

Communication protocol

Using this protocol, it is possible to address up to 256 slaves from one master. Due to the fact that this is normally not supported by RS232 type communications, special features are incorporated into the slave.

As a start, each slave should have a unique address.

Bytes transmitted by master

When transmitting data from the master to the slave(s), the start of the transmission is indicated using the STX (02) byte. All the slaves in the chain are constantly checking if there is a STX byte on the serial line. After receiving STX, an address byte is transmitted by the master. All the slaves compare this address byte with their own address. The slave that recognises his address will interpret the following bytes and do what is requested by the master. The other slaves will restart checking for STX on the serial line.

If 02 appears in the data bytes following STX, the slaves which did not recognise their address will be confused. They might even interpret the following bytes in a wrong way, reacting unpredictably.

To avoid this, a protocol is used which avoids the appearance of 02 (STX) in the data bytes sent by the master.

The protocol introduces an offset byte in the transferred bytes (ADR being the byte indicating the address of the projector for which the data is meant and DATi the data bytes):

STX BYTE1 ... BYTE_n
becomes :
STX BYTE1_e ... BYTE_n_e

BYTE_i_e are bytes which are "encoded" in a way that none of them is equal to 02. Their relation with the original bytes is :

$$\text{BYTE}_{i_e} = \text{BYTE}_i + \text{OFFS}$$

Because the number of bytes used in a transmission is always smaller than 254, a byte OFFS can always be found, so that all the encoded bytes and OFFS are different from 02.

Here is a C source example of how to find OFFS and Checksum :

```
#define STX '\x02'

unsigned char find_offacs (unsigned char *pstart_loc)
{ /* find offset and check sum */
  /* pstart_loc : pointer to record to be scanned */

  int count;
  int check[256]; /* check[i] == TRUE means code 'i' occurs in record */
  unsigned checksum;

  checksum = 0; /* initialize checksum */
  for (count = 0; count <= 255; count++) /* init code checking array */
    check[count] = FALSE;
  for (count = 0; count < 6; count++) /* scan codes in record */
  { check[pstart_loc[count]] = TRUE; /* mark in checking array */
    checksum += pstart_loc[count]; } /* build checksum value */
  checksum %= 256; /* keep least significant byte */
  check[checksum] = TRUE; /* mark in checking array */
  for (count = 1; check[count] == TRUE; count++);
  /* search checking array */
  /* skip value 0 and continue as long as code 'count' is marked TRUE */
```

```

    pstart_loc[6] = (byte) checksum;      /* store checksum in record */
    return ((unsigned char) (STX-count));
                                        /* offset=STX-not_occuring_code */
}

```

There are two formats of data transfers :

- a short one : called "record"
 - a long one : called "block"
- (this format is currently not used for ACTAS purposes)

The short format, record, consists of 7 bytes :

```

ADR   Slave address
CMD   Command
DAT1  Data byte 1
DAT2  Data byte 2
DAT3  Data byte 3
DAT4  Data byte 4
CHKS  Check sum

```

CMD is the command given to the slave with address ADR.

DAT1-4 can contain additional information (explained later in this text), however, this information is not always used. It is advised to set all non used data bytes to 0 (zero).

CHKS is :

$(ADR + CMD + DAT1 + DAT2 + DAT3 + DAT4) \text{ MOD } 256$

The transmission of the record on the serial line will be:

STX OFFS ADRe CMDe DAT1e DAT2e DAT3e DAT4e CHKSe

The long format, block, consists of a record of 9 bytes followed by either 208, 224 or 240 other data bytes plus their check sum.

Slave type	block length
800, 801, 1100, 1101, 1600, 500	208
701	224
808, 1208, 1209	240
ACTAS	- (reserved for future use)

The transmission will be :

STX OFFS ADRe CMDe DAT1e DAT2e DAT3e DAT4e CHKSe

D0e D1e ... D207e CHKSBe

with

$D_{ie} = D_i + \text{OFFS}$

$\text{CHKSBe} = (\text{CHKSb} + \text{OFFS}) \text{ MOD } 256$

$\text{CHKSb} = (\text{SUM } (i = 0 \text{ to } 207) D_i) \text{ MOD } 256$

Bytes expected back from slave

2 Main types of slave response are possible :

- no answer data bytes required;
- answer data bytes required.

In any case, at least a handshake byte has to be returned.

Note that the slave only transmits upon request from the master.

Handshake byte :

Because only the signals RxD and TxD are used for the communication, there is no way to detect if there is a slave listening and (if there is one listening) if it receives the data correctly. Therefore, the communication protocol expects a standard ASCII byte ACK or NAK (06H or 15H) from the slave.

It is very important that when the master asks the slave for information that it is ready to immediately receive the data from the slave. Otherwise data loss or overruns could occur, or after a "time out", the master could think that there is no slave responding.

Data bytes :

If answer data is required, this can be either just a record, or a record followed by a block.

No answer data from slave required

For the record format (short) the slave returns an ACK byte if it has received the data correctly or a NAK byte if the received data is not correct (NOTE : check sum is not an executable command). If there is no slave responding, then no byte is returned.

Master to slave :

RECORD

Slave to master :

ACK or NAK

For the block format (long) the slave returns an ACK (or NAK) after the first part (the record) and a second ACK (or NAK) after the second part.

Master to slave :

RECORD

BLOCK

Slave to master :

ACK or NAK

ACK or NAK

Answer data from slave required

Suppose the master asks for information to the slave and the slave returns the information in record format, the bytes transmitted on the RS232 bus look like :

record from master to slave :

STX OFFS ADRPe CMDe Dat1e Dat2e Dat3e Dat4e CHKSe

record from slave to master :

ADRP CMD Dat1b Dat2b Dat3b Dat4b CHKSb

with :

ADRP = ADRPe - OFFS

CMD = CMDe - OFFS

The slave will always return its address and the received command to the master.

Master to slave :

RECORD

Slave to master :

ACK

RECORD

For a block return, the same thing happens. But the record is followed by the 209, 225 or 241 bytes (208, 224 or 240 data bytes plus their check sum).

Master to slave :

RECORD

Slave to master :

ACK

RECORD

BLOCK

RS232 PROJECTOR COMMANDS

PAD 8759

See also : "Communication protocol.doc".

This document can be used to update or generate manuals and other publications, but not vice versa!

This revision is only meant for the BR909 projectors.

Contents

Contents

2

ADDR	01H	4
ENTER	07H	4
EXIT	08H	4
ADJUST	09H	4
TEXT	0DH	4
STDBY	0EH	4
PARK	0FH	4
NUM_0	10H	4
NUM_1	11H	4
NUM_2	12H	4
NUM_3	13H	4
NUM_4	14H	4
NUM_5	15H	4
NUM_6	16H	4
NUM_7	17H	4
NUM_8	18H	4
NUM_9	19H	4
FREEZE	1BH	4
MONO	1DH	4
EXPAND	1EH	4
MUTE	1FH	4
CUR_UP	21H	4
CUR_DN	22H	4
CUR_RI	23H	4
CUR_LE	24H	4
TOGG_A	27H	5
CON_UP	28H	5
CON_DN	29H	5
BRI_UP	2AH	5
BRI_DN	2BH	5
SAT_UP	2CH	5
SAT_DN	2DH	5
HUE_UP	2EH	5
HUE_DN	2FH	5
F_1	31H	5
F_2	32H	5
F_3	33H	5
F_4	34H	5
F_5	35H	5
SHA_UP	36H	5
SHA_DN	37H	5
VOL_UP	38H	5
VOL_DN	39H	5
BAS_UP	3AH	5
BAS_DN	3BH	5
TRE_UP	3CH	5
TRE_DN	3DH	5
BAL_UP	3EH	5
BAL_DN	3FH	6
SEL_SOUND	40H	6
SEL_PICT	41H	6
E2_COPY	42H	7
READ_MEAS	48H	8
READ_VERSION_ID	49H	9
READ_SOURCE	4AH	9
READ_STATUS	4BH	11

READ_VERSION	4CH	12
READ_SERNR	4DH	13
CONTROL_TEXT	4EH	13
LOCK_IR	50H	13
FORCE_TOGG	51H	Error!
Bookmark not defined.		
READ_TIME	60H	14
SET_INT_PATT	61H	15
CONTROL_IRIS	62H	16
CONTROL_RCVDS	62H	16
CONTROL_ACTAS	62H	17
BLK_PC_E2	71H	19
BLK_E2_PC	72H	20
BLK_DEL	73H	21
FORCE_BLK	74H	19
IDENT	77H	Error!
Bookmark not defined.		
REQ_SHAPE	78H	22
WRITE_POT	79H	23
READ_POT	7AH	23

Reserved addresses : _____ **23**

ADDR 01H

This code is emitted by the infra red remote control when the address of the remote control is programmed. All projectors react by displaying the address they are working on. This is not usable for RS232 communication, because in a chain of projectors only the projector with the correct address will display its address.

ENTER 07H

EXIT 08H

ADJUST 09H

TEXT 0DH

STDBY 0EH

PARK 0FH

NUM_0 10H

NUM_1 11H

NUM_2 12H

NUM_3 13H

NUM_4 14H

NUM_5 15H

NUM_6 16H

NUM_7 17H

NUM_8 18H

NUM_9 19H

FREEZE 1BH

MONO 1DH

Toggle mono / stereo

EXPAND 1EH

Toggle normal / expand

MUTE 1FH

Mute of the sound

CUR_UP 21H

CUR_DN 22H

CUR_RI 23H

CUR_LE 24H

TOGG_A	27H	Toggles from picture adjustments (brightness...) to sound adjustments (volume...). These are useful for remote control units made for retro projectors.
CON_UP	28H	
CON_DN	29H	
BRI_UP	2AH	
BRI_DN	2BH	
SAT_UP	2CH	
SAT_DN	2DH	
HUE_UP	2EH	
HUE_DN	2FH	
F_1	31H	Function key F1 (Blue remote control)
F_2	32H	Function key F2 (Blue remote control)
F_3	33H	Function key F3 (Blue remote control)
F_4	34H	Function key F4 (Blue remote control)
F_5	35H	Function key F5 (Blue remote control)
SHA_UP	36H	
SHA_DN	37H	
VOL_UP	38H	Volume control
VOL_DN	39H	Volume control
BAS_UP	3AH	Bass control
BAS_DN	3BH	Bass control
TRE_UP	3CH	Treble control
TRE_DN	3DH	Treble control
BAL_UP	3EH	Balance control

BAL_DN **3FH**
Balance control

SEL_SOUND **40H**
Force to picture

SEL_PICT **41H**
Force to sound

E2_COPY 42H

This function copies the cut off and gain settings

Offers the possibility to copy the custom cut off and gain settings to one of the "fixed" colour temperature settings (3200K, 6500K, 4900K or 9300K)

Bytes sent :

Dat1 = destination, Dat2 = source
Dat1/Dat2meaning
00H custom cut off settings
01H custom gain settings
02H 3200K cut offs
03H 3200K gains
04H 6500K cut offs
05H 6500K gains
06H 9300K cut offs
07H 9300K gains
08H 4900K cut offs
09H 4900K gains
0AH "custom gamma" (custom midlights & blue corr)
0BH "factory gamma" (fixed midlights & blue corr)

0CH custom Blue midpoint focus
0DH not used
0EH factory Blue midpoint focus (lowest frequency 15 or 32 Kc)
0FH factory Blue midpoint focus (ACTAS frequency 64 KC)

Dat3 = length (normally 3, for focus normally 1)

Dat4 = mode

0 E2 to E2
1..255 reserved

LEVEL_PROG 46H

Information sent : Dat1 = level, Dat2 = code LO, Dat3 = code HI, Dat4 = 0
Level from 0 to 7
Code = classified information

Information sent : Dat1 = 8, Dat2 = 0, Dat3 = 0, Dat4 = 0

Information returned : Dat1 = level, Dat2 = code LO, Dat3 = code HI, Dat4 = status
Level = current level
Code = classified information
Status = free(0) or blokked (1)

FLASH_PROG 47H

This command controls the programming of the Flash memory.

Info sent :	Dat1	Dat2	Dat3	Dat4	
	0	0	0	0	start flash programming in Low Flash (normal program)
	0	1	0	0	start flash programming in High Flash (configuration files)
	0	0	0	1	erase flash
	0	0	0	2	stop flash program
	0	0	0	3	restart flash programming
	FF	0	0	0	ask status ready

The returned bytes on the last command are :

	Dat1	Dat2	Dat3	Dat4	
	FF	0	0	0	ready for new command
	FF	0	0	1	busy
From V 2.20	FF	0	0	2	error
	FF	0	0	3	not programming

READ_MEAS 48H

Purpose : questions ACTAS Control Unit measurement values.

Cmd	Dat1	Dat2	Dat3	Dat4
48H	Which	Color	Type	Option

Dat4 Option

00H	last measured values are returned (immediate answer; 7 byte record)
10H	a new measurement is done first; with normal sensitivity (answer is ACTAS' initiative; full format 9 byte record starting with STX OFFS FEH,...)
11H	a new measurement is done first; with sensitivity/10(answer as for 10H)

Dat3 Type

00H	default
01H	signed long integer (MSB returned in Dat1, ..., LSB in Dat4)
02H	IEEE 754 floating point (Dat1 Dat2 Dat3 Dat4 return MMMMMMMM MMMMMMMM EMMMMMMM SEEEEEEE)

Dat1	Dat2	Dat3	function	default type
00H	00H	00H	reserved	
01H	00H	00..02H	full_lux	IEEE 754 float
02H	00H	00..02H	peak_lux	IEEE 754 float
03H	00H..02H	00H	lux[Dat2]	IEEE 754 float [R, G, B]
10H	00H..02H	00H	colval[Dat2]	meas. in counts, long int; [R, G, B]
11H	00H..02H	00H	colref[Dat2]	reference in counts, long int; [R, G, B]

Remark : returned values in lux are internally recalculated according to the selected range and the white box size; the values in counts aren't.

READ_VERSION_ID 49H

Information sent : Dat1=0, Dat2=0, Dat3=0, Dat4=0

Information returned :

identification number :

$2^{24} \cdot \text{Dat4} + 2^{16} \cdot \text{Dat3} + 2^8 \cdot \text{Dat2} + 2^0 \cdot \text{Dat1}$

This identification number is unique and enables one to distinguish customized releases of the same version.

READ_SOURCE 4AH

This command asks the projector to return the currently selected source and attributes.

Info sent : Dat1 = 0, Dat2 = 0, Dat3 = 0, Dat4 = 0

The returned bytes hold the following information :

Dat1 Horizontal period Hi byte
Dat2 Horizontal period Lo byte
Dat3 Source number
Dat4 Source type and installation :
bits 0,1 = installation :
0 = Front / Table
1 = Rear/ Ceiling
2 = Rear / Table
3 = Front/ Ceiling
bit 2 : set if HDTV
bits 4,5,6,7 :
0 = Video 8 = Forced video
1 = SVHS 9 = Forced SVHS
2 = 2nd RGB A = Forced 2nd RGB
3 = RGB B = Forced RGB
4 = Component V C = Forced Component V
5 = Tri Level RGB
6 = Tri Level Component Video
7 = Internal

Horizontal period = ((dat1 * 256) + dat2) * 31.25 nsec.

Info sent : Dat1 = 1, Dat2 = 0, Dat3 = 0, Dat4 = 0

The returned bytes hold the following information :

Dat1 Vertical period Hi byte
Dat2 Vertical period Lo byte
Dat3 don't care
 from V 2.20 memorybank (1 thru 8)
Dat4 don't care

Vertical period = ((dat1 * 256) + dat2) * 16 usec.

Remarks :

1. The selected block is not closed.

When the projector is working with a not closed block (copied or interpolated from other block(s)), also indicated on the controller unit inside the projector with the 'Block Match' LED, the most significant bit of the returned Source Number byte is set to 1.

e.g. When input 4 is selected, `Dat2` will be :

block matched : 4

no block matched : 132 (=128 + 4)

2. No coincidence.

No coincidence for a selected source happens when the sync pulses are missing, the source is not switched on, ...

Without coincidence `READ_SOURCE` will return :

`Dat1` = 0, indicating the no coincidence state.

`Dat2` = 0, indicating the no coincidence state

`Dat3` = requested source number (see also point 1)

`Dat4` = the source type and installation related to the requested source number.

READ_STATUS 4BH

This command asks the projector to return the status in which it is working.

Information sent : Dat1=0, Dat2=0, Dat3=0, Dat4=0

Returned information is :

Dat1 : Bit 0..5 : menu nr. (internal ref. Nr. of displayed menu)
bit 6 : Green convergence 1 = Installed
bit 7 : Fast / Slow1 = Fast

Dat2 : cursor position

Dat3 : bit 0 : Standby1 = standby
bit 1 : Pause1 = pause
bit 2 : Text on/off1 = text on
bit 3 : PC mode1 = PC mode
bit 4 : Sound available
bit 5 : see below
bit 6 : see below
bit 7 : if set, following additional info is available :

Dat3 : bit 6 : focus is magnetic
bit 5 : convergence is stripped

Dat4 : bit 0 : orbiting is installed
bit 1 : soft edge is installed
bit 2 : contrast modulation is installed
bit 3 : NS is mounted on the convergence
bit 4 : controller with ASIC
bit 5 : IRIS is installed
bit 6 : dynamic stigmators
bit 7 : reserved

from V 2.20 :

Returned information is :

Dat1 : Bit 0..5 : menu nr. (internal ref. Nr. of displayed menu)
bit 6 : Green convergence 1 = Installed
bit 7 : Fast / Slow1 = Fast

Dat2 : type of projector
0 = BR912
1 = Baron 908
2 = Prodas 919
3 = BR908
.
FF = BR909

Dat3 : bit 0 : Standby1 = standby
bit 1 : Pause1 = pause
bit 2 : Text on/off1 = text on
bit 3 : PC mode1 = PC mode
bit 4 : Sound available
bit 5 : new contrast modulation 1 = true
bit 6 : LIMO is installed
bit 7 : if set, following additional info is available :

Dat4 : bit 0 : orbiting is installed
bit 1 : soft edge is installed
bit 2 : contrast modulation is installed
bit 3 : Powerupmode 1 = operating
bit 4 : Memory-banks 1 = on
bit 5 : IRIS is installed
bit 6 : Common Address 0 or 1
bit 7 : Warm Up 1 = on

Information sent : Dat1=1, Dat2=0, Dat3=0, Dat4=0

Returned information is :

Extra info from LIMO

Information sent : Dat1=2, Dat2=0, Dat3=0, Dat4=0

Returned information is :

Extra info from the LED Wall

Information sent : Dat1=3, Dat2=0, Dat3=0, Dat4=0

Returned information is :

extra info from the BARON

Dat1 : height of the table

Dat2 : angle of the table

Dat3 : maximum height

Dat4 : zero position (horizontal)

READ_VERSION 4CH

This command asks the projector to return the soft version and the type of projector :

Information sent : Dat1=0, Dat2=0, Dat3=0, Dat4=0

The returned information :

Dat1 : alpha type identifier G = Graphics

D = Data

V = Vision

Dat2 : BCD encoded soft version : two digits preceding the decimal point.

Dat3 : BCD encoded soft version : two digits following the decimal point.

example : V01.10
Dat2 = 01
Dat3 = 10

Dat4 : information in Dat4 is only valid if in READ_STATUS

Dat3 returned bit7=1.

Dat4 Projector type

- 0 800
- 1 801
- 2 1200
- 3 1100
- 4 1600
- 5 700
- 6 1208
- 7 1101
- 8 500
- 9 808
- 10 1209
- 11 701
- 12 RCVDS
- 13 801 with 68000
- 14 1208/2
- 15 V701 with Data input
- 16 801 with fixed color temp : V5.20 and later
- 17 1609 (with 68000)
- 18 1109 (with 68000)
- 19 V701 with 32 blocks : V3.00 and later
- 20 1208S (dyn stigm block flyback dependent, has left corr since V7.20)
- 21 1209/2 (dyn stigm block flyback dependent, has seagull and left corr since 7.20)
- 22 1109 (with left corr; since V7.10)
- 23 808 (with left corr + dyn stigm block flyback dependent; since V6.30)
- 24 808s
- 25 701 with 68000 (=708)
- 26 1208s/2

27 1209/3
 28 1109/3
 29 1609/3
 30 R808s
 31 V508
 32 VSE 10
 33 VSE 20
 34 VSE 40
 35 BR812
 36 LWD10 LED Wall Digitizer
 37 CINE6
 38 BR909, CINEMAX(48 blocks of 1024 settings)
 39 BR912
 40 VSE 60
 41 Phoenix
 42 BR908

READ_SERNR 4DH

The projector's serial number is returned in BCD format (Di):
 Dat1 = D7D6
 Dat2 = D5D4
 Dat3 = D3D2
 Dat4 = D1D0
 The serial number = D6D5D4D3D2D1D0
 D7 = FH, otherwise the serial number is not programmed yet.

CONTROL_TEXT 4EH

This command switches off the text which was first programmed using PLACE_TEXT

With Dat1 = 00 the text which was previously placed using PLACE_TEXT is cleared.

LOCK_IR 50H

Dat2, Dat3, Dat4 = 00
 Dat1 = MASK :
 value allowed IR-command
 00H [None]
 01H Stand by
 02H Pause
 04H Text
 08H Adjust keys (Adj, Enter, Exit, cursors)
 10H Numeric keys
 20H Picture control keys
 40H Sound control keys
 80H Funktion keys

E.g. : To enable all IR commands : Dat1 = FFH
 To enable all except stand by : Dat1 = FEH

FORCE_TOGG 51H

Dat2 = 00H

Dat1		Dat3	Dat4	
		1/0		
00H	none	x	-	-
01H	pause	x	pause	operate
02H	mute	x	mute	mute off
03H	mono/stereo	x	mono	stereo
04H	normal/expanded	x	normal	expanded
05H	sound/pict	x	sound	pict
06H	text	x	on	off
07H	sync	x	fast	slow
08H	enhanced blue	x	on	off
09H	adjust	x	adj	operational
0AH	power	x	on	stand by
0BH	reserved for RCVDS	x	beeper on	beeper off
0CH	reserved	x	8 bl	1 bl
0DH	ABL(VIP only)	x	on	off
0EH	screen saver	x	on	off
0FH	warm up	x	on	off
10H	internal use			
11H	CRT drive mode	0	boost	normal
	CRT drive mode	1	perm. boost	eco
12H	ACTAS data valid	x	valid	invalid
13H	orbit		see below	
14H	clamp mode	x	normal	restoration
15H	bank switching	Source nr	memory bank	
16H	filmmode	x	on	off
17H	Time base correction	x	on	off
18H	Scan converter mode	x	0 thru 3(LIMO PRO thru 6)	
19H	LIMO	x	on	off
1AH	Macrovision	x	on	off
1BH	freeze	x	on	off
1CH	TestBench (BR909)	x	on	off
1DH	Splitscreen	x	on	off
1EH	Option1	x	on	off (only for R909)
20H..	reserved for the LED-Wall			
..3FH	reserved for the LED-Wall			

*Forced orbiting (13H)	Dat3	Dat4	
	0	0	orbit off; store in memory
	0	1	orbit on; store in memory
	0	2	orbit off; don't store
	0	3	orbit on; don't store
	1	0	orbit off; speed is slow
	1	1	orbit on; speed is slow
	1	2	orbit off; speed is fast
	1	3	orbit on; speed is fast
	1	4	slave mode; speed is slow
	1	5	slave mode; speed is fast

READ_TIME 60H

The projector run time is returned as an unsigned integer with the high order byte in Dat1 the low order byte in Dat2.

run time = Dat1 x 256 + Dat2

from V 2.20
Information sent :

```
Dat1 = 0 Global runtime
Dat1 = 1 Red tube runtime
Dat1 = 2 Green tube runtime
Dat1 = 3 Blue tube runtime
```

SET_INT_PATT 61H

(+)

This command allows the user to set the internal pattern of the projector to a requested set of horizontal and vertical frequencies. The displayed picture is the standard convergence raster. In the future, it is possible that `Dat4` is used to hold a number of different patterns.

```
Dat1 Hor_Period_M
Dat2 Number_of_lines_M
Dat3 16 * Hor_Period_L + Number_of_lines_L

Dat4 function
0    blank, nothing on the screen
1    convergence grid
...  reserved
```

$\text{Number_of_Lines} = 16 * \text{Number_of_lines_M} + \text{Number_of_lines_L}$

$\text{Hor_Period} = (16 * \text{Hor_Period_M} + \text{Hor_Period_L}) * \text{Hor_base_period}$

Where `Hor_base_period` depends on the type of projector and the type of controller in the projector, for the BR909 the period is 31.25 nsec.

Command can also be used to set BARON table

```
Dat1    height
Dat2    angle
Dat3    don't care
Dat4    02
```

CONTROL_IRIS 62H

Note : after a task was started, READ_STATUS can be used to monitor its progress; see column "response..."

IRIS3				
Cmd	Dat1	Dat2	Dat3	Dat4
62H	function	00H	00H	00H

Dat1	function	response to READ STATUS 4BH
		(in Dat1, bits 7 and 6 stripped off)
00H	start IRIS	00H 00H→1CH (startup done)
01H	focus the lens	1EH
02H	not used	
03H	align current source	1EH
04H	align all sources	1EH
05H	touch up (shift adjustment)	1EH
06H	not used	
07H	Quick geometry	1EH After geometry is done: give command ENTER 07H to converge -or- give command EXIT 08H to quit
08H	touch up all sources	1EH
09H	align camera horizontally,	1EH give command ENTER 07h align camera vertically, 1CH give command ENTER 07h
0AH	IRIS2 diagnostics (no answer)-none-	when finished, give command to exit operation (see next line)
0CH	full geometry	1DH
0DH	Learn reference	1DH
0EH	Toggle front/rear	dat2 = 0 => front dat2 = 1 => rear
FFH	exit operation, go back 1 menu	1EH→1CH 1CH→00H

CONTROL_RCVDS 62H

See RCVDS documentation.
(code already exists as CONTROL_IRIS, but will not interfere)

CONTROL_ACTAS 62H

(code already exists as CONTROL_IRIS and CONTROL_RCVD, but will not interfere)

Purpose : specific programming of WGEN functions.

Cmd	Dat1	Dat2	Dat3	Dat4
62H	Funct	00H	00H	On/Off

Dat1	Dat4	function
00H		reserved
01H	1/0	projector and actas interface answers come through / are muted towards host

Purpose : specific programming of ACTAS functions.

Cmd	Dat1	Dat2	Dat3	Dat4
62H	function	00H	data	00H

Dat1	Dat3	function
00H	0..3	address block offset for addr. table of configs nr. 0..3 (obsolete)
10H	0..19	select setup nr. 0..19
11H	0..19	select config nr. 0..19
20H	0	start adjust using current setup (as a manual "start adj")
20H	1	start adjust using current setup, and report when finished
20H	2	start adjust using current setup, don't stop at prompts, and report when finished

Remark : functions invoked with 20H can be aborted by sending a record containing EXIT (08H).

Response : for the function invoked with 20H option 2, ACTAS will take the initiative to report completion with 8 full format records (i.e. starting with STX OFFS) containing 0xFE as address, and following data :

Cmd	Dat1	Dat2	Dat3	Dat4
62H	20H	index	status	00H

index = report index 0..7; 0..3 being for the 1st adjustment frequency, 4..7 for the 2nd.

Dat2	Dat3	status
index	0	"OK"
index	1	"time out"
index	2	"out of range"
index	3	"aborted by user"
index	4	"skipped"

CONTROL_AUTOIMAGE 63H

Internal use only

CONTROL_RTC 64H (from V 2.20)

Control the Real Time Clock

All data is BCD coded.

Information sent :

Dat1 : selection

0 Read time

Returns :

Dat1 = 0

Dat2 = hour

Dat3 = minutes

Dat4 = seconds

1 Read date

Returns :

Dat1 = 1

Dat2 = day

Dat3 = month

Dat4 = year

2 Set time

Dat2 = hour

Dat3 = minutes

Dat4 = seconds

3 Set date

Dat2 = day

Dat3 = month

Dat4 = year

CHANGE_ADDR 6EH

Information sent :

Dat1 : selection

1 change address

2 change baudrate

3 change powerup mode

Dat2 : 0

Dat3 : databyte

Dat4 : 0

If Dat1 = 1 then Dat3 is the new projector address (0...255)

If Dat1 = 2 then :

Dat3 = 4 => baudrate = 1200

Dat3 = 5 => baudrate = 2400

Dat3 = 6 => baudrate = 4800

Dat3 = 7 => baudrate = 9600

Dat3 = 8 => baudrate = 19200

If Dat1 = 3 then :

Dat3 = 0 => powerup mode is standby

Dat3 = 1 => powerup mode is operating

BLK_PC_E2 71H

This command invokes a block transfer from computer to projector:

Such a block consists of the various adjustments of the projector. Appendix B gives a detailed list of which byte is meaning which adjustment. Before the data for the block's adjustment parameters can be transmitted, the block's header (see READ_SOURCE) is first dispatched to the projector.

If there is already a block available in the EEPROM in the projector, then the data in the EEPROM is replaced by the transmitted data. If there is no such a block available, then a new one is created. So, transmitting blocks has to be done with a lot of care. Otherwise the contents of the EEPROM may become unusable. If the EEPROM is already completely filled up, then a message will appear on the screen of the projector. This can be avoided by first checking the contents of the EEPROM before transmitting a block with a not yet existing heading.

BLK_E2_PC 72H

Using this command the projector transmits a block to the PC.

Dat1 to Dat4 have to hold the necessary information as described in READ_SOURCE. Then the projector will return the block in his EEPROM with the corresponding header. If a not existing block is requested, no error information is returned. So take care.
As a memory block consists of 1024 settings the transmission has to be done in 6 steps.

First send the first part of the header :

Dat1	Dat2	Dat3	Dat4
HP Hi	HP Lo	Src Nr	Src type(see Read Source)

Then the second part with Dat1 = FCh and Dat4 = 0. Dat4 is the offset in the table. So this command has to be send five times, offset 0 to 3 will return 240 valid bytes, offset 4 will return the last 64 bytes of the memory block followed by 144 softedge settings.

Dat1	Dat2	Dat3	Dat4
FCh	VP Hi	VP Lo	offset 0 thru 4

From V 2.20

Low nibble is offset 0 thru 5
High nibble is memorybank (1 thru 8)

Offset 5 will return all the Softedge settings in one block of 240 bytes.

A special case is used to get an overview of all blocks of the projector : (except "common" and "stigmator" blocks)

Dat1	Dat2	Dat3	Dat4	function :
FFH	FFH	FFH	00H	overview of blocks (first 240 bytes of header file)
FFH	FFH	FFH	01H	overview of blocks (last 48 bytes of header file)

From V 2.20 add 48 memorybanks after header file

FEH	FFH	FFH	00H	stigmator block 4 pole red (98 settings)
FEH	FFH	FFH	01H	stigmator block 4 pole green (98 settings)
FEH	FFH	FFH	02H	stigmator block 4 pole blue (98 settings)
FEH	FFH	FFH	03H	stigmator block 6 pole red (98 settings)
FEH	FFH	FFH	04H	stigmator block 6 pole green (98 settings)
FEH	FFH	FFH	05H	stigmator block 6 pole blue (98 settings)

FDH	FFH	FFH	FFH	used to set IRIS zones
-----	-----	-----	-----	------------------------

FCH	FFH	FFH	FFH	read diagnostics block
-----	-----	-----	-----	------------------------

From V 2.20

Current status followed by 5 logged errors

EFH	FFH	FFH	FFH	common block
F0H	00H	FFH	FFH	ACTAS serial number table transfer
F0H	01H	FFH	FFH	ACTAS serial number table comment transfer

Then the configuration of the EEPROM is returned to the computer in the way described in appendix B.

BLK_DEL 73H

This function deletes a block in the EEPROM :

First send the first part of the header :

Dat1	Dat2	Dat3	Dat4
HP Hi	HP Lo	Src Nr	Src type(see Read Source)

Then the second part with Dat1 = FCh and Dat4 = 0

Dat1	Dat2	Dat3	Dat4
FCh	VP Hi	VP Lo	0

From V 2.20

Low nibble is 0

High nibble is memorybank (1 thru 8)

FORCE_BLK 74H

This command puts the projector in "forced block mode".

(not available in the 700 series)

This commands allows the user to force the projector to work with a specific block in the EEPROM. Normally, everything is managed automatically by the software in the projector and the block is selected by looking at the selected source number, the horizontal and the vertical frequency and the installation (front, rear, table, ceiling). Using this command, the user can limit this automatic selection and force the projector to a block or to a subset of blocks. There are two data bytes relevant when using this command:

Dat1 : Source Number (from 0 to 100)

Dat2 : Source Type : 80H = Video

90H = SVHS

A0H = 2nd RGB

B0H = RGB

C0H = Component Video

Dat3 and Dat4 : meaningless

When not in the forced block mode, Source number in the header of a block in the EEPROM is related to the input selection :

Without RCVDS	With RCVDS
1 : Video	1 : Slot 1
2 : SVHS	2 : Slot 2
3 : 2 nd RGB	3 : Slot 3
4 : RGB	...
5 : Component Video	

In forced block mode, this relationship disappears. A block in the EEPROM intended to be used in forced block mode only holds information concerning the input selection of the projector with "source type". This "source type" also holds the information if the block is "normal" or "forced" (0H = Video normal, 8H = Video forced,...).

After putting the projector in forced block mode using the force_blk command, the projector will switch to the desired input selection (according to Dat2) and search a block with matching "source number" (according to Dat1), frequencies and installation. In this way the projector will work with a "subset" of blocks in the EEPROM (It is possible that there is only one block in this subset). These blocks in this subset all have the same "source number" and "source type" in their header. Which block is selected in the subset depends on the deflection frequencies and installation of the connected source.

Using an RCVDS, the relation of the source number with the slot number disappears in the forced block mode. Therefore the user first has to select the required input slot (using NUM_x) before using force_blk with the correct type specification.

Let's look at an example. Suppose the EEPROM is configured in this way :
 For source type Video, SVHS and 2nd RGB, everything is normal. However the EEPROM is organised so that RGB also has blocks to work in forced block mode. When selecting "4", the fifth block in the EEPROM will be used (suppose the deflection frequencies and installation are correct). When the command FORCE_BLK is used with Source number = 10 and Source type = B (forced RGB) the projector will select the sixth or seventh block, according to the deflection frequencies. With Source Number = 11, the eight block will be selected.
 The projector returns to the normal mode when selecting another source using the RCU or the NUM_x command using the PC communication.

How can we create a block in the EEPROM intended for forced block usage? This is done by using the force_blk command with the desired source number and appropriate source type, connect the source to the projector and adjust the source. In this way the block is automatically created.

This forced block mode can be useful when using non-BARCO switchers. Suppose this configuration is used :

And the EEPROM looks like :

The computer in this installation drives the switcher and the projector. After selecting a new input port in the switcher, the computer uses FORCE_BLK to select one of the special blocks in the EEPROM. When selecting source number 11 or 13, there are two blocks available for that source, for two different deflection frequencies.

REQ_SHAPE	78H	
	Dat1	
	0H	internal convergence pattern
	1H	horizontal line (full width) in center of zones with coordinate Y
	2H	vertical line (full height) in center of zones with coordinate X
	3H	reserved
	4H	reserved
	5H	horizontal and vertical line together, cross-ing in zone XY
	Available in operational mode only :	
	6H	place convergence contour around zone XY
	7H	erase shape; switch color as in Dat3
	8H	vert. bars; switch color as in Dat3; width in pixels in Dat4
	9H	hor. bars, switch color as in Dat3; height in lines in Dat4
	AH	reserved
	Dat2	BCD encoded coordinates
		bits7-4 : Y coordinate
		bits3-0 : X coordinate
	Dat3	color
		01H : R on
		02H : G on
		04H : B on
		+ combinations

WRITE_POT 79H

Write_pot and read_pot gives direct access to the settings in the digital potentiometers and the EEPROM. With write_pot the data in the potentiometer is changed.

Dat1 : kind of setting

Dat2 : coordinates of setting

Dat3 : delta/value

Dat4 :

Bit 0,1

0 Dat3=delta (with storage in E²PROM)

1 Dat3=immediate value (with storage in E²PROM)

2 Dat3=delta (no storage in E²PROM)

3 Dat3=immediate value (no storage in E²PROM)

Bit 2

0 = Coarse convergence

1 = Fine convergence

If the value in the addressed potentiometer was V before the command, It is V+delta after the command. The corresponding contents of the EEPROM are also changed. See appendix for coding of the potentiometers

READ_POT 7AH

Read_pot reads the contents of the addressed EEPROM. Dat1 and Dat2 are the same as with write_pot. The content of the potentiometer is returned in Dat3.

BLK_PC_FLASH 7BH

Using this command the PC transmits a block to the projector.

Info :	Dat1	Dat2	Dat3	Dat4
	Offset Lo	offset Hi	0	nr of valid bytes

The offset in the Flash memory is calculated as follows :

Offset = (Dat2 * 256) + Dat1

Reserved addresses :

0	"Always listen" (708)
249	Stand-alone LiDo
250	QUAD Switcher 350
251	VISU
252	ACTAS External Generator (Host)
253	ACTAS Interface
254	
254	ACTAS Control Unit
255	SN-addressing protocol (VIP only)

APPENDIX : ENCODING FOR DIRECT POTENTIOMETER ACCESS

Coordinates (transmitted in Dat2) are required for potentiometers in the convergence, focus stigmators, linearity controls :

	X = 0	1	2	3	4	*
Y = 0							
1							
2							
3							
4							
*							

* = depending of the number of zones

Y and X are encoded as two nibbles in Dat2 : Y : X.

- 00H Convergence Green Vertikal
- 01H Convergence Green Hor.
- 02H Convergence Red Vertical
- 03H Convergence Red Horizontal
- 04H Convergence Blue Vertical
- 05H Convergence Blue Horizontal
- 06H Blanking Left
- 07H Blanking Right
- 08H Shift Red Horizontal
- 09H Shift Green Horizontal
- 0AH Shift Blue Horizontal
- 0BH NU
- 0CH Blanking Bottom
- 0DH Shift Red Vertical
- 0EH Shift Green Vertical
- 0FH Shift Blue Vertical
- 10H Amplitude Vertical
- 11H Linearity Vertical
- 12H Phase horizontal
- 13H Blanking Top
- 14H Side Keystone
- 15H Vertical Centerline Skew
- 16H Side Bow
- 17H Vertical Centerline Bow
- 18H NU
- 19H NU
- 1AH NU
- 1BH NU
- 1CH Top Bow
- 1DH Bottom Bow
- 1EH Top Keystone
- 1FH Bottom Keystone

- 20H Fine Shift Green Vertikal
- 21H Fine Shift Green Horizontal
- 22H Fine Size Horizontal

convergence has 9 by 9 nine zones

23H NU
 24H NU
 25H NU
 26H Horizontal Centerline Bow Green
 27H Horizontal Amplitude
 28H Horizontal Centerline Skew
 29H Left Keystone
 2AH Right Keystone
 2BH Left Bow
 2CH Right Bow
 2DH Coarse Shift Red
 2EH Coarse Shift Blue
 2FH Vertical S correction
 30H Saturation
 31H Hue
 32H Sharpness
 33H Contrast
 34H Brightness
 35H Volume
 36H Bass
 37H Treble
 38H Balance
 39H Red Gain
 3AH Blue Gain
 3BH Red Cut Off
 3CH Blue Cut Off
 3DH Green Gain
 3EH Green Cut Off
 3FH NU
 40H NU
 41H Blue correction Breaks
 42H Blue correction Slope
 43H Peaking
 44H Midlights Red
 45H Midlights Blue
 46H – 4BH NU
 4CH Horizontal Linearity
 4DH – 4FH NU
 50H Focus Midpoint Red
 51H Focus Midpoint Green
 52H Focus Midpoint Blue
 53H NU
 54H Focus Bottom Red
 55H Focus Top Red
 56H Digital Focus Red (use coordinates, 5 by 5 zones)
 57H NU
 58H Focus Bottom Green
 59H Focus Top Green
 5AH Digital Focus Green (use coordinates, 5 by 5 zones)
 5BH NU
 5CH Focus Bottom Blue
 5DH Focus Top Blue
 5EH Digital Focus Blue (use coordinates, 5 by 5 zones)
 5FH NU

 60H – 6FH not implemented yet

 70H Dynamic Astigmatism Red axial 4 pole (use coordinates, 5 by 5 zones)
 71H Dynamic Astigmatism Red diagonal 4 pole (use coordinates)
 72H Dynamic Astigmatism Green axial 4 pole (use coordinates)
 73H Dynamic Astigmatism Green diagonal 4 pole (use coordinates)

74H Dynamic Astigmatism Blue axial 4 pole	(use coordinates)
75H Dynamic Astigmatism Blue diagonal 4 pole	(use coordinates)
76H Dynamic Astigmatism Red axial 6 pole	(use coordinates)
77H Dynamic Astigmatism Red diagonal 6 pole	(use coordinates)
78H Dynamic Astigmatism Green axial 6 pole	(use coordinates)
79H Dynamic Astigmatism Green diagonal 6 pole	(use coordinates)
7AH Dynamic Astigmatism Blue axial 6 pole	(use coordinates)
7BH Dynamic Astigmatism Blue diagonal 6 pole	(use coordinates)

80H Left Start
 81H Left length
 82H Right start
 83H Right Length
 84H Gamma Red Low
 85H Gamma Red High
 86H Gamma Green Low
 87H Gamma Green high
 88H Gamma Blue low
 89H Gamma Blue High

B0H Fine Side Precorrection (E/W)	(use Y coordinate, 9 zones)	Always 0 thru 8
B1H Fine Vertical Linearity	(use Y coordinate, 9 zones)	
B2H Fine Horizontal Linearity	(use X coordinate, 9 zones)	