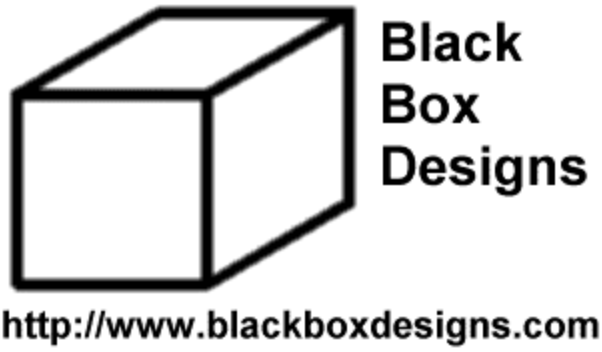

Serial Audio Video Remote (SAVR / SAVR2) Interface Description

Revision C – March 18, 2005

Authors:

Jeff Behlendorf and Brian Behlendorf

Revision History

Rev	Date	Pages	Change Description
Draft 1	Mar 9, '03	All	Initial Version
Draft 2	April 17, '03	2 3	Inserted revision history. Error code 0x11 was 0x04 Error code 0x21 was 0x05 Error code 0x22 was 0x06 Inserted error code 0x04 and 0x05 Added error codes 0x23 and 0x30 The response to command 0x00 has changed, was "Firmware Revision level"
Draft 3	June 4, '03	All 3 4 5 6	Revised all instances of NNA to SAVR Revised all instances of 'No Name Adapter' to 'Serial Audio Video Remote' Revised message length limit in table 1 to 31 bytes, was 16 Removed footnote 2 from table 1, was Control-S length limit of 3 Updated command/response code table, removed Control-S / Control-A1 references from all channels Re-identified Control-A1 channels A-C as channels D-F Added channel G to command response table Added error code 0x12 Error code 0x23 description was *Disabled Function* Revised firmware revision response example to use SAVR v0.0.1 Completely revised command message 0x02 syntax Completely revised command message 0x03 syntax
IR	Sept 20, '03	1 2 4 6	Added Black Box Designs logo and headers Section heading was "NNA Device Identifiers" Added command/message code 0x04 Added maximum receive port limitation to 0x03 command Added description of command/message 0x04
A	Jan 4, '04	4-5 5 6 7 10	Added section "SAVR Hardware Characteristics and Interface" Added discussion of message buffers below table 1 Added discussion of messages acknowledgements below table 2. Relocated error codes to sheet 11 under 0xFE message Added to command 0x03, "In general, Control-A1 …" Added SAVR response to command 0x03

A (cont.)		10	Revised options for receive configuration under command 0x03, was “Any two ports may be configured to receive…”
		11	Added command code 0x05 detail description, Query Serial Number
			Corrected 0xFE message example, was incorrectly formatted command
			Added error codes 0x06, 0x13, and 0x14
			Deleted error code 0x11, was *RS-232 overflow*, Reserved to prevent conflicts of reuse and preserve consistency in future code
		12	Added discussion of multiple error behavior in malformed messages
			Added discussion of reliability and high data rates
		All	Revised all references from “Channels” to “Ports” for clarity.
B	Sept 19, ‘04	5	Corrected maximum payload, was 31 bytes, is 30 bytes.
			Corrected payload range ref., was 0x00-0xFF, is 0x00-0xF0
			Buffer was described as large enough for 34 bytes, revised to 33.
			Added “Commands with lengths exceeding the legal limit…”
		6	Deleted “The current SAVR programming does NOT echo commands, but the option has been preserved in the protocol…“ The SAVR does echo commands by default.
		9	Corrected command 0x02 example to be legal and consistent.
		10	Corrected description of command 0x03, response to 0x03 commands was incorrectly noted as 0x02.
		11	Added commands 0x06 Query Echo and 0x07 Set Echo
C	Mar 18, ‘05	All	Changed all general references to SAVR to SAVR / SAVR2
		4	Revised “version 1 device” to “current versions of the device”
			Corrected typo: “in” was “of”
			Revised “in the version 1 device” to “in the SAVR / SAVR2”
			Corrected typo: “witch” was “with”
		4, 5	Corrected “buss” to “bus” in multiple locations
		5	Added description of “barber pole” behavior
		6	Corrected typo: “sent” was “set”
		7	Added 0x06 and 0x07 to table 3
		8	Revised “applies both to all” to “applies to all”
			Added description of SAVR vs. SAVR2 firmware version responses
		9	Revised “version 1 SAVR” to “SAVR / SAVR2”
		10	Added description of SAVR vs. SAVR2 receive port capabilities

SAVR / SAVR2 Hardware Characteristics and Interface

The SAVR (Serial Audio Video Remote) and SAVR2, referred to generically as SAVR / SAVR2 throughout this specification, provide an interface to translate between an RS-232 capable device, such as a personal computer, and the Control-A1, Control-A1 II, or SIRCS protocols generally used by Sony consumer electronics. Only these standards are implemented in the current versions of the device.

Power is provided via a coaxial DC power jack (5.5 mm OD, 2.1 mm ID, center +). The recommended input voltage is 9V DC, but the device will function with power supplies anywhere in the 6-9V DC range. The power supply must be capable of delivering at least 150 mA, although the SAVR / SAVR2 typically consumes only 70-80 mA.

The RS-232 interface is made using a standard DB9 serial cable. The serial cable length should be limited to 15 feet (4.5 meters) maximum. This is less than the more typical 25 foot recommended limit for PC serial cables. Longer runs may be possible, but the cable length should be minimized to ensure maximum reliability.

The SAVR / SAVR2 serial communication settings are:

Baud Rate : 57600
Parity : None
Data Bits : 8
Stop Bits : 1

No other baud rate settings are supported. No flow control is implemented. The SAVR / SAVR2 will generate an error when it's message buffer is full, and it is left to the controlling device to manage any retransmissions in complete messages.

There are 3 message buffers in the SAVR / SAVR2. One buffer is reserved for each implemented port (A-C). These buffers are used both for incoming and outgoing message traffic and are each large enough to hold a single maximum length message.

The control interface jacks are standard 1/8" monaural headphone style connections. Use of 1/8" stereo connections is acceptable, but only the tip and shield ring are used. When configured for Control-A1 / Control-A1 II communication, the SAVR / SAVR2 control jacks provide a weak 4.7V pull-up in the steady state condition. Signaling is accomplished by pulling the bus low to a high impedance ground. When configured for SIRCS input, the control jacks are high impedance inputs. When configured for SIRCS output, the control jacks generate 4.7V pulses consistent with cable remote standards.

NEVER APPLY A SIGNAL GREATER THAN 5V DC TO A CONTROL CONNECTION ON THE SAVR!
Damage to the input circuit or microcontroller is possible and likely when excessive voltages are applied.

The precise cable length limits for reliable communication via the control jacks is dependant on the cabling selected and environment through witch the cabling is run. The signaling rate is relatively low, which minimizes the effects of cable capacitance and signal slew. The limiting factors in cable length are therefore voltage drop due to wire resistance and environmental noise. The SAVR / SAVR2 control jacks have been tested with cable lengths of 35 feet with a very high reliability. As always, cable length should be kept to a minimum.

When using the Control-A1 / Control-A1 II protocol, bus arbitration is handled by the SAVR / SAVR2 using a wait-for-silence methodology. The wait interval selected is 3 ms, which is the length of the longest characteristic pulse used during normal communication on a Control-A1 / Control-A1 II bus.

When connecting the SAVR / SAVR2, it is important to apply power to the device after all other connections are made. Plugging or unplugging data lines will often cause a shift in the reference ground plane between interconnected components. This shift can cause lockup of the onboard microcontroller or prevent correct transmission on the control jacks. To ensure reliable operation, after connecting or disconnecting a device, remove power from the SAVR / SAVR2 for 10 seconds and then re-power the unit. Issuing a software reset of the microcontroller may not resolve issues created during a change of connections.

The Rx (Receive) and Tx (Transmit) indicator lights on the face of the SAVR / SAVR2 serve a dual purpose. When power is applied both lights are activated by the onboard microcontroller after it has initialized and the SAVR / SAVR2 is ready to transmit data. On SAVR units with firmware 0.2.7 or later and all SAVR2 adapters, the Rx / Tx lights will barber-pole (flash alternately) for 2 seconds during initialization of the device (after a reset). During this period the SAVR / SAVR2 will not respond to commands. When data is being received or transmitted on any of the control jacks, the lights will blink indicating traffic. The lights will not flash during communication over the RS-232 serial port. RS-232 traffic is easily monitored in other ways, and it would diminish the usefulness of the control jack monitoring.

SAVR / SAVR2 General Message Format

Interaction with the SAVR / SAVR2 requires the exchange of multi-byte serial messages. Those messages must be addressed to a specific output port or to the SAVR / SAVR2 itself and must include the length of the message to be sent in bits followed by the data padded to the nearest byte. The general form of this command/response message is:

Length	1 byte	1 byte	1 byte	1-30 bytes
Valid Values	0xFF	0x00 - 0x0F	0x00 - 0xFF	0x00 – 0xF0 with Escapes (0xFF)
Description	Start of message, Escape character	Device identifier	Message length in bits	Message to be sent to device in correct bit order

Table 1: Message Format

As discussed earlier, the SAVR / SAVR2 has a single message buffer for each control port. This buffer is large enough to accommodate a 33 byte message. Any additional 0xFF escape codes are dynamically stripped or added by the onboard microcontroller when processing a message and do not count towards the message length limit. Commands with lengths exceeding the legal limit will generate error 0x04 – *Invalid Length*.

SAVR / SAVR2 Device Identifiers

Unique identifiers are used for commands and responses from the same device to simplify resolving if a message is normal incoming data or an echo of a command. The following are the device identifiers recognized by the SAVR / SAVR2:

Command Code (Hex)	Response Code (Hex)	Device/Port
0x00	0x08	SAVR / SAVR2 Command Port
0x01	0x09	Port A
0x02	0x0A	Port B
0x03	0x0B	Port C
0x04[1]	0x0C[1]	Port D[1]
0x05[1]	0x0D[1]	Port E[1]
0x06[1]	0x0E[1]	Port F[1]
0x07[1]	0x0F[1]	Port G[1]

Table 2: Device Identifiers

[1] Ports D thru G are not available on the SAVR / SAVR2 due to inadequate memory space and processing ability, but the addressing has been reserved in the command language for a potential future versions of the device with more memory available.

The SAVR / SAVR2 acknowledges commands sent out over the control ports by echoing the original message back after successful transmission. This does not include direct commands to the SAVR / SAVR2 (device 0x00) which generate responses as described later in this document. Any difficulty in posting the message will generate an error code.

Therefore, a typical command and acknowledgement response for a message on port A could be:

Command : 0xFF 0x01 0x10 0xAA 0xBB

Acknowledgement : 0xFF **0x01** 0x10 0xAA 0xBB

Note that the acknowledgement does not use the response code (**0x09**) and can therefore be distinguished from traffic received on the port.

SAVR / SAVR2 Message Codes

Several commands will be available for configuration and management of the SAVR / SAVR2 itself using device identifier 0x00. These are currently defined as:

Command / Response (Hex)	Description
0x00	*Reset*, return the adapter to the default settings and flush all data
0x01	*Query Firmware Revision*, Get the firmware version from the SAVR / SAVR2
0x02	*Query Port Settings*, Get the current I/O configuration of the SAVR / SAVR2
0x03	*Set Port Settings*, Set the I/O configuration of the SAVR / SAVR2
0x04	*Query SAVR / SAVR2 Uptime*, Requests the time elapsed since the last reset
0x05	*Query SAVR / SAVR2 Serial Number*, Requests the SAVR / SAVR2 serial number
0x06	*Query Echo State*, Checks to see if the command echo is enabled or disabled.
0x07	*Set Echo State*, Activates or deactivates the SAVR / SAVR2 command echo feature.
0xFE	*Error* (Response Only), *indicates an error condition on the SAVR / SAVR2.*

Table 3: SAVR / SAVR2 Management Command/Response Codes

0x00 – Reset

The reset command essentially reboots the adapter. All data is lost, message buffers flushed and initial conditions restored. The correct format of a reset command is:

- 0xFF 0x00 0x08 0x00

The adapter will respond with the error message 0x30 after a reset. This applies to all reset conditions. Resets may occur due to a command sent over the serial port, a hard power reset of the adapter, or an internal watchdog timer detecting an internal error. The response will have the form:

- 0xFF 0x08 0x10 0xFE 0x30

0x01 – Query Firmware Revision

Requesting the firmware version for the adapter returns the code level burned into the adapter's processor. This request can be used to check for the interface on each com port for polling purposes since the first portion of the response will always be consistent. This command is also the best way to determine if a SAVR or SAVR2 is being used from a high level application. The response will always begin with the device name.

Request the firmware version of the adapter's software by sending:

- 0xFF 0x00 0x08 0x01

The request is handled by the SAVR / SAVR2 and it returns a 16 byte ASCII string with the firmware version:

- 0xFF 0x08 0x88 0x01 0x53 0x41 0x56 0x52 0x20 0x76 0 x30 0x2E 0x30 0x2E 0x31 0x20 0x20 0x20 0x20 0x20

In this case the response is "SAVR v0.0.1 ". The SAVR2 would respond similarly, but the string would be prefixed with a different device name "SAVR2 v0.0.1 "

0x02 – Query Port Settings

The input and output ports on the SAVR / SAVR2 are configurable to support a variety of protocols. Support is in place on the SAVR / SAVR2 for Control-A1/Control-A1 II or Control-S. This configuration is reported as a string of bit packed bytes, one for each physical port implemented on the SAVR / SAVR2. The bit packing is in the following form:

Bits	Description
7-5	Port identification (1-7 are possible, 3 are implemented in the SAVR / SAVR2)
2-4	Port type (001 = Control-S, 010 = Control-A1, 011-111 = Reserved for expansion)
1	Xmit enabled (Enabled TRUE/FALSE)
0	Recv enabled (Enabled TRUE/FALSE)

The SAVR / SAVR2 will only report the physical ports implemented.

A typical request and response might appear as follows:

- 0xFF 0x00 0x08 0x02

Assuming the three ports A-C are configured as Control-A1 In/Out, Control-S In, and Control-A1 Out respectively, the SAVR / SAVR2 might respond:

- 0xFF 0x08 0x20 0x02 0x2B 0x45 0x6A

Decoding the message we find:

0x---2---B (0x2B)	001 - Port A	010 - Control-A1	1 - Xmit enabled	1 - Recv enabled
0x---4---6 (0x45)	010 - Port B	001 - Control-S	0 - Xmit enabled	1 - Recv enabled
0x---6---5 (0x6A)	011 - Port C	010 - Control-A1	1 - Xmit enabled	0 - Recv disabled

0x03 – Set Port Settings

The message format for the 0x03 command is similar to the 0x02 response, but only one port may be configured at a time. Therefore each port must be configured with a separate message. The SAVR / SAVR2 generates a 0x03 – Set Port Settings response after a port setup command unless an error occurs.

A typical command might appear as follows:
- 0xFF 0x00 0x10 0x03 0x4B

The SAVR / SAVR2 will respond with the new setting for the port as follows:
- 0xFF 0x08 0x10 0x03 0x4B

Decoding the bit packed data gives:

0x---4---B (0x4B) 010 – Port B 010 - Control-A1 1 - Xmit enabled 1 - Recv enabled

This command therefore configured port B as a bi-directional Control-A1 data port. In general, Control-A1 ports should always be configured for both transmit and receive, but the option to selectively disable one or the other has been left in place to allow command only interaction with Control-A1 devices.

Issuing a port setup to enable a Control-S port for read and write will result in an error since it is not a valid feature of the hardware or the Control-S specification.

The SAVR can only support two receiving ports simultaneously. The SAVR2 can support receiving on all three ports simultaneously. On the SAVR only port A and B can be configured to receive. Any protocol standard may be used on any port, but on the SAVR only ports A and B will be monitored for incoming traffic. This limitation has been imposed on the original SAVR due to insufficient processing speed. Attempting to configure port C with receive enabled on the SAVR will generate an error 0x03 – Invalid Setting.

0x04 – Query SAVR / SAVR2 Uptime

This command requests the total up time on the SAVR / SAVR2 since the last reset and the returned data is coded in days, hours, minutes and seconds.

A typical command might appear as follows:
- 0xFF 0x00 0x08 0x04

The SAVR will respond with something similar to the following example:
- 0xFF 0x08 0x28 0x04 0x01 0x11 0x05 0x09

Which translates to an uptime of 1 day, 17 hours, 5 minutes and 9 seconds.

0x05 – Query SAVR / SAVR2 Serial Number

This command requests the unique 8-digit serial number stored in the SAVR / SAVR2 onboard microcontroller.

A typical command might appear as follows:
- 0xFF 0x00 0x08 0x05

The SAVR will respond with something similar to the following example:
- 0xFF 0x08 0x68 0x05 0x53 0x2f 0x4e 0x20 0x30 0x30 0x30 0x30 0x30 0x30 0x30 0x31

Which translates to "S/N 00000001"

0x06 – Query Echo

This command requests the current echo behavior of the SAVR / SAVR2. By default the SAVR / SAVR2 will echo all outbound commands to a data channel after the message has been successfully delivered. This allows blocking of message delivery in the higher level API and provides an indication that the SAVR / SAVR2 is present and functioning correctly during routine bus transactions. Query echo is not implemented on SAVR devices with firmware prior to 0.2.5.

A typical command might appear as follows:
- 0xFF 0x00 0x08 0x06

The SAVR / SAVR2 will respond with either a 1 or 0 data byte indicating echo on and echo off respectively.

- 0xFF 0x08 0x10 0x06 0x01

Echo **On**,

or

- 0xFF 0x08 0x10 0x06 0x00

Echo **Off**.

0x07 – Set Echo

This command sets the echo behavior of the SAVR / SAVR2. Once again, the power on default for the SAVR / SAVR2 is echo on. This maintains legacy behavior and is the recommended mode of operation for robust message handling. Appending the data byte with a value of 1 or 0 sets the echo behavior. Any other data byte will generate the error 0x03 - *Invalid Setting.* Set echo is not implemented on SAVR devices with firmware prior to 0.2.5.

A typical **Echo On** command might appear as follows:
- 0xFF 0x00 0x10 0x07 0x01

The SAVR / SAVR2 will respond with an acknowledgement of the echo state:
- 0xFF 0x08 0x10 0x07 0x01

0xFE – Error Message

In the event that a message is sent to the SAVR / SAVR2 that cannot be handled, an error will be generated. The error codes are identified at the start of the SAVR / SAVR2 message codes section. As an example, a bad request for the firmware revision might take the form:

- 0xFF 0x00 0x08 0x09

The command code 0x09 doesn't exist, so the adapter responds with an error:

- 0xFF 0x08 0x10 0xFE 0x02

Where 0xFE indicates an error message and 0x02 identifies the specific error as *Invalid Control*. A complete list of error codes appears below:

0x01 – *Invalid Escape*, The escape character 0xFF appears in an illegal way.
0x02 – *Invalid Control*, Invalid command code was issued to the SAVR / SAVR2.
0x03 – *Invalid Setting*, Request for an illegal port configuration has been made.
0x04 – *Invalid Length*, The message length was set to 0, which is ILLEGAL.
0x05 – *Malformed Header*, The message to the SAVR / SAVR2 was malformed.
0x06 – *Incomplete*, The message to the SAVR / SAVR2 appears incomplete. (250 ms elapsed between expected characters, this timeout may be less depending on firmware revision)
0x11 – *Reserved*
0x12 – *Truncated*, Internal buffer overflow, preceding message was truncated
0x13 – *Collision*, The SAVR / SAVR2 encountered a collision on an arbitrated bus (i.e. Control-A1/A1 II). When attempting to set the bus high, it was held low.
0x14 – *Busy*, The SAVR / SAVR2 was unable to send on an arbitrated bus for 1 second. Generally this indicates that the bus is either flooded or shorted low due to bad cabling.
0x21 – *Disabled*, The port being addressed has not been configured for this function.
0x22 – *Busy*, The port is already sending or receiving a message, wait and try again.
0x23 – *Unsupported Function*, The command is valid, but has been disabled in firmware.
0x30 – *Reset Complete*, The SAVR / SAVR2 has reset.

Multiple errors may be generated as a result of a malformed message:

Example, sending the request:
- 0xFF 0xAA 0x18 0x00 0xFF 0xFF 0x43

Produces the following three error messages
- 0xFF 0x08 0x10 0xFE 0x01
  Invalid Device (0xAA is undefined as a device code)
- 0xFF 0x08 0x10 0xFE 0x05
  Invalid Header (0x18 is not the appropriate message start flag)
- 0xFF 0x08 0x10 0xFE 0x05
  Invalid Header (0x00 is not the appropriate message start flag)
- 0xFF 0x08 0x10 0xFE 0x01
  Invalid Device (0xFF is undefined as a device code)
- 0xFF 0x08 0x10 0xFE 0x05
  Invalid Header (0x43 is not the appropriate message start flag)

Since the SAVR / SAVR2 determines the message to be flawed after the first two bytes (0xFF 0xAA) it generates an error, and then responds to each additional byte in turn as an improper way to begin a new message. Note that only the third 0xFF byte in this sequence generates an error. The SAVR / SAVR2 assumes the second 0xFF could be the start of a new valid message until it sees the following byte and determines it to be more garbage data.

A Word About Reliability and Very Heavy Data Rates

The Control-A1/A1-II and Control-S protocols are designed primarily for intermittent messaging. The SAVR / SAVR2 is capable of handling three distinct channels of intermittent message data, or one stream of continuous data, with a very high reliability. ***However***, certain devices (e.g. large multidisc CD and DVD changers), can generate a sustained data stream that will saturate the SAVR / SAVR2 ability to process messages. The SAVR / SAVR2 may still accept commands for other channels, but this will generally lead to lost data. It is left to the software developer to remain aware that requesting such large data streams will occupy nearly 100% of the SAVR / SAVR2 available processing capacity, and temporarily make communication on other channels unreliable. Since such requests are rare, and a single application must manage the SAVR / SAVR2 in most situations, this limitation shouldn't present any serious difficulties to the end user. It is mentioned here to provide advance warning of a condition that could produce significant dropped data if not addressed in the controlling application.