SKA SOUTH AFRICA
SQUARE KILOMETRE ARRAY

SPEAD Recommended Practice

Document number:	SPEADmetadata
Revision:	C
Classification:	Open Source, GPL
Author:	J. Manley, M. Welz, A. Parsons, S. Ratcliffe, R. van Rooyen
Date:	2012/06/07

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.

Document Software

	Package	Version	Filename
Stylesheet	casperdoc	1.1.1	casperdoc.sty
Word processor	LaTeX	3.1415926-1.40.9 (Web2C 7.5.7)	spead.tex
Diagrams	epstopdf	2.9.5gw	images/ska_logo

Contents

1 Applicable and Reference Documents **5**

1.1 Applicable Documents 5

1.2 Related Documents 5

2 Metadata Packets **5**

2.1 Additional Field Options 9

List of Figures

List of Tables

List of Abbreviations

KAT	Karoo Array Telescope
KATCP	KAT Communication Protocol
ICD	Interface Control Document
IP	Internet Protocol
FF	Fringe Finder

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] Simon Ratcliffe. http://github.com/sratcliffe/pyspead. 2010.

2 Metadata Packets

Before the start of data transmission, initial *ItemDescriptor* packets are sent in a number of metadata packets. These contain the setup information of any *Items* required to transport payloads. Once the descriptors have been transmitted the receiver will be in a position to decode the option fields.

A detailed description of the *ItemDescriptor* packets can be found in Reference [1]. In general an *ItemDescriptor* is a *SpeadStream* that may contain such metadata *Items* as NAME, DESCRIPTION, TYPE, SHAPE and ID.

```
instance.add_item(
name= " <ItemDescriptorName> ",              # string identifying variable
id= 0x<ItemDescriptorID>,                    # item's numerical code
description= " <ItemDescriptorDescription> ", # explanatory string
shape= [ <ItemDescriptorShape> ],            # variable dimensions
fmt= spead.mkfmt( <ItemDescriptorType> )     # data type ane number of bits
                 )
```

The setup for any instrument implementing per-channel gain control will require a number of metadata packets describing the digital gain setting on each frequency channel for each ADC, describe all the quantisation scalars for all ADCs, all polarisations, all channels, etc. The following tables list the current SPEAD metadata *Items* that are sent with the correlator outputs. These represent a list of recommended *ItemIdentifiers* for common astronomy applications and is valid for the narrowband and wideband modes.

Implementation Note: To remain completely 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 that is implementation specific. For example, with sync time, we are assured in KAT7 that we will sync to exactly one second boundary since by design it aligns to a 1PPS pulse. So there's no point specifying a fraction of a second. However, another system might allow syncing to within less than one second or support fractional start times. In these cases, you'd probably want to use floating point numbers.

The size spead.ADDRSIZE (‡) means that the width of the field is inherited from the SPEAD flavour. In our case 40 bits, but 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.

Id	Name Type	Description
GENERIC CORRELATOR METADATA		
0x1007	adc_clk ('u'†,64)	Clock rate of ADC (samples per second).
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.
0x100C	bls_ordering spead.STR_FMT	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')]
0x100E	input_labelling spead.STR_FMT	The physical location of each antenna's connection. It is an array of arrays, 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)
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.
0x1016	int_time ('f',64)	Approximate (it's a float!) integration 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).

0x1020	requant_bits ('u',spead.ADDRSIZE)	Number of bits after requantisation in the F engines (post FFT and any phasing stages). This is the actual number of bits used in X-engine processing.
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.
0x1045	adc_bits ('u',spead.ADDRSIZE)	ADC resolution (number of bits).
RECEIVER METADATA		
0x1022	rx_udp_port ('u',spead.ADDRSIZE)	Destination UDP port for X engine output.
0x1024	rx_udp_ip_str spead.STR_FMT	Destination IP address for X engine output UDP packets.
CASPER PACKETISED CORRELATOR SPECIFIC*		
0x100B	n_xengs ('u',spead.ADDRSIZE)	The total number of X engines in the system.
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.
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.
0x1023	feng_udp_port ('u',spead.ADDRSIZE)	Destination UDP port for F engine data exchange.
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).
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.
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 value, not the combined complex size.
KAT7 CORRELATOR TIMESTAMPS**		
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.

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.
0x1600	timestamp ('u',spead.ADDRSIZE)	Timestamp of start of this integration. *u*int 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).
KAT7 CORRELATOR WIDE BAND MODE***		
0x101E	fft_shift ('u',spead.ADDRSIZE)	The FFT bitshift pattern. F-engine correlator internals.
KAT7 CORRELATOR NARROW BAND MODE***		
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 coarse channelisation FFT. F-engine correlator internals.
KAT7 CORRELATOR RAW VOLTAGE CAPTURE***		
0x1048	t_bits_per_sample ('u',spead.ADDRSIZE)	Number of bits per sample after requantisation in the time domain.
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+input_n	raw_data_inputN (numpy.dtype(numpy.int8),(4096)) or (('i',4),('i',4))	Raw data stream from the ADC(s). The values in the stream may either be represent by a signed integer value, or two four bit samples that are interleaved (ie sample from one polarisation, sample from other polarisation).

* Some of the fields are appropriate for general correlators and some are specific to our implementation.
* This is how we output timestamps but this is application specific to each system.
** The KAT-7 correlator has a number of modes that have application specific parameters.

2.1 Additional Field Options

Id	Name Type	Description
0x1200+input_n	rf_gain_inputN ('f',64)	The analogue RF gain applied at the ADC for input N in dB.
0x1400+in_n	eq_coef_inputN ('u',32)	The unitless per-channel digital scaling factors implemented prior to re-quantisation, post-FFT, for input N. Complex number real,imag 32 bit integers.

The KAT7 correlator output is collapsed into a single HEAP by the receiver. Obviously this won't work for large systems with distributed RX. In these cases we will need multiple xeng_raw outputs, each going to a separate receiver.

Id	Name Type	Description
0x1800	xeng_raw (numpy.dtype(numpy.int32),(n_chans,n_bls,2))	Raw data for each xengine in the system. 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.