Client:	KAT-7, SKA-SA and NRF
Project:	SPEAD
Type:	KAT-7 Data Interface Definition

KAT-7 Data Interface

Document number:	K0000-2006V1-05
Revision:	2
Classification:	Open Source, GPL
Author:	Jason Manley, Marc Welz and Simon Ratcliffe
Date:	2013/02/11

Document Approval

Function	Name	Designation	Affiliation	Date	Signature
Submitted by	R. van Rooyen	DBE DSP Specialist	SKA SA		
Approved by	M. Welz	DBE Software	SKA SA		
Approved by	J. Manley	DBE DSP Specialist	SKA SA		
Approved by	S. Ratcliffe	SPT Subsystem Manager	SKA SA		
Accepted by	F. Kapp	DBE Subsystem Manager	SKA SA		

Document History

Revision	Date of Issue	ECN Number	Comments
A	2010/12/28	N/A	Creation of SPEAD recommended practice document.
B	2011/09/31	N/A	Updated SPEAD items.
C	2012/06/06	N/A	Add new items defined for KAT-7 narrow band and single dish.
1	2012/10/16	N/A	First revision release.
2	2013/02/11	N/A	Addition of beamformer metadata omitted from summary table.

Document Software

Task	Package	Version
Stylesheet	katdoc	1.1.2
Text processor	LaTeX	3.1415926-1.40.10
Editor	Vim	7.2.445

Company Details

Name	SKA South Africa Office
Physical/Postal Address	Third Floor The Park Park Road Pinelands 7405
Tel.	+27 21 506 7300
Fax	+27 21 506 7375
Website	http://www.ska.ac.za/

Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog to Digital Converter
CAM	Control and Monitoring
DBE	Digital Backend
FF	Fringe Finder
ICD	Interface Control Document
IP	Internet Protocol
KAT	Karoo Array Telescope
KATCP	KAT Communication Protocol
LRU	Line Replacement Unit

1 Applicable and Reference Documents

1.1 Applicable Documents

The following documents are applicable to the extent stated herein. In the event of conflict between the contents of the applicable documents and this document, the applicable documents shall take precedence.

- *No applicable documents*

1.2 Related Documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, this document shall take precedence.

[1] Jason Manley, Marc Welz, Aaron Parsons, Simon Ratcliffe. Spead: Streaming protocol for exchanging astronomical data. SSA4700-0000-001, Rev 1.

[2] Simon Ratcliffe. https://github.com/ska-sa/pyspead. 2010.

2 Introduction

The KAT-7 digital backend uses an FX-correlator architecture that is expected to output correlated visibility data, single dish raw voltage data, and tied-array data products.

The correlator uses the SPEAD protocol [1] for the transfer of these radio astronomy data products.

This document aims to describe the specifics of the SPEAD protocol as it is applied to the KAT-7 correlator.

3 Data exchange procedure

Before the start of the data transmission, initial *ItemDescriptor* packets are sent as a number of metadata packets. In addition to instrument configuration parameters, these also contain the setup information of any *Item*s required to transport the FPGA payloads. Once the descriptors have been transmitted, the receiver will be in a position to decode the option fields. These *ItemDescriptor* packets are typically issued by a control computer, and not by FGPAs, even in cases where the FPGAs stream the actual SPEAD data. This leverages SPEAD's ability for multiple transmitters to contribute data to a single receiver in order to simplify the FPGA designs.

A detailed description of the format of the *ItemDescriptor* packets can be found in Reference [2]. In general, an *ItemDescriptor* is a special SPEAD *Item* that may contain such metadata *Item*s such as NAME, DESCRIPTION, TYPE, SHAPE and ID. This metadata allows the receiver to automatically unpack and interpret the datastream.

For example, any instrument implementing a per-channel gain control will require a number of parameters to describe the digital gain setting on each frequency channel for every ADC. The configured parameters for this gain control will differ for individual observations, and is dependent on the sky temperature. The current settings for a given observation are propagated to the data receiver to be stored with the actual output datastream of the instrument.

The following tables list the current SPEAD metadata *Item*s that are sent by the KAT-7 Digital BackEnd (DBE).

In the KAT-7 implementation, retransmission of all the metadata may be requested at any time, through an out-of-band KATCP control interface. This is to allow any new receivers to lock on to an existing datastream.

4 KAT-7 FX Correlator

For the KAT-7 correlator, all the SPEAD data descriptors and metadata relevant to a selected observation mode are issued to the assigned receiver IP and port. The port and IP is assigned by software on the central DBE control computer. The following steps are performed, in order, upon initialisation or mode change:

1. If the system was already outputting data products, send a SPEAD end of stream notification.
2. Send a single SPEAD heap containing the *ItemDescriptor*s and system time synchronisation items.
3. Send the SPEAD heaps containing all the *ItemDescriptor*s and initial values for each ADCs' analogue gain as well as the system's digital gain stages as each input is initialised.
4. Send the SPEAD heaps containing the *ItemDescriptor*s and initial values of the labels, i.e. the names of all physical inputs, as well as baseline ordering as each input is initialised.
5. Send a single SPEAD heap containing the static *ItemDescriptor*s and initial values for most static variables and system parameters.
6. Send a single SPEAD heap containing the *ItemDescriptor*s only for the FPGA-based 10GbE output products, to enable receivers to decode the emitted data.

Each stage is detailed below, along with its initial values. Please note that, while indicative of typical received values, the initial values are subject to change. Unless otherwise noted, the initial values are for the c16n400M1k (1024 channel wideband) mode.

4.1 Timing metadata

The correlator time metadata notifies the receiver that the system has been resynchronised, or that a parameter has changed that could potentially affect the system timing. An example is a change in the accumulation length. See Table 1 for the initial values.

Timing metadata is issued:

- upon entering a given correlator mode,
- when resynchronising,
- changing the accumulation period,
- or when explicitly requesting a metadata reissue.

Table 1: Correlator timing metadata

Name	**Initial Value**
n_accs	390625
int_time	1.000
scale_factor_timestamp	12207.03125
sync_time	≥ 0

4.2 Static metadata

Data descriptors and initial values for all correlator-specific static metadata are emitted in a single heap. See Table 2 for a full listing.

The static metadata is issued whenever:

- the system enters a correlator or beamformer mode,
- the user explicitly requests a metadata reissue,
- if the center frequency selection change in a narrowband mode, when the DBE is only processing a subset of the digitised bandwidth.

Table 2: Static metadata issued by correlator modes. This is metadata that is unlikely to change during operation and is normally only issued at mode change.

Name	**Initial Value**
adc_clk	800000000
n_bls	36
n_chans	1024
n_ants	8
n_xengs	16
center_freq	200000000
bandwidth	400000000
xeng_acc_len	128
requant_bits	4
feng_pkt_len	128
fft_shift	1023
rx_udp_port	7148
feng_udp_port	8888
rx_udp_ip_str	'192.168.10.10'
feng_start_ip	167772160
xeng_rate	200000000
x_per_fpga	2
n_ants_per_xaui	1
ddc_mix_freq	0
adc_bits	8
xeng_out_bits_per_sample	32

In the event that the KAT-7 correlator is in a wideband correlator mode (is processing the full digitised bandwidth), *Item* `fft_shift` will also be issued. This item is a bitfield describing the internal divisors used between FFT butterfly stages.

The KAT-7 narrowband correlator modes do not use digital downconverters, but implement cascaded polyphase filterbanks. In this case, there are two FFT_shift schedules (*Items* `fft_shift_coarse` and `fft_shift_fine`) and two additional metadata *Items*, `coarse_chans` and `current_coarse_chan`, which describe the total number of channels and the selected channel in the first PFB. These items are only issued in narrowband modes.

4.3 Gain metadata

This SPEAD heap contains metadata for the RF gain and EQ settings. See Table 3 initial values expected in wide band mode.

Gain metadata is issued whenever:

- the system enters a correlator or beamformer mode,
- the user explicitly requests a metadata reissue,
- or any input's gain setting is altered.

Table 3: Correlator gain metadata

Name	Initial Value
rf_gain_*ant_str0*†	4
rf_gain_*ant_str1*†	4
rf_gain_*ant_str2*†	4
⋮	⋮
rf_gain_*ant_str15*†	4
eq_coef_*ant_str0*†	[300,300,300... 300]
eq_coef_*ant_str1*†	[300,300,300... 300]
eq_coef_*ant_str2*†	[300,300,300... 300]
⋮	⋮
eq_coef_*ant_str15*†	[300,300,300... 300]

†The italicised suffix refers to the user-assigned input name.

The equaliser coefficients are complex scale factors, applied by the F-engines prior to 4 bit re-quantisation. Each input applies a separate complex scale factor for each frequency in the system. The length of each array is thus given by the *Item* `n_chans`.

4.4 Input and baseline labelling

The SPEAD labelling metadata items describe the labelling, location and connections of the system's analogue inputs and their mappings to output baselines. See Table 4 for expected labelling metadata.

The correlator input labelling metadata product is issued whenever:

- the system enters a correlator or beamformer mode,
- the user explicitly requests a metadata reissue,
- or any input is renamed by the user.

For KAT-7, the input labelling item is a 2-D array with dimensions $N \times 4$ for N analogue inputs. Each analogue input N, indexed by a system-wide unique input number, is described by an array of the form:

Table 4: Correlator input labelling metadata

<table>
<tr><th>Name</th><th>Initial Value</th></tr>
<tr><td>bls_ordering</td><td><table>
<tr><td>‘0x’</td><td>‘0x’</td></tr>
<tr><td>‘0y’</td><td>‘0y’</td></tr>
<tr><td>‘0x’</td><td>‘0y’</td></tr>
<tr><td>⋮</td><td>⋮</td></tr>
<tr><td>‘2y’</td><td>‘7x’</td></tr>
</table></td></tr>
<tr><td>input_labelling</td><td><table>
<tr><td>‘0x’</td><td>0</td><td>‘roach030267’</td><td>0</td></tr>
<tr><td>‘0y’</td><td>1</td><td>‘roach030267’</td><td>1</td></tr>
<tr><td>‘1x’</td><td>2</td><td>‘roach030277’</td><td>0</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>‘7y’</td><td>15</td><td>‘roach030234’</td><td>1</td></tr>
</table></td></tr>
</table>

[string] user-assigned_antenna_name,
[integer] instrument-wide_unique_input_number,
[string] LRU,
[integer] input_number_on_this_LRU

The user-assigned antenna name is as specified through the DBE CAM interface.

The instrument-wide unique input number is an integer describing the logical interface of this instrument. The afore-mentioned antenna is connected to this interface. In the case of KAT-7, this number ranges from 0 to 15. The first two system inputs are connected to either a dummy load or a noise sources, which are used for self-tests while the remaining 14 inputs are used for antenna inputs.

The LRU is a hardware identifier for a line-replaceable-unit. KAT-7 F-engines are populated with a single dual-input ADC each and hence each have two analogue inputs per LRU. The input numbers thus ranges from 0 to 1 on each LRU.

An example example baseline entry might be: *(‘antC23y’ , 12 , ‘roach030267’ , 1)*.

The baseline ordering item lists the output order of the X-engines. The form is a matrix with dimensions of $N \times 2$, for N baselines. Each line contains two strings of user-defined antenna names (‘input1’,‘input2’). Two baseline example: [(‘antC23x’,‘antC23y’), (‘antB12y’,‘antA29y’)].

4.5 Hardware heaps

SPEAD data descriptors are issued from the control computer for the correlator’s FPGA-based 10GbE output in order to enable receivers to decode the data. This is the highspeed stream of heaps emitted from all X-engines. See Table 5.

The KAT-7 correlator’s datastream output (ID 0x1800) is collected into a single *Heap* by the receiver. This provides a convenient model for a single receiver, but this solution does not scale well for large systems with

Table 5: Correlator hardware heaps

Name	Initial Value
timestamp	≥ 0
xeng_raw	multi-dimensional binary array

multiple, distributed receivers. In these cases we will need multiple `xeng_raw` outputs, each going to a separate receiver. The simplest way to achieve this is to split the output of each X-engine into a separate SPEAD stream, as is done for the beamformer. In this case, each receiver would receive all baselines for a subset of the band. This was implemented for early prototype systems with separate heaps from each X-engine in the system, with each heap containing the SPEAD item with ID 0x1800+*inputN* (instead of item ID 0x1800) representing a subset of the band. KAT-7 no longer supports this output mode as, in all cases, the correlator output datarates are considered low enough for a single receiver to cope with the entire structure of all dataproducts.

5 Frequency-domain beamformer

KAT-7's beamformer is added-on to the correlator discussed in §4. Much of the underlying infrastructure is shared between these two instruments. If the selected correlator mode also supports frequency-domain beamforming, additional beamformer metadata will be issued during initialisation, after the normal correlator metadata. This metadata is detailed here.

Multiple, simultaneous beams are supported by KAT-7 (initially one for each of the two polarisations). It is intended that each beamformer's output will be routed to separate SPEAD receiver(s) and so no attempt was made to keep variable names globally unique. All beamformer-applicable metadata is reissued to each receiver's network port using the same SPEAD Item IDs and variable names.

The beamformer metadata is issued in the following order upon initialisation, mode change or metadata reissue request:

1. If the system was already outputting data products, a SPEAD end of stream notification is issued.
2. A single SPEAD *Heap* containing the *ItemDescriptor*s and system time synchronisation items is issued.
3. SPEAD *Heap*s containing all the *ItemDescriptor*s and initial values for each ADCs' analogue gain as well as the beamformer's static weights and digital gain stages are issued as each input is initialised.
4. A single SPEAD *Heap* containing the static *ItemDescriptor*s and initial values for most static variables and system parameters is issued.
5. SPEAD *Heap*s containing the *ItemDescriptor*s and labels of all physical inputs are issued as each input is initialised.
6. A single SPEAD *Heap* is issued, containing the *ItemDescriptor*s only for the FPGA-based 10GbE output products, to enable receivers to decode the emitted data.

The following subsections describe the *Heap*s issued for KAT-7's frequency-domain beamformer.

Table 6: Beamformer input labelling metadata

<table>
<tr><th>Name</th><th>Initial Value</th></tr>
<tr><td>input_labelling</td><td><table>
<tr><td>‘0x’</td><td>0</td><td>‘roach030267’</td><td>0</td></tr>
<tr><td>‘1x’</td><td>2</td><td>‘roach030277’</td><td>1</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>‘7x’</td><td>14</td><td>‘roach030234’</td><td>7</td></tr>
</table></td></tr>
</table>

5.1 Input labelling

Beamformer input labelling metadata items describing the labelling, location and connections of the system's analogue inputs are issued whenever:

- the system enters a correlator mode that supports beamformer functions
- the user explicitly requests a metadata reissue
- any input is renamed by the user.

KAT-7's beamformer follows the same labelling scheme as the KAT-7 correlator (see §4.4), but with only the subset of items described in Table 6.

In the case of sub-arraying, where only a subset of the system's inputs are summed, only inputs applicable to this beam are emitted. This item thus informs the receiver which inputs are being summed to construct this beam.

5.2 Input scaling and weightings

This is designed to propagate the current beam weightings that are being applied to each input. In KAT-7's case, this is a per-input, per-frequency complex value, allowing for correction of phase and amplitude mismatches between analogue inputs and also to configure the relative contributions of each input to the summed beam.

The beam weightings are complex scale factors, applied by the B engines to the incoming 4 bit numbers before summing. Each input applies a separate complex scale factor for each frequency in the system (as with `eq_coef`, the length of each array is thus given by the *Item* `n_chans`).

5.3 Beamformer static metadata

The beamformer static metadata will take the form of a SPEAD heap containing static variables.

KAT-7's wideband sample clock rate is fixed for the wideband mode, however, the processed centre frequency and bandwidth values will change depending on the selected beamformer band (see §5.5).

5.4 Beamformer timing

The beamformer's timing follows the correlator's timing precisely and is detailed in Table 9.

Table 7: Beamformer scaling metadata

Name	Initial Value
rf_gain_*ant_str0*†	4
rf_gain_*ant_str1*†	4
rf_gain_*ant_str2*†	4
...	...
rf_gain_*ant_str15*†	4
eq_coef_*ant_str0*†	[300,300,300... 300]
eq_coef_*ant_str1*†	[300,300,300... 300]
eq_coef_*ant_str2*†	[300,300,300... 300]
...	...
eq_coef_*ant_str15*†	[300,300,300... 300]
beamweight_*ant_str0*†	[1.0+1.0j,1.0+1.0j, ... 1.0+1.0j]
beamweight_*ant_str1*†	[1.0+1.0j,1.0+1.0j, ... 1.0+1.0j]
beamweight_*ant_str2*†	[1.0+1.0j,1.0+1.0j, ... 1.0+1.0j]
...	...
beamweight_*ant_str15*†	[1.0+1.0j,1.0+1.0j, ... 1.0+1.0j]

†The italicised suffix refer to the user-assigned input name.

Table 8: Beamformer static metadata

Name	Nominal Value
adc_clk	800000000
n_chans	1024
n_ants	8
n_bengs	16
center_freq	200000000
bandwidth	400000000
xeng_acc_len	128
requant_bits	4
fft_shift	1023
feng_pkt_len	128
ddc_mix_freq	0
adc_bits	8
rx_udp_port	7148
rx_udp_ip_str	‘192.168.10.10’
beng_out_bits_per_sample	8

Table 9: Beamformer timing metadata

Name	Initial Value
scale_factor_timestamp	12207.03125
sync_time	≥ 0

5.5 Beamformer hardware output

The beamformer hardware outputs a number of frequency channels, given by the *Item* `n_chans`. These channels are processed by a number of engines (*n_bengs*) and the beamformer output can be chosen from each of these engines). KAT-7 has 16 B-engines and digitises a 400MHz band, allowing for the output centre frequency and bandwidth to be chosen in increments of $\frac{400MHz}{16} = 25$MHz. Data for unused portions of the band will not be emitted, so this feature can be used to save data bandwidth in the event that the full analogue band is not needed.

Beams can be arbitrarily renamed and the output SPEAD variable name will reflect this string precisely. No provision is made to prevent overlapping namespaces nor are there any safety or sanity checks done on the chosen name.

The beam name, centre frequency, bandwidth, output IP address and port etc. are configured through a separate CAM interface which falls outside of the scope of this document.

Table 10: Additional beamformer hardware metadata

Name	Nominal Value
MyBeam	The raw beamformer data.
timestamp	The FPGA timestamp of the first data sample in the data block.

Beamformer data is emitted as a time series of spectra. It is presented in frames of `n_chans` by `feng_pkt_len` complex integer numbers.

6 Complete KAT-7 DBE metadata listing

Implementation Note: To remain completely implementation agnostic, Unix time and SI units of measurement are used wherever possible. It is strongly encouraged that this model be used moving forward.

In some cases, uint data types were specified which might be specific to KAT-7. For example, when propagating the synchronisation time, we are assured in KAT-7 that we will always synchronise to exactly a second boundary since the FPGAs realign to a 1PPS pulse by design. So there is no point being able to specify fractions of a second for sync_time. For this reason, the sync_time variable is an integer. However, another system might allow syncing to within less than one second or support fractional start times. In these cases, you would probably want to use floating point numbers with a defined precision.

In this document, the size `spead.ADDRSIZE` means that the width of the field is inherited from the SPEAD flavour. In KAT-7's case, this is 40 bits from SPEAD64-40. However, this could be any SPEAD-compliant value. The reason it's not hard-coded is so that if we change the SPEAD flavour, the code scales appropriately. This allows for use of the more efficient IMMEDIATE addressing mode. See the SPEAD reference document for more information [1].

Table 11: A list of all the defined metadata in use on KAT-7, ordered by *Item* ID.

ID	Name and Type	Description
`0x1007`	`adc_clk` (`'u'`†`,64)`	Clock rate of ADC (samples per second).

Continued on next page...

Table 11 ... continued from previous page

ID	Name and Type	Description
0x1008	n_bls (‘u’,spead.ADDRSIZE‡)	The total number of baselines in the data product.
0x1009	n_chans (‘u’,spead.ADDRSIZE)	The total number of frequency channels present in any integration.
0x100A	n_ants (‘u’,spead.ADDRSIZE)	The total number of dual-pol antennas in the system.
0x100B	n_xengs (‘u’,spead.ADDRSIZE)	The total number of X engines in a correlator system.
0x100C	bls_ordering 2-D array of strings	The X-engine baseline output ordering. The form is a list of arrays of strings of user-defined antenna names (’input1’,’input2’). For example, [(’antC23x’,’antC23y’), (’antB12y’,’antA29y’)]
0x100D	crosspol_ordering string pair	The output ordering of the cross-pol terms. Packed as a pair of characters, pol1,pol2. DEPRECATED: New mechanism uses 0x100E and 0x100C to describe each physical input individually, irrespective of polarisation.
0x100E	input_labelling string	The physical location of each antenna’s connection. It is an array of structures, each with the following form in the case of KAT-7: (user-assigned_antenna_name, systemwide_unique_input_number, LRU, input_number_on_this_LRU). An example entry might be: (’antC23y’,12,’roach030267’,3)
0x100F	n_bengs (‘u’,spead.ADDRSIZE)	The total number of B engines in a beamformer system.
0x1011	center_freq (‘f’,64)	The center frequency of the DBE in Hz, 64-bit IEEE floating-point number.
0x1013	bandwidth (‘f’,64)	The analogue bandwidth of the digitally processed signal in Hz.
0x1015	n_accs (‘u’,spead.ADDRSIZE)	The number of spectra that are accumulated per integration.

Continued on next page...

Table 11 ... continued from previous page

ID	Name and Type	Description
0x1016	int_time ('f',64)	Approximate (it's a float!) time per accumulation in seconds. This is intended for reference only. Each accumulation has an associated timestamp which should be used to determine the time of the integration rather than incrementing the start time by this value for sequential integrations (which would allow errors to grow).
0x1017	coarse_chans ('u',spead.ADDRSIZE)	The number of coarse channels (ie of the first PFB) in a cascaded PFB narrowband correlator design.
0x1018	current_coarse_chan ('u',spead.ADDRSIZE)	The currently selected coarse channel (ie of the first PFB) in a cascaded PFB narrowband correlator design.
0x101C	fft_shift_fine ('u',spead.ADDRSIZE)	The FFT bitshift pattern for the fine channelisation FFT. F-engine correlator internals.
0x101D	fft_shift_coarse ('u',spead.ADDRSIZE)	The FFT bitshift pattern for the first (coarse) channelisation FFT. F-engine correlator internals.
0x101E	fft_shift ('u',spead.ADDRSIZE)	The FFT bitshift pattern. F-engine correlator internals.
0x101F	xeng_acc_len ('u',spead.ADDRSIZE)	Number of spectra accumulated inside X engine. Determines minimum integration time and user-configurable integration time stepsize. X-engine correlator internals.
0x1020	requant_bits ('u',spead.ADDRSIZE)	Number of bits per sample after requantisation. For FX correlators, this represents the number of bits after requantisation in the F engines (post FFT and any phasing stages) and is the actual number of bits used in X-engine processing. For time-domain systems, this is requantisation in the time domain before any subsequent processing.
0x1021	feng_pkt_len ('u',spead.ADDRSIZE)	Payload size of 10GbE packet exchange between F and X engines in 64 bit words. Usually equal to the number of spectra accumulated inside X engine. F-engine correlator internals.
0x1022	rx_udp_port ('u',spead.ADDRSIZE)	Destination UDP port for data output.

Continued on next page...

Table 11 ... continued from previous page

ID	Name and Type	Description
0x1023	feng_udp_port (‘u’,spead.ADDRSIZE)	Destination UDP port for F engine data exchange.
0x1024	rx_udp_ip_str string	Destination IP address for output UDP packets.
0x1025	feng_start_ip (‘u’,spead.ADDRSIZE)	F engine starting IP address.
0x1026	xeng_rate (‘u’,spead.ADDRSIZE)	Target clock rate of processing engines (xeng).
0x1027	sync_time (‘u’,spead.ADDRSIZE)	Time at which the system was last synchronised (armed and triggered by a 1PPS) in seconds since the Unix Epoch.
0x1040	n_stokes (‘u’,spead.ADDRSIZE)	The number of Stokes parameters processed by this correlator. DEPRECATED. This field has been replaced by individual labelling of each physical input and correlator baseline output listing, irrespective if two polarisations of the same antenna or different antennas. Thus, it makes no difference how many cross-pol terms are calculated as each is listed individually. See 0x100C and 0x100E instead.
0x1041	x_per_fpga (‘u’,spead.ADDRSIZE)	Number of X engines per FPGA.
0x1042	n_ants_per_xaui (‘u’,spead.ADDRSIZE)	Number of antennas’ data per XAUI link.
0x1043	ddc_mix_freq (‘f’,64)	Digital downconverter mixing freqency as a fraction of the ADC sampling frequency. eg: 0.25. Set to zero if no DDC is present.
0x1044	ddc_bandwidth (‘f’,64)	Digital downconverter output bandwidth or decimation factor. DEPRECATED. Use ID 0x013 instead.
0x1045	adc_bits (‘u’,spead.ADDRSIZE)	ADC resolution (number of bits).
0x1046	scale_factor_timestamp (‘f’,64)	Timestamp scaling factor. Divide the SPEAD data packet timestamp by this number to get back to seconds since last sync.
0x1047	b_per_fpga (‘u’,spead.ADDRSIZE)	Number of B engines per FPGA.
0x1048	xeng_out_bits_per_sample (‘u’,spead.ADDRSIZE)	The number of bits per value of the xeng accumulator output. Note this is for a single component value, not the combined complex size.
0x1049	f_per_fpga (‘u’,spead.ADDRSIZE)	Number of F engines per FPGA.

Continued on next page...

Table 11 ... continued from previous page

ID	Name and Type	Description
0x1050	beng_out_bits_per_sample ('u',spead.ADDRSIZE)	The number of bits per value of the beng accumulator output. Note this is for a single component value, not the combined complex size.
0x1200+*inputN*	rf_gain_*MyAntStr* ('f',64)	The analogue RF gain applied at the ADC for *inputN* in dB.
0x1400+*inputN*	eq_coef_*MyAntStr* ('u',32)	The unitless per-channel digital scaling factors implemented prior to requantisation, post-FFT, for *inputN*. Complex number real,imag 32 bit integers.
0x1600	timestamp ('u',spead.ADDRSIZE)	Timestamp of start of this integration. *uint* counting multiples of ADC samples since last sync (`sync_time`, `id=0x1027`). Divide this number by `timestamp_scale` (`id=0x1046`) to get back to seconds since last sync when this integration was actually started. Note that the receiver will need to figure out the centre timestamp of the accumulation (eg, by adding half of `int_time`, `id 0x1016`).
0x1800	xeng_raw (int32,(n_chans,n_bls,2))	Raw data stream from all the X-engines in the system. For KAT-7, this item represents a full spectrum (all frequency channels) assembled from lowest frequency to highest frequency. Each frequency channel contains the data for all baselines (`n_bls` given by SPEAD `Id=0x100B`). Each value is a complex number – two (real and imaginary) unsigned integers.
0x3100	n_inputs ('u',spead.ADDRSIZE)	The total number of analogue inputs in the stream.
0x3103	pkt_len ('u',spead.ADDRSIZE)	Payload size of 10GbE packet.
0x3300+*inputN*	raw_data_*MyAntStr* (int8,4096) or (('i',4),('i',4))	Raw time-domain data stream from the ADC(s). The values in the stream may either be represented by a signed 8b integer value, or else by two 4b integer samples, one from each polarisation, that are interleaved.
0xb000	*MyBeamName* (int8, (n_chans, xeng_acc_len,2))	Beamformer output for frequency-domain beam, band subset number *subsetN*.
0x2000+*inputN*	beamweight_*MyAntStr* ('u',32)	The unitless per-channel digital scaling factors implemented prior to combining signals for beamformer, for *inputN*, per beam. Complex number real,imag 32 bit integers.