TDC Tec. Rep.
T.KONDO
2010.8.20

K5/VSSP and K5/VSSP32 Data Format

update history

2008.07.27 Add a table of sampled data format in a 32-bit word

2007.10.20 Positions of items (format # and filter frequency) in a aux field were wrong (switched). It was corrected.

2010.07.06 Extended format (Format #21) was newly defined.

2010.08.20 Extended format (Format #21) was a bit revised.

1 Data Structure

K5/VSSP or K5/VSSP32 sampler output is made up of multiple frames, which consist of a header and a data block (Fig.1). A header block of VSSP consists of 8-byte data which contain two sync blocks, time information, and sampling parameters. A header block of VSSP32 usually consists of 32-byte data, of which first 8-byte data are as same as that of VSSP excepting the second sync pattern. A rest of header block contains year, date, and user definable auxiliary field. A header block doesn't affect a data block, i.e., no data are lost by the insertion of the header blcok.

The size of a data block can be expressed by [sampling frequency (Hz)]×[# of AD bits]×[# of channels], however actual size is limitted by the maximum data rate between a sampler and host PC. In case of VSSP, which adopts PCI-bus interface, 64 Mbits are maximum data block size. As for VSSP32, which adopts USB-2.0 interface, the maximum data block size is 256 Mbits.

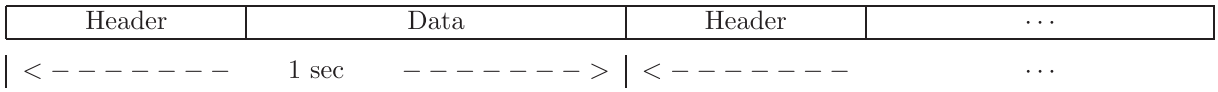

Figure 1: Data structure of K5/VSSP or K5/VSSP32 sampler output.

2 Header block

2.1 K5/VSSP header format

Table 1 shows K5/VSSP header format. It consists of 8 byte data.

Table 1. K5/VSSP Header Format (8 bytes).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern (all 0xFF)															
0x01																
0x02	time (seconds from 0h UTC) (17 bits)															(LSB)
0x03	2nd sync pattern (0x8B)								AD bits		sampling frequency				ch	(M)

where

sync pattern	–	0xFFFFFFFF
time	–	seconds from 0h UTC
ch	–	number of channels used 0: 1ch 1: 4ch

sampling frequency	–	index for sampling frequency (Note: sampling frequencies faster than and equal to 32 MHz are for supporting data converted from other format data) 0: 40kHz / 1: 100kHz / 2: 200kHz / 3: 500kHz 4: 1MHz / 5: 2MHz /6: 4MHz / 7: 8MHz / 8: 16MHz 9: 32MHz / 10: 64MHz / 11: 128MHz / 12: 256MHz / 13: 512MHz 14: 1024MHz / 15: 2048MHz
AD bits	–	number of AD resolution bits 0: 1 bit / 1: 2 bits / 2: 4 bits / 3: 8 bits
2nd sync pattern	–	0x8B (0x8C for VSSP32)

2.2 K5/VSSP32 header format (general specifications)

Table 2 shows the specifications of K5/VSSP32 data header. All current K5 utilities adopt 20 bytes for the size of auxiliary field, so that total size of header is 32 bytes.

Table 2. General specifications of K5/VSSP32 data header.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern (all 0xFF)															
0x01																
0x02	seconds from 0h UTC (17 bits)															(LSB)
0x03	2nd sync pattern (0x8C) (0x8B for VSSP)								AD bits		sampling frequency			ch		(M)
0x04	eflg	year (2 digits) (6bits : 0-63)							total day (9bits)							
0x05	major version #				minor version #				AUX FIELD size (in bytes : default is 20)							
0x06									AUX FIELD Format #							
0x07	AUX FIELD (user definable)															
0x08																
0x09																
0x0A																
0x0B																
0x0C																
0x0D																
0x0E																
0x0F																

where

sync pattern	–	0xFFFFFFFF
time	–	seconds from 0h UTC
ch	–	number of channels used 0: 1ch 1: 4ch
sampling frequency	–	index for sampling frequency (Note: sampling frequencies faster than and equal to 128 MHz are for supporting data converted from other format data) 0: 40kHz / 1: 100kHz / 2: 200kHz / 3: 500kHz 4: 1MHz / 5: 2MHz /6: 4MHz / 7: 8MHz / 8: 16MHz 9: 32MHz / 10: 64MHz / 11: 128MHz / 12: 256MHz / 13: 512MHz 14: 1024MHz / 15: 2048MHz
AD bits	–	number of AD resolution bits 0: 1 bit / 1: 2 bits / 2: 4 bits / 3: 8 bits
2nd sync pattern	–	0x8C (0x8B for VSSP)
total day	–	day of year
year	–	lower 2 digits of year
eflg	–	error flag (set when error occurred in a previous frame)
AUX FIELD size	–	size of auxiliary field (bytes)
major version #	–	major version # of VSSP32 control ROM
minor version #	–	minor version # of VSSP32 control ROM
AUX FIELD Format #	–	format # of auxiliary field

User can define the portion of “AUX FIELD” freely except for the first byte of auxiliary field, which is defined as the format number. When user defines a new format, confliction with the format number, that is predefined or reserved, should be avoid.

2.3 Predefined and reserved format numbers

Table 3 summarizes predefined and reserved format numbers for auxiliary field.

Table 3. Predefined and reserved format numbers

Format #	Note
0	for test
1	for observation (“autoobs” output data format)
2	for observation (“sampling” output data format)
30∼39	reserved (for ISAS group)
85	for test
170	for test
21	extended format (to support arbitrary sampling frequency and channels up to 16)

Predefined format numbers are 0, 1, 2, 85, 170, and 21. Numbers 30∼39 are reserved for ISAS group. Format #21 is defined to support arbitrary sampling frequency and channels up to 16. This format allows to conbine multiple sampler output. It is also possible to convert data obtained by a sampler other than VSSP-type sampler to VSSP32 format data.

2.3.1 Format #0

Table 4. VSSP32 Format #0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern (all 0xFF)															
0x01																
0x02	seconds from 0h UTC (17 bits)															(LSB)
0x03	2nd sync pattern (0x8C) (0x8B for VSSP)								AD bits		sampling frequency				ch	(M)
0x04	eflg	year (2 digits) (6bits : 0-63)							total day (9bits)							
0x05	major version #				minor version #				AUX FIELD size (in bytes : default is 20)							
0x06									AUX FIELD Format # (0)							
0x07	All 0															
0x08																
0x09																
0x0A																
0x0B																
0x0C																
0x0D																
0x0E																
0x0F																

2.3.2 Format #1

This is “autoobs” output data format.

Table 5. VSSP32 Format #1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern （all 0xFF)															
0x01																
0x02	seconds from 0h UTC (17 bits)															(LSB)
0x03	2nd sync pattern (0x8C) (0x8B for VSSP)								AD bits		sampling frequency				ch	(M)
0x04	eflg	year (2 digits) (6bits： 0-63)							total day (9bits)							
0x05	major version #				minor version #				AUX FIELD size (in bytes : default is 20)							
0x06	LPF frequency (MHz: 0 means through)								AUX FIELD format # (1)							
0x07	Station ID (max 2 charcters)															
0x08	Station name (max 8 characters)															
0x09																
0x0A																
0x0B																
0x0C	PC host name (max 8 characters)															
0x0D																
0x0E																
0x0F																

2.3.3 Format #2

This is “sampling” output data format.

Table 6. VSSP32 Format #2

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern （all 0xFF)															
0x01																
0x02	seconds from 0h UTC (17 bits)															(LSB)
0x03	2nd sync pattern (0x8C) (0x8B for VSSP)								AD bits		sampling frequency				ch	(M)
0x04	eflg	year (2 digits) (6bits： 0-63)							total day (9bits)							
0x05	major version #				minor version #				AUX FIELD size (in bytes : default is 20)							
0x06	LPF frequency (MHz: 0 means through)								AUX FIELD format # (2)							
0x07	filler data (0x5555)															
0x08	filler data (0x5555)															
0x09	filler data (0x5555)															
0x0A	filler data (0x5555)															
0x0B	filler data (0x5555)															
0x0C	PC host name (max 8 characters)															
0x0D																
0x0E																
0x0F																

2.3.4 Format #85

Table 7. VSSP32 Format #85

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern (all 0xFF)															
0x01																
0x02	seconds from 0h UTC (17 bits)															(LSB)
0x03	2nd sync pattern (0x8C) (0x8B for VSSP)								AD bits		sampling frequency				ch	(M)
0x04	eflg	year (2 digits) (6bits : 0-63)							total day (9bits)							
0x05	major version #				minor version #				AUX FIELD size (in bytes : default is 20)							
0x06	LPF frequency (MHz: 0 means through)								AUX FIELD format # (85 = 0x55)							
0x07	filler data (0x5555)															
0x08	filler data (0x5555)															
0x09	filler data (0x5555)															
0x0A	filler data (0x5555)															
0x0B	filler data (0x5555)															
0x0C	filler data (0x5555)															
0x0D	filler data (0x5555)															
0x0E	filler data (0x5555)															
0x0F	filler data (0x5555)															

2.3.5 Format #170

Table 8. VSSP32 Format #170

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern (all 0xFF)															
0x01																
0x02	seconds from 0h UTC (17 bits)															(LSB)
0x03	2nd sync pattern (0x8C) (0x8B for VSSP)								AD bits		sampling frequency				ch	(M)
0x04	eflg	year (2 digits) (6bits : 0-63)							total day (9bits)							
0x05	major version #				minor version #				AUX FIELD size (in bytes : default is 20)							
0x06	LPF frequency (MHz: 0 means through)								AUX FIELD format # (170 = 0xAA)							
0x07	filler data (0xAAAA)															
0x08	filler data (0xAAAA)															
0x09	filler data (0xAAAA)															
0x0A	filler data (0xAAAA)															
0x0B	filler data (0xAAAA)															
0x0C	filler data (0xAAAA)															
0x0D	filler data (0xAAAA)															
0x0E	filler data (0xAAAA)															
0x0F	filler data (0xAAAA)															

2.3.6 Format #21 (extended format)

Table 9. VSSP32 Format #21 (extended format)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	sync pattern (all 0xFF)															
0x01																
0x02	seconds from 0h UTC (17 bits)															(LSB)
0x03	2nd sync pattern (0x8C) (0x8B for VSSP)								AD bits			sampling frequency *1			—	(M)
0x04	year (2 digits) (7bits)							total day (9bits)								
0x05	major version #				minor version #			AUX FIELD size (in bytes : default is 20)								
0x06	LPF frequency (MHz: 0 means through)							AUX FIELD format # (21)								
0x07	sampling frequency (MHz: 13 bits) *2													# of CH (2^n)		
0x08	any data (or 16 characters)															
0x09																
0x0A																
0x0B																
0x0C																
0x0D																
0x0E																
0x0F																

*1: this is effective only when sampling frequency in AUX field is set to 0.
*2: when this is set to 0, sampling frequency field in the conventional header portion becomes effective.

where the number of channels is given by 2^n and "n" is set in a header block. Currently the maximum number of channels is supposed to be 16, so that "n" takes only 0, 1, 2, 3, 4. In this format, the sampling frequency of less than 1 MHz is not supported.

3 Data block

3.1 Data format for native sampler

Sampled data output from the VSSP or VSSP32 sampler consists of 4-byte (32-bit) unit. Tables 10 and 11 show the relation among bit position, sample number, and channel number in a 32-bit unit.

Table 10. Data block format for 1-ch mode.

	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1ch x 1bit	sample #	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1ch x 2bit	sample #	16		15		14		13		12		11		10		9		8		7		6		5		4		3		2		1	
1ch x 4bit	sample #	8				7				6				5				4				3				2				1	
1ch x 8bit	sample #	4								3								2								1					

Table 11. Data format for 4-ch mode

	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4ch x 1bit	ch #	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
	sample #	8				7				6				5				4				3				2				1			
4ch x 2bit	ch #	4		3		2		1		4		3		2		1		4		3		2		1		4		3		2		1	
	sample #	4								3								2								1							
4ch x 4bit	ch #	4				3				2				1				4				3				2				1			
	sample #	2																1															
4ch x 8bit	ch #	4								3								2								1							
	sample #	1																															

3.2 Data block format for Format #21 (extended format).

Format #21 (extended format) supports 2, 8, and 16-ch mode besides 1 and 4-ch mode as shown in Tables 12, 13, and 14.

Table 12. Extended data format for 2-ch mode.

	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2ch x 1bit	ch #	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
	sample #	16		15		14		13		12		11		10		9		8		7		6		5		4		3		2		1	
2ch x 2bit	ch #	2		1		2		1		2		1		2		1		2		1		2		1		2		1		2		1	
	sample #	8				7				6				5				4				3				2				1			
2ch x 4bit	ch #	2				1				2				1				2				1				2				1			
	sample #	4								3								2								1							
2ch x 8bit	ch #	2								1								2								1							
	sample #	2																1															

Table 12. Extended data format for 8-ch mode.

	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8ch x 1bit	ch #	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
	sample #	4								3								2								1							
8ch x 2bit	ch #	8		7		6		5		4		3		2		1		8		7		6		5		4		3		2		1	
	sample #	2																1															
8ch x 4bit	ch #	8				7				6				5				4				3				2				1			
	sample #	1																															
8ch x 8bit	ch #	4								3								2								1							
	sample #	1																															
	bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	ch #	8								7								6								5							
	sample #	1																															

Table 14. Extended data format for 16-ch mode.

	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16ch x 1bit	ch #	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	sample #	2																1															
16ch x 2bit	ch #	16		15		14		13		12		11		10		9		8		7		6		5		4		3		2		1	
	sample #	1																															
16ch x 4bit	ch #	8				7				6				5				4				3				2				1			
	sample #	1																															
	bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	ch #	16				15				14				13				12				11				10				9			
	sample #	1																															
16ch x 8bit	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ch #	4								3								2								1							
	sample #	1																															
	bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	ch #	8								7								6								5							
	sample #	1																															
	bit position	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
	ch #	12								11								10								9							
	sample #	1																															
	bit position	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
	ch #	16								15								14								13							
	sample #	1																															

The size of data block should be an integer multiple of a 4-byte (32-bit) unit. In case when a fractional figure occurs, a rest of data block is filled by '0'. How to define the size of data block is as follows. The size of data sampled in 1 sec can be caclculated as

$$B = F \cdot A \cdot N \quad \text{[bits]}$$

where F, A, and N are sampling frequency (Hz), the number of AD bits, and the number of channels, respectevely. If $B/32$ has a fractional part (not an integer), thse size of data block is defined as follows,

$$S = (int(B/32) + 1) \times 32 \quad \text{[bits]}$$

when int is a function which returns the integer part of a specified number. Then a rest of data block starting from $B + 1$ bits is filled out by '0'. If $B/32$ is an integer, S is set to be B.