx40....Rx47	
13	Rx Enable Rx48....Rx55	
14	Rx Enable Rx56....Rx63	
15	Rx Enable Rx64....Rx71	
16	Rx Enable Rx72....Rx79	[0] = Rx72.....[7] = Rx79
17	ADC Rx0	ADC(n) that Rx0 is allocated to
18	Sampling Rate Rx0	[15:8] 48/96/192/384/768/1536
19	Sampling Rate Rx	[7:0]
20	CIC1 Rx0	For Future use
21	CIC2 Rx0	For Future use
22	Sample Size Rx0	Default 24 bits
23	ADC Rx1	ADC(n) that Rx1 is allocated to.
24	Sampling Rate Rx1	[15:8]
25	Sampling Rate Rx1	[7:0]
26	CIC1 Rx1	
27	CIC2 Rx1	
28	Sample Size Rx1	
29	ADC Rx2	ADC(n) that Rx2 is allocated to.
30	Sampling Rate Rx2	[15:8]
31	Sampling Rate Rx2	[7:0]
32	CIC1 Rx2	

33	CIC2 Rx2	
34	Sample Size Rx2	
35	ADC Rx3	
36	Sampling Rate Rx3	[15:8]
37	Sampling Rate Rx3	[7:0]
38	CIC1 Rx3	
39	CIC2 Rx3	
40	Sample Size Rx3	
41	ADC Rx4	
42	Sampling Rate Rx4	[15:8]
43	Sampling Rate Rx4	[7:0]
44	CIC1 Rx4	
45	CIC2 Rx4	
46	Sample Size Rx4	
...		
...		
485	ADC Rx78	
486	Sampling Rate Rx78	
487	Sampling Rate Rx78	
488	CIC1 Rx78	
489	CIC2 Rx78	
490	Sample Size Rx78	
491	ADC Rx79	
492	Sampling Rate Rx79	
493	Sampling Rate Rx79	
494	CIC1 Rx79	
495	CIC2 Rx79	
496	Sample Size Rx79	
...		
...		
1363	SyncRx0	[7:0] If bit set then Rx(n) is synched or muxed to Rx0
1364	SyncRx1	[7:0] If bit set then Rx(n) is synched or muxed to Rx1
1365	SyncRx2	[7:0] If bit set then Rx(n) is synched or muxed to Rx2
...		
1442	SyncRx79	[7:0] If bit set then Rx(n) is synched or muxed to Rx79
1443	Mux	[7:0] If bit set then Rx(n) is in Multiplexed mode

#### *Destination Port*

This may be set using the General packet and if zero defaults to 1025.

#### *Sequence Number*

A 4-byte integer set to 0x00000000.

#### *Byte 4*

Indicates the number of ADC that the hardware supports. This will be up to four on Atlas based systems, one on Hermes (ANAN-10/10E/100) and two on Angelia and Orion (ANAN-100D/200D),

#### *Byte 5*

A set bit activates Dither on the associated ADC e.g. Bit[0] activates ADC0, Bit[1] activates ADC1 etc.

#### *Byte 6*

A set bit activates Random on the associated ADC e.g. Bit[0] activates ADC0, Bit[1] activates ADC1 etc.

#### *Byte 7*

A set bit enables the associated receiver e.g. Bit[0] activates Receiver 0, Bit[1] activates Receiver 1 etc.

#### *Bytes 8 to 16*

A set bit enables receivers 8 through 79.

#### *Byte 17*

Selects the ADC that Receiver 0 is connected to where 0 connects to ADC0, 1 to ADC1 etc.

#### *Bytes 18 and 19*

These two bytes from a 16 bit word that selects the sampling rate of Receiver 0. Valid rates are 48/96/192/384/768/1536 ksp/s.

#### *Bytes 20 & 21*

For future use – to enable selection of the decimation rates of Receiver 0 CIC filters.

#### *Byte 22*

Sets Receiver 0 I&Q data sample size – default is 24 bits.

#### *Bytes 23 to 496*

Sets the ADC, sampling rate, CIC rates and data sample size of receiver 1 to 79 as above.

#### *Bytes 1363 to 1442*

Sets the Receiver that Receiver (n) is synchronised or multiplexed with. If a bit is set then Rx(n) is synchronised or multiplexed to the associated receiver. See the description of synchronous and multiplexed receivers that follows.

#### *Byte 1443*

A set bit indicates that the receiver indicated by the set bit is multiplex with others. The receiver(s) to which it is multiplexed is indicated by set bits in the relevant Byte 497 to 1442.

#### *Synchronous and Multiplexed Receivers*

NOTE: The sampling rate of all Synchronous or Multiplexed receivers must be the same and is the responsibility of the PC Control program to ensure this.

Receivers that are connected to a base receiver, either Synchronised or Multiplexed, are usually disabled so they are not also sent from an Ethernet port. It is the responsibility of the PC Control program to ensure this.

The selection code will allow unsuitable or unnecessary receiver combinations e.g. Rx0 + Rx0 or Rx0 + Rx1 and Rx1 + Rx0. It is the responsibility of the PC Control program to prevent this.

There is no special provision for PureSignal operation. For PureSignal use, the PC Control program is responsible for setting the sampling rates, selecting DAC data as the source for one receiver and selecting either Synchronous or Multiplex operation of the RF and DAC receivers.

*Synchronous Receivers:* (where a number of receivers are phase synchronus)

The maximum number of synchronised receivers is equal to the number of ADCs.

For synchronous receivers, if SyncRx[n] is > 0 then Rx[n] is synchronised with another receiver(s). The bit(s) set indicate which receiver(s) are synchronised e.g. bit[0] = Rx0, bit[1] = Rx1.....bit[7] = Rx7.

All receivers frequencies will be set to the frequency of the base receiver. If SyncRx[n] is = 0 then there are no synchronous receivers selected.

*Multiplexed Receivers:* (where a number of receivers are multiplexed over the one Ethernet port and may or may not be at a common frequency)

The maximum number of Multiplexed receivers is equal to the number of receivers.

For multiplexed receivers if Mux(n) bit n is set then Rx(n) is multiplexed with another receiver(s). SyncRx[n] bits [7:0] when set indicate which receiver(s) are multiplexed together e.g. bit[0] = Rx0, bit[1] = Rx1.....bit[7] = Rx7.

## TRANSMITTER SPECIFIC PACKET

This sets the number of DACs and transmitter parameters that change infrequently. This packet is sent following a successful Discovery command and prior to a Run command or when a parameter changes and, optionally, periodically.

The format of the packet is as follows:

Control Elements - PC to Hardware		
Byte	Tx Specific	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Number of DACs	Max of 4
5	Bits - Mode, CW, Reverse, Key Mode	See Below
6	Sidetone Level	
7	Sidetone Frequency (Hz)	[15:8]
8	Sidetone Frequency (Hz)	[7:0]
9	Keyer Speed	
10	Keyer Weight	
11	Hang delay	[15:8]
12	Hang delay	[7:0]
13	RF Delay	
14	Tx0 Sampling Rate	[15:8]
15	Tx0 Sampling Rate	[7:0]
16	Tx0 Bits	
17..25		Reserved for future use
26	Tx0 Phase Shift (0 - 359 degrees)	[15:8]
27	Tx0 Phase Shift	[7:0]
28..33		Reserved for future use
...		
...		
50	Bits - line in, mic boost, Orion mic	See Below
51	Line in gain	
52..58		Reserved for future use
59	Step Attenuator ADC0 on Tx0 (0 - 31dB)	Reserved for future use

### Destination Port

This may be set using the General packet and if zero defaults to 1026.

#### *Sequence Number*

A 4-byte integer set to 0x00000000.

#### *Byte 4*

Indicates the number of DACs the hardware supports. Presently unused.

#### *Byte 5*

If no bits are set then CW is not selected, otherwise indicates the selection of CW options as follows:

Bits - Mode, CW, Reverse, Key Mode. 0 = off, 1 = on.	
[0]	EER
[1]	CW
[2]	Reverse CW Keys
[3]	Iambic
[4]	Sidetone
[5]	Mode B (Mode A if not set)
[6]	Strict Character Spacing
[7]	Break_in

#### *Byte 6*

Sets the CW sidetone level, 0 = off, 255 = max

#### *Bytes 7 & 8*

Sets the CW sidetone frequency in Hz

#### *Byte 9*

Sets the CW keyer speed, 0 to 60 WPM

#### *Byte 10*

Sets the CW weight, 33 to 66, nominal is 50

#### *Bytes 11 & 12*

Sets the CW hang delay in mS

#### *Byte 13*

Sets the RF delay in mS

#### *Bytes 14 & 15*

Sets Tx0 sampling rate. For current hardware fixed at 192ksps.

#### *Byte 16*

Sets number of bits in the Tx I&Q data. For current hardware fixed at 24 bits per sample.

#### *Bytes 17 to 49.*

Reserved for future use.



#### Byte 50

Allows the selection of Line in or Microphone and Microphone selection for an Orion board (ANAN-200D) as follows:

Bits - line in, mic boost, Orion mic. 0 = off, 1 = on
[0] = Line in
[1] = Mic Boost
[2] = 0 = Orion mic PTT enabled, 1 = Orion mic PTT disabled
[3] = 0 = Orion mic PTT to ring and mic/mic bias to tip, 1 = Orion mic PTT to tip and mic/mic bias to ring
[4] = 0 = disables Orion mic bias, 1 = enables Orion microphone bias

## TRANSMITTER SYNCHRONISATION PACKET

The Transmitter Synchronisation packet is used to configure synchronous transmitters.

The data is sent prior to a Run command, whenever a value changes and, optionally, periodically.

For future use.

## HIGH PRIORITY DATA PACKET

A High Priority Packet is sent to the associated SDR hardware whenever data changes and may also be sent periodically. It should be sent at a higher priority than any other packet. It should be sent after a successful Discovery process after configuration is complete. The format is as follows:

Control Elements - PC to Hardware Port (Default 1027)		
Byte	High Priority	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Bits - run, PTT(n)	[0] = run, [1] = PTT0...[4] = PTT3
5	CWX0	[0] = CWX, [1] = Dot, [2] = Dash
6	CWX1	Reserved for future use
7	CWX2	Reserved for future use
8	CWX3	Reserved for future use
9	Frequency/phase word Rx0	[31:24]
10		[23:16]
11		[15:8]
12		[7:0]
13	Frequency/phase word Rx1	[31:24]
14		[23:16]
15		[15:8]
16		[7:0]
17	Frequency/phase word Rx2	[31:24]
18		[23:16]
19		[15:8]
20		[7:0]
...		
...		
325	Frequency/phase word Rx79	[31:24]
326		[23:16]
327		[15:8]
328		[7:0]
329	Frequency/phase word Tx0	[31:24]
330	Tx0	[23:16]
331	Tx0	[15:8]
332	Tx0	[7:0]
333	Frequency/phase word Tx1	Reserved for future use
334	Tx1	Reserved for future use
335	Tx1	Reserved for future use

336	Tx1	Reserved for future use
337	Frequency/phase word Tx2	Reserved for future use
338	Tx2	Reserved for future use
339	Tx2	Reserved for future use
340	Tx2	Reserved for future use
341	Frequency/phase word Tx3	Reserved for future use
342	Tx3	Reserved for future use
343	Tx3	Reserved for future use
344	Tx3	Reserved for future use
345	Tx0 Drive Level	0-255
346	Tx1 Drive Level	Reserved for future use
347	Tx2 Drive Level	Reserved for future use
348	Tx3 Drive Level	Reserved for future use
...		
...		
1417	Open Collector Outputs	[1] = 0 .....[7] = 7
1418	User Outputs DB9 pins 1-4	[0] = pin1....[3] = pin4
1419	Mercury Attenuator (20dB)	[0] = Mercury1....[3] = Mercury4
1420	Alex 3	Reserved for future use
1421	Alex 3	Reserved for future use
1422	Alex 3	Reserved for future use
1423	Alex 3	Reserved for future use
1424	Alex 2	Reserved for future use
1425	Alex 2	Reserved for future use
1426	Alex 2	Reserved for future use
1427	Alex 2	Reserved for future use
1428	Alex 1	Reserved for future use
1429	Alex 1	Reserved for future use
1430	Alex 1	Reserved for future use
1431	Alex 1	Reserved for future use
1432	Alex 0	[31:24]
1433	Alex 0	[23:16]
1434	Alex 0	[15:8]
1435	Alex 0	[7:0]
1436	Step Attenuator 7 (0 - 31dB)	Reserved for future use
1437	Step Attenuator 6 (0 - 31dB)	Reserved for future use
1438	Step Attenuator 5 (0 - 31dB)	Reserved for future use
1439	Step Attenuator 4 (0 - 31dB)	Reserved for future use
1440	Step Attenuator 3 (0 - 31dB)	Reserved for future use
1441	Step Attenuator 2 (0 - 31dB)	Reserved for future use
1442	Step Attenuator 1 (0 - 31dB)	
1443	Step Attenuator 0 (0 - 31dB)	

#### *Destination Address*

This will be the Ethernet address assigned to the hardware.

#### *Destination Port*

This may be set using the General packet and if zero defaults to 1027.

#### *Sequence Number*

A 4-byte integer set to 0x00000000.

#### *Byte 4*

Bit [0] when set enables the associated SDR hardware and when clear disables. Bit [1] enables transmit of the associated SDR hardware. Bits [2] to [7] are reserved for future use.

#### *Byte 5*

Bit[0] when set selects CW mode from the Host (e.g. a CW keyboard) with bit[1] and[2] being dot and dash respectively. A set bit will send a dot or dash at the speed selected by the Tx Specific packet.

#### *Bytes 9 to 12*

These bytes represent a 32bit word that is used to set the frequency of Rx0 in Hz.

#### *Bytes 13 to 16*

These bytes represent a 32bit word that is used to set the frequency of Rx1 in Hz.

#### *Bytes 17 to 328*

These bytes represent a 32bit word that is used to set the frequency of Rx2 to Rx79.

#### *Bytes 329 to 332*

These bytes represent a 32bit word that is used to set the frequency of Tx0 in Hz.

#### *Byte 345*

This byte sets the power out from Tx0. 0 represents 0 power out and 255 maximum.

#### *Byte 1417*

A set bit enables the associated open collector output. [0] = OC0, [1] = OC2.....[6] = OC6.

#### *Byte 1418*

A set bit enables the associated Atlas Metis board User Outputs DB9 connector pins 1-4. [0] = pin1....[3] = pin4

#### *Byte 1419*

A set bit enables the 20dB attenuate on the associated Atlas Mercury board. [0] = Mercury1....[3] = Mercury4.

#### *Bytes 1432 to 1435*

These bytes form a 32bit word that selects the various functions on the (Alex) High and Low pass filters, preamplifier and antenna switching. The functions of each bit are as follows:

Data to send to Alex Rx filters is in the following format:		
Bit	Function	I.C. Output
0	YELLOW LED	U2 - D0
1	13 MHz HPF	U2 - D1
2	20 MHz HPF	U2 - D2
3	6M Preamp	U2 - D3
4	9.5 MHz HPF	U2 - D4
5	6.5 MHz HPF	U2 - D5
6	1.5 MHz HPF	U2 - D6
7	N.C.	U2 - D7
8	XVTR RX In	U3 - D0
9	RX 2 In	U3 - D1
10	RX 1 In	U3 - D2
11	RX 1 Out	U3 - D3
12	Bypass	U3 - D4
13	20 dB Atten.	U3 - D5
14	10 dB Atten.	U3 - D6
15	RED LED	U3 - D7

Data to send to Alex Tx filters is in the following format:		
Bit	Function	I.C. Output
16	N.C.	U2 - D0
17	N.C.	U2 - D1
18	N.C.	U2 - D2
19	YELLOW LED	U2 - D3
20	30/20 Meters LPF	U2 - D4
21	60/40 Meters LPF	U2 - D5
22	80 Meters LPF	U2 - D6
23	160 Meters LPF	U2 - D7
24	ANT #1	U4 - D0
25	ANT #2	U4 - D1
26	ANT #3	U4 - D2
27	T/R Relay	U4 - D3
28	RED LED	U4 - D4
29	6 Mtrs(Bypass)	U4 - D5
30	12/10 Meters LPF	U4 - D6
31	17/15 Meters LPF	U4 - D7

Where all bits are active high and bit 27 high selects transmit.

*Byte 1442*

Selects the attenuation applied to the 0-31dB attenuator before ADC1 (Angelia, Orion, ANAN-100D, ANAN-200D only).

*Byte 1443*

Selects the attenuation applied to the 0-31dB attenuator before ADC0 (not Mercury boards).

## RECEIVER AUDIO PACKET

The Receiver Audio packet contains left and right audio to be presented to the hardware audio DAC.

It is sent whenever 360 Left and Right audio samples are available. Data defaults to 16 bits per sample at 48ksps.

	Rx Audio Data (default port 1028)	
Byte	Data	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Left Audio Sample 0	[15:8]
5	Left Audio Sample 0	[7:0]
6	Right Audio Sample 0	[15:8]
7	Right Audio Sample 0	[7:0]
8	Left Audio Sample 1	[15:8]
9	Left Audio Sample 1	[7:0]
10	Right Audio Sample 1	[15:8]
11	Right Audio Sample 1	[7:0]
....		
....		
1440	Left Audio Sample 358	[15:8]
1441	Left Audio Sample 358	[7:0]
1442	Right Audio Sample 359	[15:8]
1443	Right Audio Sample 359	[7:0]

### *Destination Port*

This may be set using the General packet and if zero defaults to 1028.

### *Sequence Number*

A 4-byte unsigned integer representing a sequence of packets from this port. The sequence starts at 0x00000000 and rolls over after exceeding 0xFFFFFFFF.

Each port decoder keeps a separate sequence count such that the hardware can determine if a decoder is missing samples in its stream. The sequence number is set to zero at program start or whenever a stop command is sent.

### *Bytes 4 to 144*

A sequence of 360 2-byte signed integer values representing demodulated samples for the audio DAC on the hardware. These are sequential in time from the first to last sample.



## TRANSMITTER I&Q DATA PACKET

The I & Q data packet contains data to be presented to the hardware DUC(n). It is sent whenever 480 I&Q samples are available. Samples are 24 bits at 192ksps.

	<b>Tx I&amp;Q Data (default port 1029)</b>	
<b>Byte</b>	<b>Data</b>	<b>Notes</b>
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	I Sample 0	[23:16]
5	I Sample 0	[15:8]
6	I Sample 0	[7:0]
7	Q Sample 0	[23:16]
8	Q Sample 0	[15:8]
9	Q Sample 0	[7:0]
10	I Sample 1 (or Envelope data)	[23:16]
11	I Sample 1 (or Envelope data)	[15:8]
12	I Sample 1 (or Envelope data)	[7:0]
13	Q Sample 1 (or Envelope data)	[23:16]
14	Q Sample 1 (or Envelope data)	[15:8]
15	Q Sample 1 (or Envelope data)	[7:0]
...		
...		
1441	Q Sample 479	[23:16]
1442	Q Sample 479	[15:8]
1443	Q Sample 479	[7:0]

### *Destination Port*

This may be set using the General packet and if zero defaults to 1029 for DAC0. It is automatically incremented by one for each subsequent DAC.

### *Sequence Number*

A 4-byte unsigned integer representing a sequence of packets from this port. The sequence starts at 0x00000000 and rolls over after exceeding 0xFFFFFFFF.

Each port decoder keeps a separate sequence count such that the hardware can determine if a decoder is missing samples in its stream. The sequence number is set to zero at program start or whenever a stop command is sent.

### *I/Q Sample*

A sequence of 480 2-byte signed integer I & Q values representing samples for the DUC on the SDR hardware. These are sequential in time from the first to last sample.

NOTE: When Envelope Tracking (ET) or Envelope Elimination & Restoration (EER) mode is selected then alternative I&Q pairs are for Transmitter and Envelope use respectively.

## MEMORY MAPPED REGISTERS FROM PC

The memory mapped registers from PC packet contains address and data to be sent to the hardware. The format is as follows:

Memory Mapped - PC to hardware (default port xxxx)		
Byte	Memory Mapped	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Address0	[15:8]
5	Address0	[7:0]
6	Data0	[31:24]
7	Data0	[23:16]
8	Data0	[15:8]
9	Data0	[7:0]
10	Address1	[15:8]
11	Address1	[7:0]
12	Data1	[31:24]
13	Data1	[23:16]
14	Data1	[15:8]
15	Data1	[7:0]
16	Address2	[15:8]
17	Address2	[7:0]
18	Data2	[31:24]
19	Data2	[23:16]
20	Data2	[15:8]
21	Data2	[7:0]
22	Address3	[15:8]
23	Address3	[7:0]
24	Data3	[31:24]
25	Data3	[23:16]
26	Data3	[15:8]
27	Data3	[7:0]
...		
1438	Address239	[15:8]
1439	Address239	[7:0]
1440	Data239	[31:24]
1441	Data239	[23:16]
1442	Data239	[15:8]
1443	Data239	[7:0]

#### *Destination Port*

This may be set using the General packet and if zero defaults to xxxx.

#### *Sequence Number*

A 4-byte unsigned integer representing a sequence of packets from this port. The sequence starts at 0x00000000 and rolls over after exceeding 0xFFFFFFFF.

Each port decoder keeps a separate sequence count such that the hardware can determine if a decoder is missing samples in its stream. The sequence number is set to zero at program start or whenever a stop command is sent.

#### *Bytes 4 & 5 (en sequence)*

These form the 16 bit address the 32 bit data in the next 4 bytes will be sent to.

#### *Bytes 6 to 9 (en sequence)*

This form a 32 bit data word that will be sent to the address specified by the address bytes.

## OPENHPSDR HARDWARE TO HOST PROTOCOL

The hardware will communicate with the Host using standard UDP protocol. Byte order shall be MSB first (little Endian) and all values are interpreted as unsigned integers unless otherwise noted.

IP and UDP headers are as per UDP/IP standards.

Key:

IP Header (24 bytes)
UDP Header (8 bytes)
UDP Data (variable bytes)

<div>Bits</div> <div>Bytes</div>	0-7	8-15	16-23	24-31	32- 39	40-47	48-55	56-63
0	Version/IHL	Type of Service	Total Length		Identification		Flags/Fragment Offset	
8	TTL	Protocol	Header Checksum		Source IP Address			
16	Destination IP Address				IP Options			Pad
24	Source Port		Destination Port		UDP Length		UDP Checksum	
32... 1443	UDP Data							

Future diagrams will only show the UDP Data with a starting byte reference of zero.

## DISCOVERY REPLY PACKET

A Discovery Reply Packet is sent in response to a Host broadcasting a Discovery packet. The Host sends a discovery packet in order to determine what hardware is present on the network and how it is configured.

In order to allow for SDR hardware to provide complete details of all features available two reply formats are provided.

1. Where prior knowledge of the hardware is known by the Host software, and
2. Where the full description of the configuration of the hardware and its capabilities are provided in the response to the Discovery packet.

The data format for option 1 is shown below.

Response to Discovery - Hardware to Host ( from port 1024)		
Byte	To IP address and Port of Host PC	Notes
0	Seq # = 0	[31:24]
1	Seq # = 0	[23:16]
2	Seq # = 0	[15:8]
3	Seq # = 0	[7:0]
4	0x02 (or 0x03)	Response to Discovery
5	Board MAC	MSB
6	Board MAC	
7	Board MAC	
8	Board MAC	
9	Board MAC	
10	Board MAC	LSB
11	Code Version	
12	Board Type	See Below
13	Mercury 0	
14	Mercury 1	
15	Mercury 2	
16	Mercury 3	
17	Penny	
18	Metis	
19	DSP clock frequency (Hz)	[31:24]
20	DSP clock frequency (Hz)	[23:16]
21	DSP clock frequency (Hz)	[15:8]
22	DSP clock frequency (Hz)	[7:0]
23	Number of receivers implemented	
....		
59	0	

#### *Source Port*

This will be set to 1024.

#### *Destination Port*

This will be set to the Source Port of the Host that initiated the Command.

#### *Sequence Number*

A 4-byte integer set to 0x00000000.

#### *Byte 4*

Normally 0x02 but if the hardware is running and already connected to a different Host it will be 0x03.

If byte 4 is 0xFF then subsequent bytes contain the hardware configuration of the SDR. Appendix A describes the format of the hardware configuration.

#### *MAC*

This is 6 bytes and holds the MAC address of the hardware that is responding to the Command request

#### *Code Version*

A byte that indicates the version of code loaded into the associated board(s). The Host interprets this as a decimal number e.g. 104 would be interpreted as version 10.4

#### *Board Type*

A byte interpreted as follows:

0: board = "ATLAS"
1: board = "HERMES" (ANAN-10, 10E, 100)
2: board = "GRIFFIN"
3: board = "ANGELIA" (ANAN-100D)
4: board = "ORION" (ANAN-200D)
5: board = Not presently allocated
6: board = "Hermes Lite"
7....254 reserved for future hardware

#### *Bytes 13 to 18*

Where an Atlas based system is identified, bytes 13 to 18 contain the code versions loaded into the respective boards. A version number of zero indicates a board is not present.

#### *Bytes 19 to 22*

These contain the DSP clock frequency that the hardware is running at and can be used by the PC software to determine parameters that require this to be known e.g. frequency to phase word conversion.

#### *Byte 23*

This holds the number of independent receivers that is implemented in the hardware.

## COMMAND REPLY PACKET

A Command reply packet is returned to the Host making the Command request. The format of the reply is the same as a Discovery reply, up to and including byte 18, except that byte 4 (and for Program the sequence number) is different.

The reply is returned in response to the following:

*Erase.* Following an EEPROM erase command the hardware will respond with a reply packet that indicates the command has been received and, after the erase is complete – which may take 10's of seconds, that the erase has been successful. Byte 4 will be set to 0x03.

*Program.* Following a Program command the hardware will respond with a reply packet that indicates that the next 256 bytes of data should be sent by the Host. Byte 4 will be set to 0x04 and the sequence number will be set to the last sequence number received from the Host.

During programming the Host should check the received sequence number in order to verify that a programming packet has been received and in the correct sequence. An incorrect sequence number should be used by the Host to initiate a new Erase and Program cycle.

*Set IP.* Following a successful Set IP command the hardware will respond with a Discovery reply packet.



## HIGH PRIORITY STATUS PACKET

The High Priority Status packet contains data that indicates the status of the transceiver subsystems. This data is sent when the data changes and, optionally, periodically.

Any change of Status data will be sent as a priority and has precedence over any other packet.

Since actual status changes will be hardware specific the format will vary between different SDR hardware.

The packet format is as follows:

Control Elements - Hardware to PC (default port 1025)		
Byte	High Priority	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Bits - PTT, Dot, Dash	[0] = PTT, [1] = Dot, [2] = Dash
5	Bits - ADC Overload	[0] = ADC0...[7] = ADC7
6	Exciter Power 0	[15:8]
7	Exciter Power 0	[7:0]
8	Exciter Power 1	[15:8] Reserved for future use
9	Exciter Power 1	[7:0] Reserved for future use
10	Exciter Power 2	[15:8] Reserved for future use
11	Exciter Power 2	[7:0] Reserved for future use
12	Exciter Power 3	[15:8] Reserved for future use
13	Exciter Power 3	[7:0] Reserved for future use
14	Forward Power - Alex 0	[15:8] (Set to zero if Alex not selected)
15	Forward Power - Alex 0	[7:0] (Set to zero if Alex not selected)
16	Forward Power - Alex 1	[15:8] Reserved for future use
17	Forward Power - Alex 1	[7:0] Reserved for future use
18	Forward Power - Alex 2	[15:8] Reserved for future use
19	Forward Power - Alex 2	[7:0] Reserved for future use
20	Forward Power - Alex 3	[15:8] Reserved for future use
21	Forward Power - Alex 3	[7:0] Reserved for future use
22	Reverse Power - Alex 0	[15:8] (Set to zero if Alex not selected)
23	Reverse Power - Alex 0	[7:0] (Set to zero if Alex not selected)
24	Reverse Power - Alex 1	[15:8] Reserved for future use
25	Reverse Power - Alex 1	[7:0] Reserved for future use
26	Reverse Power - Alex 2	[15:8] Reserved for future use
27	Reverse Power - Alex 2	[7:0] Reserved for future use
28	Reverse Power - Alex 3	[15:8] Reserved for future use
29	Reverse Power - Alex 3	[7:0] Reserved for future use
30	0	Presently unused

...		
48	0	Presently unused
49	Supply Volts	[15:8]
50	Supply Volts	[7:0]
51	User ADC3	[15:8]
52	User ADC3	[7:0]
53	User ADC2	[15:8]
54	User ADC2	[7:0]
55	User ADC1	[15:8]
56	User ADC1	[7:0]
57	User ADC0	[15:8]
58	User ADC0	[7:0]
59	Bits - User logic inputs	[0] = IO0....[7] = IO7

#### *Source Port*

This may be set using the General packet and if zero defaults to 1025.

#### *Destination Port*

This will be set to the Source Port of the Host that initiated the Discovery Packet.

#### *Sequence Number*

A 4-byte integer set to 0x00000000. The sequence starts at 0x00000000 and rolls over after exceeding 0xFFFFFFFF. Each port decoder keeps a separate sequence count such that the Host can determine if a decoder is missing samples in its stream. The sequence number is set to zero at power on or whenever a stop command is received.

#### *Byte 4*

Bits [0] to [2] indicate the status of an attached PTT button or CW key. A bit is set when the input is active.

#### *Byte 5*

Bits [0] to [7] are set if the associated ADC[n] overload bit is set.

#### *Bytes 6 & 7*

These form a 16 bit word that represents the forward power from the exciter (or Penelope or Penny Lane boards). The conversion to power (mW) is to multiply the word by ????

#### *Bytes 14 & 15*

These form a 16 bit word that represents the forward power from the exciter Power Amplifier. The conversion to power (Watts) is to multiply the word by 0.0168.

#### *Bytes 22 & 23*

These form a 16 bit word that represents the reverse power from the exciter Power Amplifier. The conversion to power (Watts) is to multiply the word by 0.0168.

#### *Bytes 49 & 50*

These form a 16 bit word that represents the supply voltage applied to the hardware (not Atlas bus systems). The conversion to Volts is to multiply the word by 0.00525.

*Bytes 51 to 58*

These form 16 bit words that represent the general purpose analogue voltage inputs to the hardware. These are presently undefined since they are user specific.

*Byte 59*

Bits [0] to [7] are set if the associated general purpose digital input is set. These are presently undefined since they are user specific.

## MICROPHONE DATA PACKET

The microphone data packet contains 720 samples from the microphone or line-in inputs on the hardware.

The sample rate is 48kps and packet format is as follows:

Microphone Data (default port 1026)		
Byte	Data	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Mic Sample 0	[15:8]
5	Mic Sample 0	[7:0]
6	Mic Sample 1	[15:8]
7	Mic Sample 1	[7:0]
8	Mic Sample 2	[15:8]
9	Mic Sample 2	[7:0]
10	Mic Sample 3	[15:8]
11	Mic Sample 4	[7:0]
....		
....		
1440	Mic Sample 718	[15:8]
1441	Mic Sample 718	[7:0]
1442	Mic Sample 719	[15:8]
1443	Mic Sample 719	[7:0]

### *Source Port*

This may be set using the General packet and if zero defaults to 1026.

### *Destination Port*

This will be set to the Source Port of the Host that initiated the Command & Control Packet.

### *Sequence Number*

A 4-byte unsigned integer representing a sequence of packets from this port. The sequence starts at 0x00000000 and rolls over after exceeding 0xFFFFFFFF.

Each port decoder keeps a separate sequence count such that the Host can determine if a decoder is missing samples in its stream. The sequence number is set to zero at power on or whenever a stop command is received.

### *Mic Sample*

A sequence of 720 2-byte signed integer values representing samples from the microphone or line-in on the hardware. These are sequential in time from the first to last sample.

## WIDEBAND DATA PACKET

The Wideband data packet contains raw samples from the ADCs selected by the user. The number of bits per sample, number of samples per packet and update rate specified in the General to Hardware settings will be sent when the Wideband enable bit for the associated ADC is set.

The following assumes 512 by 16 bit samples per packet.

Wideband Data (default port 1027)		
Byte	Data	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Wideband Sample 0	[15:8]
5	Wideband Sample 0	[7:0]
6	Wideband Sample 1	[15:8]
7	Wideband Sample 1	[7:0]
8	Wideband Sample 2	[15:8]
9	Wideband Sample 2	[7:0]
10	Wideband Sample 3	[15:8]
11	Wideband Sample 3	[7:0]
....		
....		
1024	Wideband Sample 510	[15:8]
1025	Wideband Sample 510	[7:0]
1026	Wideband Sample 511	[15:8]
1027	Wideband Sample 511	[7:0]

### Source Port

This may be set using the General packet and if zero defaults to 1027 for ADC0. The source port is automatically incremented by one for each subsequent Wideband data source.

### Destination Port

This will be set to the Source Port of the Host that initiated the Command & Control Packet.

### Sequence Number

A 4-byte unsigned integer representing a sequence of packets from this port. The sequence starts at 0x00000000 and rolls over after exceeding 0xFFFFFFFF.

Each port decoder keeps a separate sequence count such that the Host can determine if a decoder is missing samples in its stream. The sequence number is set to zero at power on or whenever a block of raw samples has been sent or whenever a stop command is received.

### ADC Sample

A sequence of 2-byte signed integer values (default is 512) representing raw samples from the selected ADC

on the hardware. These are sequential in time from the first to last sample. Following a request to send wideband data, the sequence number will be set to zero and incremented in each sequential packet until the number of samples set in the General packet has been sent. The sequence number is reset to zero for the next packet or when a stop command is received.

## RECEIVER PACKET

I & Q data from a DDC connected to an ADC will be sent from a UDP port with bits per sample specified by the General settings. I & Q data from either a single receiver or multiple synchronous or multiplexed receivers can be presented.

The following assumes one receiver using 24 bits per sample.

Receiver I&Q Data ( Rx0 default port 1035)		
Byte	Data	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Time Stamp	[63:56]
5	Time Stamp	[55:48]
6	Time Stamp	[47:40]
7	Time Stamp	[39:32]
8	Time Stamp	[31:24]
9	Time Stamp	[23:16]
10	Time Stamp	[15:8]
11	Time Stamp	[7:0]
12	Bits per sample	[15:8]
13	Bits per sample	[7:0]
14	I&Q Samples per frame	[15:8]
15	I&Q Samples per frame	[7:0]
16	I Sample 0	[23:16]
17	I Sample 0	[15:8]
18	I Sample 0	[7:0]
19	Q Sample 0	[23:16]
20	Q Sample 0	[15:8]
21	Q Sample 0	[7:0]
22	I Sample 1	[23:16]
23	I Sample 1	[15:8]
24	I Sample 1	[7:0]
25	Q Sample 1	[23:16]
26	Q Sample 1	[15:8]
27	Q Sample 1	[7:0]
...		
...		
1441	Q Sample 327	[23:16]
1442	Q Sample 327	[15:8]
1443	Q Sample 327	[7:0]

#### *Source Port*

The receiver port will be as allocated by the General Packet and if zero defaults to 1035 for Receiver 0. Each additional receiver will be allocated the next sequential port number e.g. Receiver 1 will originate from port 1036.

#### *Destination Port*

This will be set to the Source Port of the Host that initiated the Discovery Command

#### *Sequence Number*

A 4-byte unsigned integer representing a sequence of packets from this port. The sequence starts at 0x00000000 and rolls over after exceeding 0xFFFFFFFF.

Each port decoder keeps a separate sequence count such that the Host can determine if a decoder is missing samples in its stream. The sequence number is set to zero at power on or whenever a stop command is received.

#### *Time stamp*

This is a 64 bit unsigned integer that is incremented at the sample rate of the ADC from which the data is being generated and is set to zero at the 0 to 1 transition of the 1 PPS from a GPS receiver. This complies with the 'Fractional -Seconds Timestamp - The Sample Count Timestamp' of the VITA-49 specification section 6.1.5.1.

#### *Bits per sample*

A two byte unsigned integer that indicates the number of bits per sample. Current hardware FPGA code supports 24 bits per sample.

#### *I&Q Samples per frame*

A two byte unsigned integer that indicates the number of samples that follow. The number of samples will not exceed the maximum length of the data payload for a UDP packet and will contain an integer number of I&Q samples. Where synchronous or multiplexed receiver data is presented then the UDP packet will contain an integer number of samples e.g. for two synchronous receivers [(I0, Q0), (I1, Q1)] x 164.

The data diagram above illustrates the case for one receiver. Should a greater number of synchronous or multiplexed receivers be used the above data format will be modified accordingly. The number of samples per UDP frame will depend on the width of each I & Q sample i.e. 8, 16, 24, or 32 bits.

#### *Receiver I/Q Samples*

This is a sequence of signed integer values representing one I or one Q sample from the receiver hardware. These are sequential in time from the first to last sample.

For synchronous or multiplexed receivers the sample rate of each receiver will be the same and specified by the register settings set using the Receiver Specific packet.



## MEMORY MAPPED REGISTERS FROM HARDWARE

The memory mapped registers from PC packet contains address and data to be sent to the PC. The data should be sent whenever data changes and optionally periodically. The format is as follows:

Memory Mapped - Hardware to PC (default port xxxx)		
Byte	Memory Mapped	Notes
0	Seq #	[31:24]
1	Seq #	[23:16]
2	Seq #	[15:8]
3	Seq #	[7:0]
4	Address0	[15:8]
5	Address0	[7:0]
6	Data0	[31:24]
7	Data0	[23:16]
8	Data0	[15:8]
9	Data0	[7:0]
10	Address1	[15:8]
11	Address1	[7:0]
12	Data1	[31:24]
13	Data1	[23:16]
14	Data1	[15:8]
15	Data1	[7:0]
16	Address2	[15:8]
17	Address2	[7:0]
18	Data2	[31:24]
19	Data2	[23:16]
20	Data2	[15:8]
21	Data2	[7:0]
22	Address3	[15:8]
23	Address3	[7:0]
24	Data3	[31:24]
25	Data3	[23:16]
26	Data3	[15:8]
27	Data3	[7:0]
...		
1438	Address239	[15:8]
1439	Address239	[7:0]
1440	Data239	[31:24]
1441	Data239	[23:16]
1442	Data239	[15:8]
1443	Data239	[7:0]

#### *Destination Port*

This may be set using the General packet and if zero defaults to xxxx.

#### *Sequence Number*

A 4-byte unsigned integer representing a sequence of packets from this port. The sequence starts at 0x00000000 and rolls over after exceeding 0xFFFFFFFF.

Each port decoder keeps a separate sequence count such that the hardware can determine if a decoder is missing samples in its stream. The sequence number is set to zero at program start or whenever a stop command is sent.

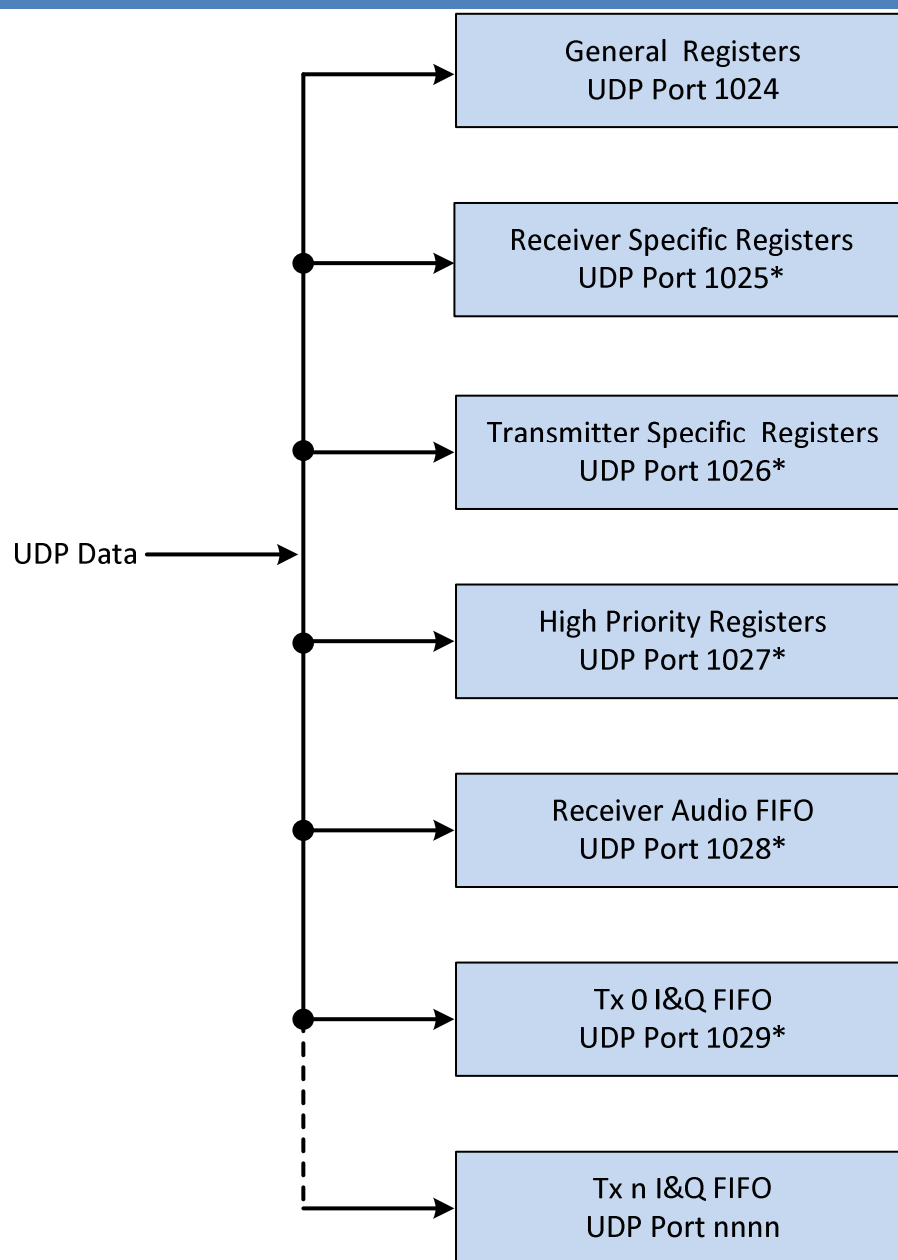
#### *Bytes 4 & 5 (en sequence)*

These form the 16 bit address the 32 bit data in the next 4 bytes will be sent from.

#### *Bytes 6 to 9 (en sequence)*

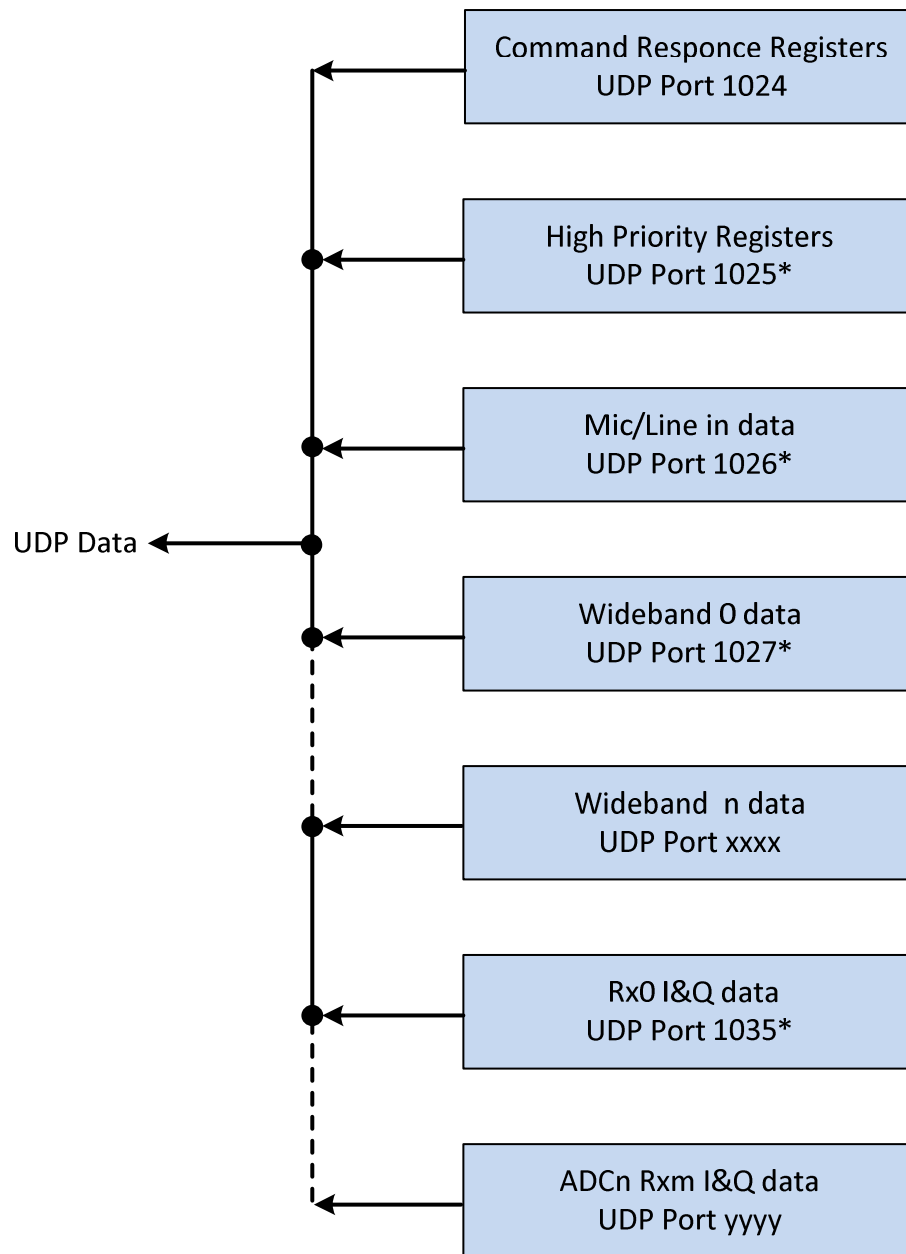
This form a 32 bit data word that is the data contained at the address specified by the address bytes.

FIGURE 1 UDP DATA FROM HOST



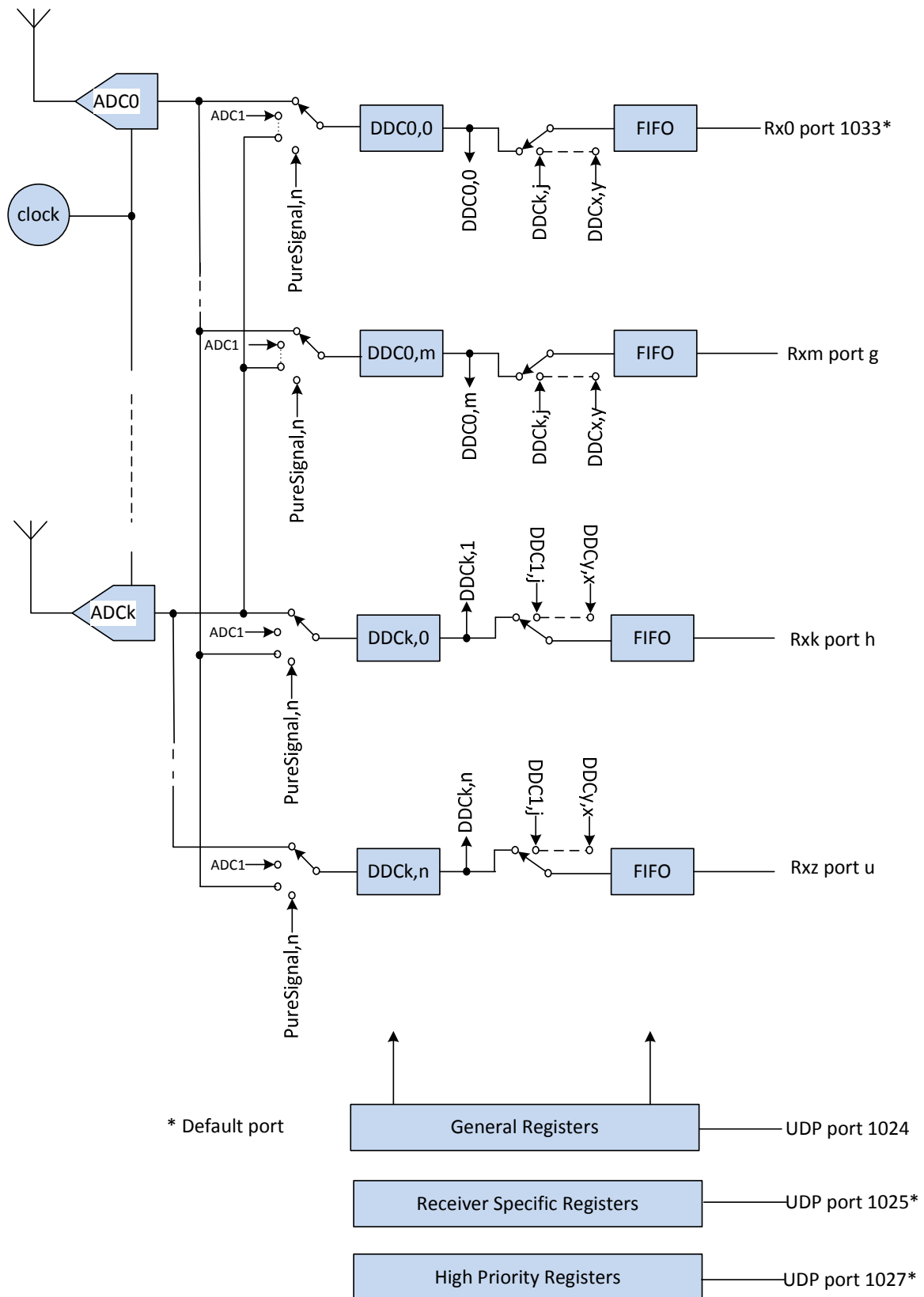
\* Default Port

FIGURE 2 UDP DATA TO HOST

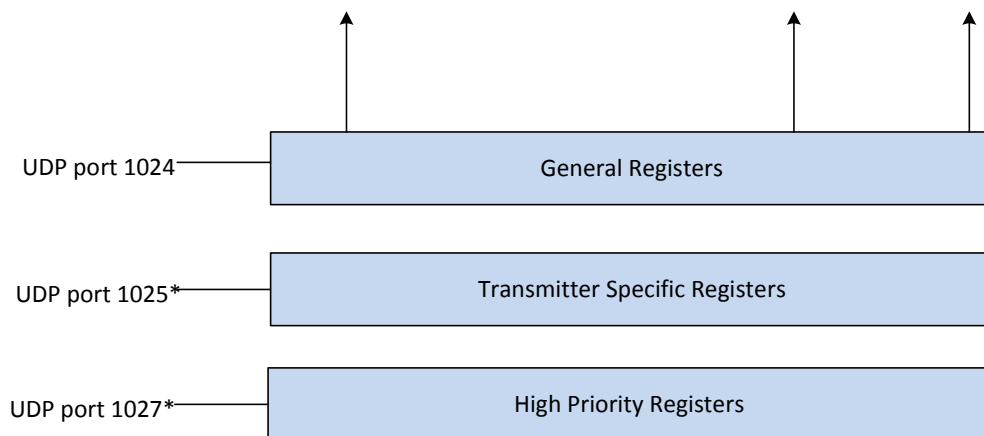
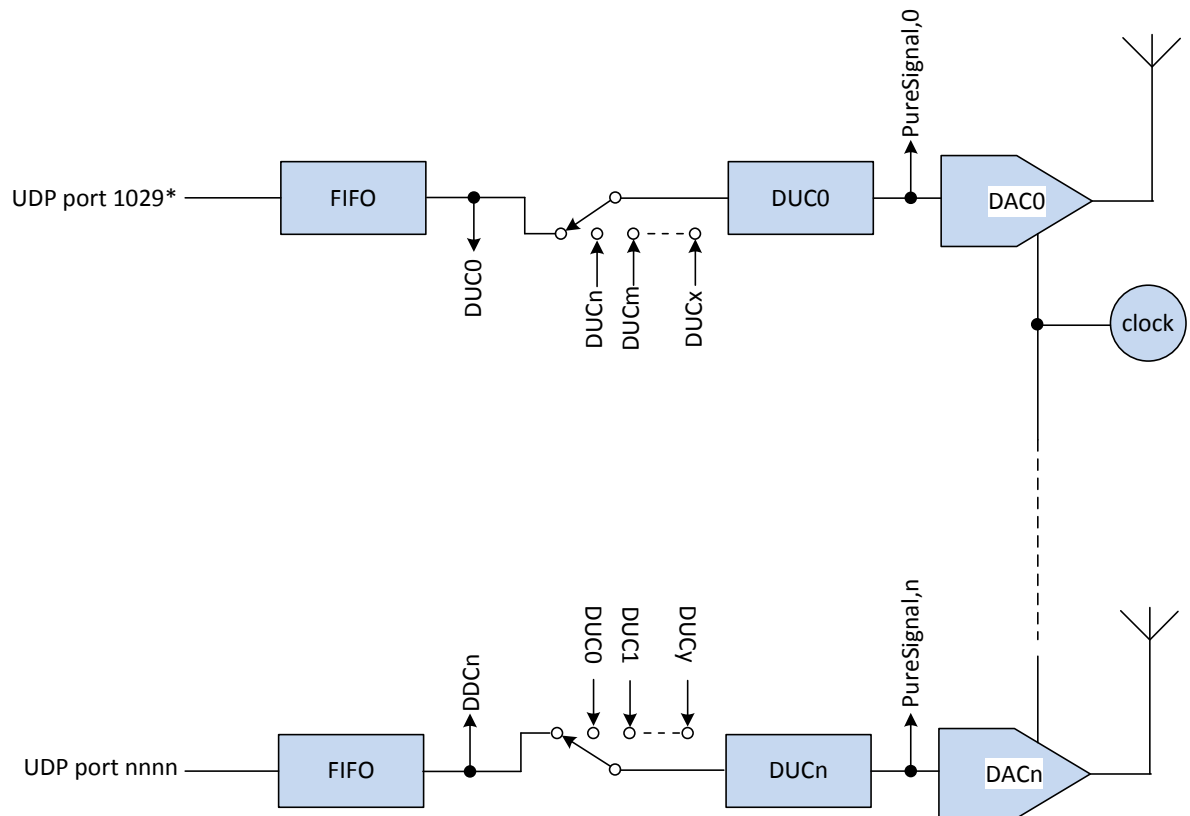


\* Default Port

**FIGURE 3 RECEIVER ARCHITECTURE**



**FIGURE 4 TRANSMITTER ARCHITECTURE**



\* Default Port

## APPENDIX A

To be written. Describes the hardware features of the SDR responding to a Discovery request.

Potential items to include:

- Master clock frequency e.g. 125MHz
- Number of ADCs
- Number of DACs
- Firmware serial numbers