



HD/SD 8/12 Channel Mobile Video Studio HS-2800



RS-422/232
Remote Control Protocol

DATE : 2016.01.26

CONTENTS

1. Physical layer

2. Data link layer

2.1 Frame format

3. Application layer

3.1 Command data format

3.2 Return data format

4. Operated

4.1 The command group = 10h (HS-2800 Main board control)

4.1.1 The control mode = 00h (ASK status mode)

4.1.2 The return data stream from HS-2800 (ASK status mode)

4.2 The control mode = 01h (Simple control mode)

4.2.1 The control data stream

4.2.2 The return data stream from HS-2800

4.2.3 The KEY Code

4.2.4 Example

4.3 The control mode = 05h (NORMAL remote control mode)

4.3.1 The control data stream

4.3.2 The return data stream from HS-2800

4.3.3 The Command Code

4.3.4 Example

1. Physical layer

- 1.1 Control I/O format: RS-422 or RS-232C
- 1.2 HS-2800 remote RS-422 pin definition: (If use DSUB-9-female)
(Pin2=TX_P, Pin7=TX_N),(Pin3=RX_P, Pin8=RX_N), (Pin1, 5, 9=GND),
- 1.3 HS-2800 remote RS-232 pin definition: DSUB-9-female: (Pin2=TX, Pin3=RX, Pin5=GND),
- 1.4 Communication Baud rate: **115200 BPS**
- 1.5 Data format: **8** bits serial, **LSB** first, **1** start bit, **1** stop bit, **none** parity

2. Data link layer

2.1 Frame format

1st	2nd	3rd	4th	5th	6th	7th	„	Last-2	Last-1	Last
Header	ID	Length	Data0	Data1	Data2	Data3	„	Chksum_L	Chksum_H	End

1) Header

The Code consisting of one byte is for frame synchronization.

The frame header send from the **master machine (controller)** is termed the command header.

The command header byte is fixed as follow.

1st: F0h (base 16)

The frame header send from the **slave machine (HS-2800)** is termed the *return header*.

The *return header* byte is fixed as follow.

1st: FCh (base 16)

2) ID number

The equipment ID number is composed of 8 bits

The ID of HS-2800 = 28h (base 16)

3) Length

The length is the sum of bytes from the header to the end.

It is composed of 8 bits and the length must less than **128**.

Note: **06h (base 16) < Length <= 80h (base 16)**

4) Data

Data block used by application layer. Refer to Section 3 ~.

5) Checksum

The 8 bits checksum is obtain from header to the last data,
and then convert to two numeric ASCII code.

$\text{Checksum} = \text{header} + \text{ID} + \text{legth} + \text{data0} + \text{data1} + \dots + \text{last_data}$

$\text{chksum_L} = 30\text{h} + (\text{low nibble of checksum})$

$\text{chksum_H} = 30\text{h} + (\text{high nibble of checksum shift to right 4bits})$

6) End

The end byte is fixed to **FFh** (base 16).

3. Application layer

The application layer designates the command structure and contents.

3.1 Command data format (Send by the master machine)

4th	5th	6th	7th	8th	9th	10th	11th	...
Command Group	Control Mode	Operated #1	Operated #2	Operated #3	Operated #4	Operated #5	Operated #6	...

1) The command group

10h = HS-2800 main control.

2) The Control Mode

00h = ASK status mode.

01h = SIMPLE key mode.

04h = UPDATE software mode (IF necessary)

05h = NORMAL remote control mode.

3) The operated refer to the section 4~.

3.2 Return data format (Send by the slave machine)

4th	5th	6th	7th	8th	9th	10th	11th	...
Command Group	Control Mode	parameter #1	parameter #2	parameter #3	parameter #4	parameter #5	parameter #6	...

1) The Command Group

10h = HS-2800 main control.

2) The Control Mode (Same as the 3.1 command data format's control mode.)

00h = ASK status mode.

02h = T-BAR mode.

04h = UPDATE software mode (If necessary)

05h = NORMAL remote control mode.

3) The parameters refer to the follow description.

4. Operated

4.1 The command group = 10h (HS-2800 Main board control)

4.1.1 The control mode = 00h (ASK status mode): Ask the HS-2800 main set's status.

1st	2nd	3rd	4th	5th	6th	7th	8th
F0h	28h	08h	<u>10h</u>	<u>00h</u>	30h	33h	FFh
Header	ID	length	group	ASK status	cksm_L	cksm_H	end

4.1.2 The return data stream from HS-2800 (ASK status mode)

1st	2nd	3rd	4th	5th	6th	7th	8th	9th
FCh	28h	xxh	<u>10h</u>	<u>00h</u>	Parameter #1	Parameter #2	Parameter #3	Parameter #4
		length						

10th ~ 15th	16th	17th	18th
Parameter #5~ 10	cksm_L	cksm_H	FFh

4.1.2.1 *parameter-1* = busy flags

bit0=busy in transition

bit1= T-bar is active

bit2=HD(0) or SD (1)

bit3=NTSC (0) or PAL (1)

bit4=PROG source be frozen

bit5=PST source be frozen

bit6=fade to black function is active

bit7=0;

4.1.2.2 *parameter-2* = transition speed , LOGO & timer

bit0~2= Speed1~8, 0=speed1, 1=speed2, 2=speed3

bit3=logo_1 on/off, 1=on

bit4=logo_2 on/off, 1=on

bit5=timer on/off, 1=on

bit6=clock on/off, 1=on

bit7=0;

4.1.2.3 ***parameter-3*** = current transition effect

bit0~3: 0=MIX (dissolve) 1~6=wipe_1 ~ wipe_6

bit4=INV

bit5=SOFT

bit6=FREEZE

bit7=0;

4.1.2.4 ***parameter-4*** = current PRESET source

0=> PST source= background color

1~12=> PST source=CH1~CH12

13 => PST source= FS

14 => PST source= color bar

bit7=0;

4.1.2.5 ***parameter-5*** = current PROGRAM source

0=> PGM source= background color

1~12=> PGM source=CH1~CH12

13=> PGM source= FS

14 => PGM source= color bar

bit7=0;

4.1.2.6 ***parameter-6*** = current preset PIP & DSK status

bit0= PST PIP1 on/off, 1=on

bit1= PST PIP2 on/off, 1=on

bit2= PST DSK1 on/off, 1=on

bit3= PST DSK2 on/off, 1=on

bit7=0;

4.1.2.7 ***parameter-7*** = current program PIP & DSK status

bit0= PGM PIP1 on/off, 1=on

bit1= PGM PIP2 on/off, 1=on

bit2= PGM DSK1 on/off, 1=on

bit3= PGM DSK2 on/off, 1=on

bit7=0;

4.1.2.8 ***parameter-8*** = some special effect' s status

bit0 = AUX select active

bit1 = SET key active

bit2 = PC control active

bit7 = 0

4.1.2.9 ***parameter-9*** = T-BAR position high bits

bit5~bit0=T-bar bit11~bit6 (if only 8 bits resolution,use the bit11~bit4)

bit7= 0;

4.1.2.10 ***parameter-10*** = T-BAR position low bits

(bit5~bit0)=T-bar bit5~bit0

bit7= 0;

4.2 The control mode = 01h (SIMPLE key control)

Control the HS-2800 main board's function by simple key code. (like as a keyboard controller)

4.2.1 The control data stream

1st	2nd	3rd	4th	5th	6th	7th	8th	9th
F0h	28h ID	09h	10h group	01h Simple key CTRL mode	<i>Key code</i>	cksm_L	cksm_H	FFh

4.2.2 The return data stream from HS-2800

The HS-2800 return the HS-2800's status same as 4.1.2 normally.

4.2.3 The Simple Key code

- | | |
|---------------------------|-----------------------|
| 1 = KEY_PRESET_1 | 2 = KEY_PRESET_2 |
| 3 = KEY_PRESET_3 | 4 = KEY_PRESET_4 |
| 5 = KEY_PRESET_5 | 6 = KEY_PRESET_6 |
| 7 = KEY_PRESET_7 | 8 = KEY_PRESET_8 |
| 9 = KEY_PRESET_9 | 10 = KEY_PRESET_10 |
| 11 = KEY_PRESET_11 | 12 = KEY_PRESET_12 |
| 13 = KEY_PRESET_FS | 14 = KEY_PRESET_BG |
| 15 = KEY_PRESET_COLORBAR | |
| 16 = KEY_CUT | 17 = KEY_AUTO |
| 18 = KEY_FT B | |
| 19 = KEY_PROGRAM_1 | 20 = KEY_PROGRAM_2 |
| 21 = KEY_PROGRAM_3 | 22 = KEY_PROGRAM_4 |
| 23 = KEY_PROGRAM_5 | 24 = KEY_PROGRAM_6 |
| 25 = KEY_PROGRAM_7, | 26 = KEY_PROGRAM_8 |
| 27 = KEY_PROGRAM_9 | 28 = KEY_PROGRAM_10 |
| 29 = KEY_PROGRAM_11 | 30 = KEY_PROGRAM_12 |
| 31 = KEY_PROGRAM_FS | 32 = KEY_PROGRAM_BG |
| 33 = KEY_PROGRAM_COLORBAR | |
| 34 = KEY_SPEED_1 | 35 = KEY_SPEED_2 |
| 36 = KEY_SPEED_3 | |
| 37 = KEY_PRESET_PIP_1 | 38 = KEY_PRESET_PIP_2 |

39 = KEY_PRESET_DSK_1

41 = KEY_PROG_PIP_1

43 = KEY_PROG_DSK_1

45 = KEY_LOGO_1

47 = KEY_TIMER

49 = KEY_AUX

50 = KEY_WIPE_1

52 = KEY_WIPE_3

54 = KEY_WIPE_5

56 = KEY_WIPE_INV

58 = KEY_WIPE_SOFT

60 = KEY_SET

62 = KEY_MENU_UP

64 = KEY_MENU_LEFT

66 = KEY_MENU_ENTER

68 = KEY_AUDIO_UP

40 = KEY_PRESET_DSK_2

42 = KEY_PROG_PIP_2

44 = KEY_PROG_DSK_2

46 = KEY_LOGO_2

48 = KEY_CLOCK

51 = KEY_WIPE_2

53 = KEY_WIPE_4

55 = KEY_WIPE_6

57 = KEY_WIPE_MIX

59 = KEY_WIPE_FREEZE

63 = KEY_MENU_DOWN

65 = KEY_MENU_RIGHT

67 = KEY_PC_CONTROL

69 = KEY_AUDIO_DOWN

4.2.4 The example about the simple key control

Example#1: **Switch program out to color bar**

The Controller send: *F0h, 28h, 09h, 10h, 01h, 21h, 33h, 35h, FFh*

The HS-2800 respond the all status after completed the command:

FCh, 28h, 12h, 10h, 00h, 08h, 01h, 12h, 03h, 0Eh, 05h, 0ch, 00h, 00h, 00h, 33h, 38h, FFh

Byte1=FCh=> Header from slave machine
Byte2=28h=> HS-2800 ID
Byte3=12h=> The length of data stream=18 bytes
Byte4=10h=> HS-2800 command group
Byte5=00h=> Ask statusmode
Byte6=08h=> T-BAR off, HD mode, PAL system
Byte7=01h=> Speed=2, logo & timer all off
Byte8=12h => Wipe2, soft on,
Byte9=03h=> PST=ch3,
Byte10=0eh=> PGM=colorbar
Byte11=05h=> PST PIP1=on, PST DSK1=on
Byte12=0ch=> PGM PIP2=on, PGM DSK2=on
Byte13=00h=> AUX, SET, PC_control all off
Byte14=00h => T-BAR high bits
Byte15=00h => T-BAR low bits
Byte16=33h=> cksm_L
Byte17=38h=> cksm_H
Byte18=FFh=> END

4.3 The control mode = 05h (NORMAL remote control)

Control the HS-2800 main board's functions.

4.3.1 The control data stream

1st	2nd	3rd	4th	5th	6th	7th	8th
F0h	28h ID	0fh	10h group	05h Normal remote CTRL	<u>Command</u> <u>code</u>	<u>content</u> <u>#1</u>	<u>content</u> <u>#2</u>

9th	10th	11th	12th	13th	14th	15th
<u>content#3</u>	<u>content#4</u>	<u>content#5</u>	<u>content#6</u>	cksm_L	cksm_H	FFh

4.3.2 The return data stream from HS-2800

1st	2nd	3rd	4th	5th	6th	7th	8th
FCh	28h ID	0fh	10h group	05h Normal remote CTRL	<u>Command</u> <u>code</u>	<u>content</u> <u>#1</u>	<u>content</u> <u>#2</u>

9th	10th	11th	12th	13th	14th	15th
<u>content#3</u>	<u>content#4</u>	<u>content#5</u>	<u>content#6</u>	cksm_L	cksm_H	FFh

4.3.3 The Key Command Code

00h = Set system mode (HD/SD mode, 59.94/50Hz, SYNC. mode)

01h = Set T-BAR position

02h = Set program out source

03h = Set preset out source

04h = Set AUX1 out source

05h = Set LOGO & TIMER on/off

06h = Set transition effect

07h = Set active speed No.

08h = Set the time of speed

09h = Set PIP on preset or program out

0Ah = Set PIP parameter

0Bh = Set DSK on prset or program out

0Ch = Set DSK parameter

0Dh = Set freeze

0Eh = Set output port mode

0Fh = Set Multi. Screen mode

10h = CUT
11h = AUTO
12h = Fade to black
13h = Set PC control on /off
14h = Set GPI mode
15h = Set Video in mode & color cali
16h = Set audio level
17h = TBD
18h = TBD
; TBD

4.3.4 The Command content

1) Set system mode (code=00h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= **HD/SD** mode, 00h=HD, 01h=SD

Content#3= **59.94 /50** Hz, 00h=59.94Hz, 01h=50Hz

Content#4= **SYNC**. Mode, 00h=internal, 01h=external

(Note: bit7=0 always)

2) Set T-BAR position (code=01h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= **T-bar on/off**, 00h=off, 01h=active

Content#3(bit5~bit0)=T-bar bit11~bit6

Content#4(bit5~bit0)=T-bar bit5~bit0

(If only 8 bits resolution, use the bit11~bit4)

(Note: bit7=0 always)

3) Set program out source (code=02h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= **source_No**, 00h~xxH=BG, CH1~CH12,,

(Note: bit7=0 always)

4) Set preset source (code=03h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= **source_No**, 00h~xxH=BG, CH1~CH12,,

(Note: bit7=0 always)

5) Set Aux1 out source (code=04h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= **source_No**, 00h~xxH=BG, CH1~CH12,,

(Note: bit7=0 always)

6) Set LOGO & timer on/off (code=05h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= bit0= logo1 on/off, 1=on

Bit1= logo2 on.off

Bit2= Timer on/off

Bit3= Clock on/off

(Note: bit7=0 always)

7) Set transition effect (code=06h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= Bit0~2 = wipe_no, 0=mix

Bit3 =INV

Content#3= soft border, 0=off, 1=on

Content#4= freeze (For what ?)

(Note: bit7=0 always)

8) Set active speed No (code=07h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2=current active speed no, 00h=speed1, 01h=speed1, 02h=speed2

(Note: bit7=0 always)

9) Set the time of speed (code=08h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2=4 low bits of speed1, 00h~0Fh

Content#3=4 high bits of speed1, 00h~0Fh,

speed1= (content#3<<4 + content#2) frames

Content#4=4 low bits of speed1, 00h~0Fh

Content#5=4 high bits of speed1, 00h~0Fh,

speed2= (content#5<<4 + content#4) frames

Content#6=4 low bits of speed1, 00h~0Fh

Content#7=4 high bits of speed1, 00h~0Fh,

speed3= (content#7<<4 + content#6) frames

(Note: bit7=0 always)

10) Set PIP on preset or program out (code=09h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= PIP1 =>bit0= on preset, bit1 = on program out

Content#3= PIP2 =>bit0= on preset, bit1 = on program out

11) Set PIP parameter (code=0Ah)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= PIP1 size => **TBD**

Content#3= PIP2 size => **TBD**

Content#4= PIP1 position_x0 => **TBD**

Content#5= PIP1 position_x1 => **TBD**

Content#6= PIP1 position_y0 => **TBD**

Content#7= PIP1 position_y1 => **TBD**

Content#8= PIP2 position_x0 => **TBD**

Content#9= PIP2 position_x1 => **TBD**

Content#10= PIP2 position_y0 => **TBD**

Content#11= PIP2 position_y1 => **TBD**

(Note: bit7=0 always)

12) Set DSK on preset or program out (code=0Bh)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= DSK1 =>bit0= on preset, bit1 = on program out

Content#3= DSK2 =>bit0= on preset, bit1 = on program out

(Note: bit7=0 always)

13) Set DSK parameter (code=0Ch)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= DSK1_LEVEL =>**TBD**

Content#3= DSK? _??? =>**TBD**

Content#4= DSK? _??? =>**TBD,,,**

(Note: bit7=0 always)

14) Set freeze (code=0Dh)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= =>**TBD**

(Note: bit7=0 always)

15) Set output port mode (code=0Eh)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2=OUT1 mode, 00h=program, 01h=preset, 02h=PGM w/o logo,
03h=PGM w/o logo & w/o subtitles
10h~1Bh= one of the input signal

Content#3=OUT2 mode, 00h=program, 01h=preset, 02h=PGM w/o logo,
03h=PGM w/o logo & w/o subtitles
10h~1Bh= one of input the input signal

Content#4=OUT3 mode, 00h=program, 01h=preset, 02h=PGM w/o logo,
03h=PGM w/o logo & w/o subtitles
10h~1Bh = one of the input signal

20h = SD signal that is downscaled from HD program out

(Note: bit7=0 always)

16) Set Multi. Screen mode (code=0Fh)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2=Multi. Screen mode, 00h=way_1, 01h=way_2, 02h=way_3

(Note: bit7=0 always)

17) CUT (code=10h)

No content

(Note: bit7=0 always)

18) AUTO (code=11h)

No content

(Note: bit7=0 always)

19) Fade to black (code=12h)

No content

(Note: bit7=0 always)

20) Set PC control on /off (code=13h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= PC control on/off, 00h=off, 01h=ON

(Note: bit7=0 always)

21) Set GPI mode (code=14h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= GPI trigger mode => **TBD**

Content#3= delay time => **TBD**

Content#4 => TBD

(Note: bit7=0 always)

22) Set Video in mode & color calibration (code=15h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2=**Ch No**, 00h~0Bh=CH1~CH12

Content#3=**input mode**, 00h=HD-SDI, 01h=SD-SDI,

02h=CV, 03h=HDMI, 04h=static picture

Content#4=**Brightness**, 0~+/-63, normal=0=60h;

Content#5=**Contrast**, 0~+/-63, normal=0=60h;

Content#6=**Color Saturation**, 0~+/-63, normal=0=60h;

(Note: bit7=0 always)

23) Set audio level (code=16h)

Content#1= **ASK or SET**, 00h=ASK, 01h=SET

Content#2= **TBD**

4.3.4 Example

Exampe#0.1 Ask system mode (command=0, content#1=0)

Controller: *F0h,28h,0Ah,10h,05h, 00h,00h, 36h,33h,FFh*

HS-2800:FCh,28h,0Dh,10h,05h,00h,00h,00h,01h,00h,37h,34h,FFh

HD mode , 50 Hz, internal SYNC.

Exampe#0.2 Set system mode to SD & 59.94Hz (command=0, content#1=1)

Controller: *F0h,28h,0Dh,10h,05h, 00h,01h,01h,00h,00h, 3Ch,33h,FFh*

HS-2800:FCh,28h,0Dh,10h,05h,00h,01h,01h,00h,00h,38h,34h,FFh

SD mode , 59.9 Hz, internal SYNC.

Example#1.1: Ask the T-BAR position (command=01h, content#1=0)

Controller: *F0h,28h,0Ah,10h,05h, 01h,00h, 38h,33h,FFh*

HS-2800:FCh,28h,0Dh,10h,05h,01h,00h,01h,20h,20h,38h,38h,FFh

T-bar active, T-bar_position=820H(12bits)=82h(8bit)=130

Example#1.2: Set the T-BAR position to 130 (8 bits) (command=01h, content#1=1)

Controller: *F0h,28h,0Dh,10h,05h, 01h,01h,01h,20h,20h, 3Ch,37h,FFh*

HS-2800:FCh,28h,0Dh,10h,05h,01h,00h,01h,20h,20h,38h,38h,FFh

T-bar active, T-bar_position=820H(12bits)=82h(8bit)=130

Example#2.1: Ask program out source (command=02h, content#1=0)

Controller: *F0h,28h,0Ah,10h,05h, 02h,00h, 39h,33h,FFh*

HS-2800:FCh,28h,0Bh,10h,05h,02h,00h,03h,39h,34h,FFh

Program out = CH3

Example#2.2: Set program out to CH5 (command=02h, content#1=1)

Controller: *F0h,28h,0Bh,10h,05h, 02h,01h,05h, 30h,34h,FFh*

HS-2800:FCh,28h,0Bh,10h,05h,02h,01h,05h,3Ch,34h,FFh

Program out = CH5

Example#3.1 Set preset to background color (command=03h, content#1=1)

Controller: *F0h,28h,0Bh,10h,05h, 03h,01h,00h, 3Ch,33h,FFh*

HS-2800:FCh,28h,0Bh,10h,05h,03h,01h,00h,38h,34h,FFh

Preset = BG

Example#5.1 Ask logo & timer on/off (command=05h, content#1=0)

Controller: *F0h,28h,0Ah,10h,05h, 05h,00h, 3Ch,33h,FFh*

HS-2800:FCh,28h,0Bh,10h,05h,05h,00h,09h,32h,35h,FFh

Logo1 on, logo2 off, timer off, clock on

Example#5.2 Set logo2 on (command=05h, content#1=1)

Controller: *F0h,28h,0Bh,10h,05h, 05h,01h,02h, 30h,34h,FFh*

HS-2800:FCh,28h,0Bh,10h,05h,05h,01h,02h,3Ch,34h,FFh

Logo1 off, logo2 on, timer off, clock off

Example#6.1 Ask transition effect (command=06h, content#1=0)

Controller: *F0h,28h,0Ah,10h,05h, 06h,00h, 3Dh,33h,FFh*

HS-2800:FCh,28h,0Dh,10h,05h,06h,00h,0ah,01h,00h,37h,35h,FFh

Wipe_2, INV, soft border

Example#6.2 Set wipe6, not INV (command=06h, content#1=1)

Controller: *F0h,28h,0Dh,10h,05h, 06h,01h,06h,00h,00h, 37h,34h,FFh*

HS-2800:FCh,28h,0Dh,10h,05h,06h,01h,06h,00h,00h,33h,35h,FFh

Wipe_6, not INV, soft off

Example#7.1 Set to speed 2 (command=07h, content#1=1)

Controller: *F0h,28h,0Bh,10h,05h, 07h,01h,01h, 31h,34h,FFh*

HS-2800:FCh,28h,0Bh,10h,05h,07h,01h,01h,3Dh,34h,FFh

Speed_2

Exampe#8.1 Ask the time of speeds (command=08h, content#1=0)

Controller: *F0h,28h,0Ah,10h,05h, 08h,00h, 3Fh,33h,FFh*

HS-2800:FCh,28h,10h,10h,05h,08h,00h,01h,00h,02h,00h,05h,00h,39h,35h,FFh

Speed1= 1 frame, speed2=2 frames, speed3= 5 frames

Exampe#9.1 Set PIP1 & PIP2 on program out (command=09h, content#1=1)

Controller: *F0h,28h,0Ch,10h,05h, 09h,01h,02h,02h, 37h,34h,FFh*

HS-2800:FCh,28h,0Ch,10h,05h,09h,01h,02h,02h,33h,35h,FFh

PIP1 on program out, PIP2 on program out

Exampe#11.1 Set DSK1 & DSK2 on preset & program out

(Command= 0Bh, content#1=1)

Controller: *F0h,28h,0Ch,10h,05h, 0Bh,01h,03h,03h, 3bh,34h,FFh*

HS-2800:FCh,28h,0Ch,10h,05h,0Bh,01h,03h,03h,37h,35h,FFh

DSK1 on preset & program out, DSK2 on preset & program out

Exampe#14.1 Ask the output por mode (command=0Eh, content#1=0)

Controller: *F0h,28h,0Ah,10h,05h, 0Eh,00h, 35h,34h,FFh*

HS-2800:FCh,28h,0Dh,10h,05h,0Eh,00h,00h,03h,20h,37h,37h,FFh

Out1= program out

Out2= program without logo & without

subtitles Out3= SD signal (downscaled)

Exampe#14.2 Set the OUT1=PGM, OUT2=PST, OUT3=CH4

(Command=0Eh, content#1=1)

Controller: *F0h,28h,0Dh,10h,05h, 0Eh,01h,00h,01h,13h, 3dh,35h,FFh*

HS-2800:FCh,28h,0Dh,10h,05h,0Eh,00h,00h,01h,13h,39h,36h,FFh

Out1= program out, Out2=

preset Out3= video in CH4

Example#17.1: **take Auto transition**

Controller: *F0h,28h,09h,10h,05h,11h,38h,34h,FFh*

*HS-2800:FCh, 28h, 12h, 10h,00h, 08h, 01h, 12h,03h,0Eh,05h, 0ch,00h, 00h, 00h,
33h, 38h, FFh*

Byte1=FCh=> Header from slave machine
Byte2=28h=> HS-2800 ID
Byte3=12h=> The length of data stream=18 bytes
Byte4=10h=> HS-2800 command group
Byte5=00h=> Ask statusmode
Byte6=08h=> T-BAR off, HD mode, PAL system,
Byte7=01h=> Speed=2, logo & timer all off
Byte8=12h=> Wipe2, soft on,
Byte9=03h=> PST=ch3,
Byte10=0eh=> PGM=colorbar
Byte11=05h=> PST PIP1=on, PST DSK1=on
Byte12=0ch=> PGM PIP2=on, PGM DSK2=on
Byte13=00h=> AUX, SET, PC_control all off
Byte14=00h=> T-BAR high bits
Byte15=00h=> T-BAR low bits
Byte16=33h=> cksm_L
Byte17=38h=> cksm_H
Byte18=FFh=> END

Example#18.1: **Fade program out to black**

Controller: *F0h,28h,09h,10h,05h,12h,38h,34h,FFh*

*HS-2800:FCh, 28h, 12h, 10h,00h, 48h, 01h, 12h,03h,0Eh,05h, 0ch,00h, 00h, 00h,
33h, 3Ch, FFh*

Byte1=FCh=> Header from slave machine
Byte2=28h=> HS-2800 ID
Byte3=12h=> The length of data stream=18 bytes
Byte4=10h=> HS-2800 command group
Byte5=00h=> Ask statusmode
*Byte6=48h=> T-BAR off, HD mode, PAL system, **fade_to_black ACTIVE***
Byte7=01h=> Speed=2, logo & timer all off
Byte8=12h=> Wipe2, soft on,
Byte9=03h=> PST=ch3,
Byte10=0eh=> PGM=colorbar
Byte11=05h=> PST PIP1=on, PST DSK1=on
Byte12=0ch=> PGM PIP2=on, PGM DSK2=on
Byte13=00h=> AUX, SET, PC_control all off
Byte14=00h=> T-BAR high bits
Byte15=00h=> T-BAR low bits
Byte16=33h=> cksm_L
Byte17=3Ch=> cksm_H
Byte18=FFh=> END