SVC Serial Protocol v1.0

Software Verification Communications

Doc. ID: gsa-p0011.001.01

Gaming Standards Association
Technical Committee

Standard Adopted: July 19, 2000

Document Released: August 22, 2005

SVC Serial Protocol v1.0, Document ID gsa-p0011.001.01

Released on August 22, 2005, as Final, by Gaming Standards Association (GSA).

Patents and Intellectual Property

The user's attention is called to the possibility that implementation of the Gaming Standards Association (GSA) standard or specification contained herein may require the use of inventions/ technologies covered by patents or other intellectual rights held by third parties. By publication of this GSA standard or specification the GSA makes no representation or warranty that the implementation of the standard or specification will not infringe on any third party rights. The GSA takes no position with respect to any claim that has been or may be asserted by any third party regarding intellectual property rights or [on] the validity of any such rights related to any such claims, or the extent to which a license to use any such rights may or may not be available.

Trademarks and Copyright

GSA Contact Information

GSA – Gaming Standards Association
48377 Fremont Blvd., Suite 117
Fremont, CA 94538

Phone: +1(510) 492-4063

Fax: +1(510) 492-4001

E-mail: sec@gamingstandards.com

WWW: http://www.gamingstandards.com

STANDARD / SPECIFICATION LICENSE AGREEMENT (SSLA)

IMPORTANT - READ CAREFULLY - THIS AGREEMENT DEFINES YOUR RIGHTS TO USE THE STANDARD AND/OR SPECIFICATION DESCRIBED HEREIN. USING THIS STANDARD AND/OR SPECIFICATION CONSTITUTES YOUR ACCEPTANCE OF THE TERMS OF THIS AGREEMENT AND ALL RIGHTS AND CONDITIONS THEREIN.

The standard and/or specification is owned by the GAMING STANDARDS ASSOCIATION ("Licensor"), and licensed to you as a Licensee ("Licensee").

License

Licensor grants to Licensee a non-transferable, non-exclusive license to use the standard and/or specification (hereinafter referred to as the "Standard"). Requests to reproduce, distribute or modify the Standard should be directed to the Licensor: Gaming Standards Association, 39355 California Street, Suite 307, Fremont, CA 94538 – Tel: (510) 744 4007 / Fax (510) 608 5917 / email: sec@gamingstandards.com. Further Licensee agrees that the licensed product will be treated as GSA "confidential information" and proprietary.

Limited Warranty

THE FOREGOING WARRANTY IS THE SOLE AND EXCLUSIVE WARRANTY AND IS GIVEN IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, THE STANDARD IS BEING LICENSED TO YOU "AS IS," AND LICENSOR DOES NOT WARRANT THAT THE STANDARD WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE STANDARD WILL BE ERROR-FREE.

Liability

In no event shall Licensor, its employees, agents, suppliers, or contractors be liable for any damages of any kind or character, including without limitation any compensatory, incidental, direct, indirect, special, punitive, liquidated or consequential damages, loss of use, loss of data, loss of income or profit, loss of or damage to property, claims of third parties, or other losses of any kind or character or attorneys' fees in connection with any claim relating to this Agreement or the performance of the Standard. In the event that liability is nevertheless imposed on Licensor, its employees, agents, suppliers or contractors, the liability shall not exceed the annual license fee paid for this Standard and or Specification.

Termination

This License shall automatically terminate in the event of a breach of any of the terms of this Agreement. Upon termination, you will be required to cease all use of the Standard and return to Licensor all materials related to the Standard, all copies of any kind, and any and all accompanying documentation.

General Terms

This Agreement is not assignable or transferable. The rights under this Agreement, or any License granted hereunder, may not be assigned, sublicensed or otherwise transferred by Licensee without the prior written consent of Licensor. This Agreement constitutes the entire Agreement between the parties relating to the subject matter hereof and may only be modified in writing, signed by each party. This Agreement supersedes any proposal or prior agreement(s), oral or written, and any other communications between the parties relating to the subject matter of this Agreement. The laws of the State of California shall govern this Agreement. Any questions or comments regarding this Agreement or the Standard should be directed to the Gaming Standards Association, 39355 California Street, Suite 307, Fremont, CA 94538 – Tel: (510) 744 4007 / Fax (510) 608 5917 / email: sec@gamingstandards.com.

Table of Contents

About This Document

The scope of this document is to layout a specification for a serial communication protocol. This protocol is to be used for communication between a gaming machine (EGM) and an external Master device such as a PC (generally a portable computer). The main purpose of this protocol is to standardize the information and commands needed for software authentication and verification of the software in a gaming machine.

Acknowledgements

The Gaming Standards Association would like to express its appreciation to all members of the SVC committee, past and present, for their significant contribution and dedication to the creation of this standard.

Revision History

The following table lists the changes made this document.

Revisions (Sheet 1 of 2)

Rev	Date	Change Description
1.0.2	8/22/2005	• No version change. Corrected the contact phone number on the copyright page and removed "GSA Confidential" from the page footers.
1.0.2	08/12/2004	• Added the SSLA to document after the Copyright page. • Deleted the Limited Liability paragraphs from the Copyright page as this informatin is in the SSLA. • Updated the document ID.
1.0.1	05/11/2004	• Reformatted document to current GSA format. • Changed document identifier and protocol version to be consistent with current GSA requirements. • Moved section "1.1 Revision History" to a separate page and deleted sections 1.2 through 1.5 from section "1 Foreward" as this information is now included in the copyright page. Also renamed section "1 Forward" to "About This Document" to be consistent with current GSA format. • About This Document section, first paragraph, last sentence: changed "...in gaming machine." to "...in a gaming machine." • About This Document section, Software Verification Committee Memebers sub-section: replaced with standard "Acknowledgements" section. • Revision history: Updated. • Renumbered section headings beginning with "1 Introduction", which was previously numbered "2" • Section 2, second paragraph, first sentence: changed "...undefined, however..." to "...undefined; however..." • Section 3, list item 3: changed "...atleast..." to "at least". • Section 3.1, table, under Message Data: changed "...1 ... 251 bytes" to "1 to 251 bytes..."

Revisions (Sheet 2 of 2)

Rev	Date	Change Description
1.0.1 continued		• Section 3.2.2, below table in Status Data1 description: last line, changed "...Bit 4 - 7..." to "...Bit 4 to 7..." • Section 3.2.2, below table in Data Format description: last line, changed "... '3' ... '255' ..." to "... '3' to '255' ..." • Section 3.2.5 below table in Data Format description: Changed "...format that..." to "...format of..." • Section 3.2.6, in table under Data, changed "0...248 bytes" to "0 to 248 bytes" • Appendix A: Deleted appendix (review comments appendix must be removed in final published document). • Added a "end of document" page.
01.0	05/02/2000	00-004r0 with 30 day review comment resolution changes.
00-004r0	03/05/2000	First issue of this document.

1 Introduction

The communication protocol is simple in order to reduce complexity of design, implementation, testing and usage. Due to the simplicity of this protocol, a standard layered approach is not necessary. Only the physical layer and the application layer command set are specified.

2 Physical Layer

The physical layer between the EGM and the Master is point-to-point, full duplex, no handshaking, 3 wire (Tx/Rx/Gnd) RS232C. The connector on the EGM is undefined. The EGM is a DCE that transmits on line 2 and receives on line 3 with line 5 being ground. The default communication speed is 9600 baud with eight data bits, no parity and one stop bit.

The physical connector within the EGM is undefined; however, a cable capable of being plugged into the EGM port and with a standard DB9 connector at the other end, suitable for connection to a PC serial communications port shall be provided by the manufacturer of the EGM.

NOTE

This standard does not specify whether a dedicated physical port is (or is not) required. This leaves the option open to the manufacturer as to whether port sharing is an acceptable solution within the particular jurisdiction that it will be used. It is up to the manufacturer to determine whether the jurisdiction will allow port sharing, if that is the preferred choice.

3 Application Command Layer

At the Application Layer, the Master sends a Query to the EGM and waits for the Response before sending another command. The EGM always responds to a Query with a Response. As a consequence no more than one Query / Response may be pending at the Master / EGM side at any given time.

The following timeouts will be in affect:

1. The slave must respond within 200msec of receiving a complete packet from the master.
2. The inter-byte timeout value is 5msec.
3. The master will wait at least 10msec upon receipt of a response before transmitting again.

3.1 Application Layer Frame

Command 1 byte	**Length** 1 byte	**Message Data** 1 to 251 bytes	**CRC** 2 bytes

This frame consists of the following fields:

Command [1 byte in length] This is a command byte that indicates the packet format and it's purpose.

Length [1 byte in length] The total number of bytes in frame (including Command, Length, Data and CRC bytes). Note: The maximum packet length is arbitrarily restricted to 255 bytes.

Message Data [1 - 251 bytes in length] This field contains any data relevant to the command.

CRC [2 bytes in length] CRC-16 check sum of all fields. Each frame is protected with a 16-bit Cyclic Redundant Check sequence. The CRC uses the industry standard CRC-16 polynomial generator of x16 + x15 + x2 + 1 starting with a seed of 0xFFFF(hex). See A-Link specification for further details on correct implementation of this CRC.

3.2 Commands - Query / Response Pairs

All *Queries* are from Master to Slave (Master to EGM)

All *Responses* are from Slave to Master (EGM to Master)

Each Query has one corresponding Response. The appropriate matched Response pair should be returned by the EGM when a Query is received and processed. The command byte for a Response is the same the same as that of the Query, except the high bit is set (ie. 0x02-0x82).

3.2.1 Status Query [0x01 – SQ]

Request the current status information from the EGM. [Master ⇨ EGM]

Cmd = Status Query (0x01)	**Length** = (0x04)	**CRC**
1 byte	1 byte	2 bytes

3.2.2 Status Response [0x81 – SR]

Return the current status information. [EGM ⇨ Master]

Cmd = Status Response (0x81)	**Length** = (0x08)	**Version ID**	**Status Data1**	**Data Format**	**CRC**
1 byte	1 byte	2 bytes	1 byte	1 byte	2 bytes

Version ID: 2 bytes to indicate version number. The version is a 4-digit number stored as packed BCD, where the first byte is 2-digit major revision number, second byte is 2-digit minor revision number. For example; [0x02,0x17] would indicate a version of 2.17. The initial version will be 1.00, encoded as [0x01,0x00]. Then incrementing to 1.01 [0x01,0x01]

Status Data1: General Status

Bit **0** – Calculation Status [‘0’ - Idle, ‘1’ - Calculating]

Bit **1** – Last Authentication Results [‘0’ - Not Available, ‘1’ - Available]

Bit **2** & **3** – Current Calculation:

‘**0**’ & ‘**0**’ – Requested,

‘**0**’ & ‘**1**’ – Calculating,

‘**1**’ & ‘**0**’ – Finished,

‘**1**’ & ‘**1**’ – Error, cannot complete or failed.

Bit **4** to **7** –Reserved [Always Off]

Data Format: Data formats supported

‘**0**’ – Reserved, Do not used,

‘**1**’ – Plain ASCII text,

‘**2**’ – XML format,

‘**3**’ to ‘**255**’ – Reserved for future use.

3.2.3 Last Authentication Status Query [0x02 – LASQ]

Request the Last Results status information. [Master ⇨ EGM]

Cmd = Last Results Authentication Query (0x02)	**Length** = (0x04)	**CRC**
1 byte	1 byte	2 bytes

3.2.4 Last Authentication Status Response [0x82 – LASR]

Return the Last Results status information. [EGM ⇨ Master]

Cmd = Last Results Authentication Response (0x82)	**Length** = (0x09)	**Authentication Level**	**Time Stamp (sec)**	**CRC**
1 byte	1 byte	1 byte	4 bytes	2 bytes

Authentication Level: This number indicates the level or type of authentication that was calculated. A value of 0x01 refers to Level 1 Authentication, 0x02 refers to Level 2 Authentication, and so on. A value of 0x00 is illegal.

Time Stamp: Time (in seconds) since last results were calculated.

3.2.5 Last Authentication Results Query [0x03 – LARQ]

Request the previous / currently available Authentication results. [Master ⇨ EGM]

Cmd = Last Authentication Results Query (0x03)	**Length** = (0x07)	**Data Format**	**Frame Number**	**CRC**
1 byte	1 byte	1 byte	2 bytes	2 bytes

Data Format: The format of the data

0x01 – Plain text format

0x02 – XML format

Frame Number: A 2-byte number with most significant byte first. Used to indicate the Data Frame that should be returned as data in the Last Authentication Results Response (0x83). The frame number data is indexed from '1', so a value of '0' is illegal. The range is large enough to handle a file containing up to 65535 frames.

NOTE

It is important to note that this mechanism of accessing the authentication results is linear, not random access. The rule exists in order to reduce any possible load or restrictions on the implementation within the EGM. The implications of this are that the master can only request frame *'1'* the *first* time. After that the master can only request either the *first* frame, frame *'n'*, or frame *'n+1'*, where *'n'* was the previous frame requested. This results in a linear request process, with the ability to reset back to the first frame, or request a retransmit of the current frame, or request that the next frame be transmitted.

3.2.6 Last Authentication Results Response [0x83 – LARR]

Return a data frame of the previous or currently available Authentication results. [EGM ⇨ Master]

Cmd = Last Authentication Results Response (0x83)	**Length** = (7 to 255)	**Status Data**	**Frame Number**	**Data**	**CRC**
1 byte	1 byte	1 byte	2 bytes	0 to 248 bytes	2 bytes

Status Data: General Status

Bit **0** – Error Status [‘0’ – No error, ‘1’ – Error] Note: Error would usually indicate either no data available, or and invalid frame.

Bit **1** – Frame Status [‘0’ – Not Last Frame, ‘1’ – Last Frame]

Frame Number: A 2-byte number with most significant byte first. Used to indicate the frame that is being returned in Data field.

Data: Contains requested Authentication information (formatted as requested). Refer to the following section regarding the formatting of this data, and how it is generated.

NOTE

Authentication Results are not available while an Authentication Calculation is in progress.

3.2.7 Initiate Authentication Calculation Query [0x04 – IACQ]

Request that the EGM start authentication calculation. [Master ⇨ EGM]

Cmd = Initiate Authentication Calculation Query (0x04)	**Length** = (0x05)	**Authentication Level**	**Authentication Seed**	**CRC**
1 byte	1 byte	1 byte	0 to 250 bytes	2 bytes

Authentication Level: This number indicates the level or type of authentication calculation that should be returned. A value of 0x01 refers to Level 1 Authentication, 0x02 refers to Level 2 Authentication, and so on. A value of 0x00 is illegal.

Authentication Seed: The Authentication Seed value is used as the starting value, or seed, for some Authentication Methods. If an Authentication Method requires a seed and one is not sent in the Initiate Authentication Calculation Query the seed value defaults to 0. The same seed value is used for all modules and authentication methods. If the seed value is longer than required by an Authentication Method it is truncated, the high order bytes discarded.

NOTE

If an Authentication Calculation is in progress when this command is received by the EGM the EGM aborts the calculation and starts the new Authentication Calculation. Issuing a new Authentication Calculation while the EGM is calculating is not recommended. The Master can determine the state of the EGM using the Status Response [0x01 – SQ] command.

3.2.8 Initiate Authentication Calculation Response [0x84 – IACR]

Indicate that the EGM has received *Start Authentication* request. [EGM ⇨ Master]

Cmd = Initiate Authentication Calculation Response (0x84)	**Length** = (0x04)	**Status**	**CRC**
1 byte	1 byte	1 byte	2 bytes

Status:

0x01 – ACK/NACK [Off – Cannot Acknowledge, On – Acknowledged]

0x02 – Calculation Started [Off – Not started, On – Started]

0x04 – Level Compliance Error [Off – Valid Level, On – Invalid Level requested]

4 Authentication File Format

The authentication file data has a minimum of information to make its parsing easier. The essential information includes the following:

- Information to allow identification of EGM software (manufacturer, game version, …)
- Authentication level
- Authentication method used
- File(s) verified and their authentication data (hash, signature, …)
- Elapsed time (in seconds) since the file authentication was last performed

Each line is an arbitrary length, which may be split over several 248 byte application data frames. The byte data are standard printable ASCII characters, and each line of data is terminated with the PC carriage return and line feed (0x0D 0x0A.).

The proposed file format is divided into sections. Some of the sections are standard while others are manufacturer-specific. The order of the sections is not specified. It is, however, preferable that the authentication file data section be at the top of the file to allow for early resource allocation. The predefined section headers are:

[Authentication File Data]

[Module List]

[Manufacturer Specific Section]

The [Module List] section is the list of modules on the game. These can be any logical grouping of information. The format for this section is:

[Name]

[Elapsed time since authentication was last performed on this module (in seconds)]

[Authentication method]

[Authentication seed]

[Authentication data]

Each module is given a unique name [Name]. This name will be used by an application on the Master to look up the proper authentication value for comparison.

The *[Manufacturer Specific Section]* data may be a convention between the manufacturer and the regulatory agency used to improve the efficiency of the authentication process. This field can also be used to produce additional information when an error is detected.

4.1 Sample Format [Plain ASCII text]

```
[Authentication File Data]

Authentication Level     : 3
Authentication Seed      : 0123456789ABCDEF
Manufacturer             : Killer Games Inc.
Game Version             : Super Duper Poker

[Module List]

\Super Duper Poker\Game1\Bitmap1.bmp
24
SHA-1
0123456789ABCDEF

\Super Duper Poker\Game1\Main.exe
200
MD5
0123456789ABCDEF

\Super Duper Poker\Game1\Movie.mpg
32567
MD5
0123456789ABCDEF

\Super Duper Poker\Verification\Check.exe
25966
CRC16
0123

\Super Duper Poker\Xfiles\Molder\UFO.exe
5
SHA-1
0123456789ABCDEF

[Manufacturer Specific Section]

Graphics data signature          : 0123456789ABCDEF
Executable data signature        : 0123456789ABCDEF
Paytable data signature          : 0123456789ABCDEF
Game1 data signature             : 0123456789ABCDEF
Game2 data signature             : 0123456789ABCDEF
```

5 Operational Scenarios

It is assumed that the Master is correctly connected to the EGM via an appropriate cable, and that the EGM is configured correctly (if required).

- The Master should poll the EGM with the SQ and wait for a SR from the EGM.
  - If any results are available, then an LASQ can be sent to retrieve the level and time the results were calculated.
  - Then a LARQ can be sent to retrieve the last results.
- If the EGM is 'idle', then the Master can send an IACQ to start a new calculation, and then check the IACR to ensure that the calculation was actually started.
- If a calculation is in progress, then continue polling with SR waiting for Current Calculation to 'Finish', then request the results (LASQ & LARQ).

During each Query, the Response should be checked for various errors, and a modified Query should be resent.

END OF DOCUMENT

Document ID: gsa-p0011.001.01
Released: August 22, 2005