

SLIMP3 client protocol

From SqueezeboxWiki

Contents

- 1 The SLIMP3 Client Protocol
 - 1.1 Overview
 - 1.2 Server -> Client
 - 1.2.1 'D' - Discovery response
 - 1.2.2 'h' - Say hello
 - 1.2.3 'l' - Send data to the lcd/vfd display
 - 1.2.4 'm' - write MPEG data into the player's buffer
 - 1.2.5 '2' - read/write the i2c bus
 - 1.3 Client -> Server
 - 1.3.1 'd' - Discovery request
 - 1.3.2 'h' - Say hello
 - 1.3.3 'i' - IR code
 - 1.3.4 '2' - i2c response
 - 1.3.5 'a' - ack response
- 2 Other protocols supported by the player

The SLIMP3 Client Protocol

Overview

This documents the SLIMP3 protocol, as implemented in SlimServer 5.0.1 and SLIMP3 v2.2 firmware. The protocol is UDP based. The SLIMP3 communicates using a custom UDP-based protocol. This protocol is designed to be extremely light-weight. It gives the server low-level access to the hardware and full control over the user interface, so as to minimize the amount of hardware and software required on the client side.

All packets are preceded by an 18-byte header. The first byte in this header is a token which indicates the format of the rest of the header. Some kinds of packets may have also have variable amounts of data after the header. Variable length data is not delimited - the length is taken from the UDP header. All numbers are unsigned integers, in network order. The server listens on port 1069, and replies to the source port of the packets it receives from the client. The server identifies individual clients by MAC address. The last 6 bytes of any Client -> Server message are the client's MAC address.

Server -> Client

'D' - Discovery response

Field	Value/Description
0	'D' as in "discovery"
1	reserved
2..5	server's IP address, or 0.0.0.0
6..7	server's port
8..17	reserved

On receiving a (d)iscovery request broadcast (see below) from a supported client, the server replies with a unicast (D)iscovery response, containing the IP and port number that the client should contact. If the server's IP address is left blank (0.0.0.0), then the client will instead get the IP address and port from the IP/UDP packet headers. A server replying to a discovery request may redirect a client to another server (for load balancing) using this mechanism. If multiple discovery responses are received, the client will select the first one it receives.

'h' - Say hello

Field	Value/Description
0	'h' as in "hello"1..17 reserved

This is used by the server to determine if previous known clients are up, and to obtain the model and firmware revision of those clients. Clients reply with another (h)ello, see below.

'l' - Send data to the lcd/vfd display

Field	Value/Description
0	'l' as in "lcd"1..17 reserved (ignored)18... variable length string

The Noritake vacuum fluorescent display used in the SLIMP3 implements the standard Hitachi LCD interface, plus additional codes for controlling brightness. The data sheet is in CVS under docs/datasheets. The SLIMP3 protocol allows both "character data" and "commands" to be sent directly to the display. It also allows you to insert a delay between characters. This is used to insert the required delay after the "clear screen" command, and can also be used to create animated sequences. The delay feature is not currently used by the server, and probably shouldn't be used in its current implementation, because the player has to spin for the specified time and can not be interrupted. Instead, delays should be implemented in an interruptible manner on the server side.

The 16-bit codes are:

00XX, where XX specifies a delay in ms, up to 25502XX, where XX is a command03XX, where XX is one of the
--

'm' - write MPEG data into the player's buffer

Field	Value/Description
0	'm' as in "mpeg"1 control signal2..5 reserved (used in earlier

This message feeds data into the client's buffer, and also controls whether the decoder is running. The control signal indicates what the decoder should be doing:

0	decoding1	stopped (but do not reset buffer)3	stopped, reset read pointer to 0
---	-----------	------------------------------------	----------------------------------

The write pointer gives the address in the buffer to which the data should be written. The sequence number is an identifier for the packet. The client must acknowledge each MPEG data packet received with an "ack" message.

'2' - read/write the i2c bus

Field	Value/Description
0	'2' as in "i2c"1..17 reserved18.. string of i2c commands and d

The protocol allows the i2c bus to be manipulated down to the level of discrete start / stop / ack / nack / read / write operations. The data sent after the header is a string of these operations, specified as follows:

wX	write the byte X onto the busr	read a byte from the buss	startp	stopa	ackn	nack
----	--------------------------------	---------------------------	--------	-------	------	------

The client acknowledges all i2c packets. See '2' under the "Client->Server" section.

Please see the Philips web site for more information on i2c. At the bottom of that page, you'll find the i2c spec sheet in PDF format.

Client -> Server

'd' - Discovery request

Field	Value/Description
0	'd' as in "discovery"

1	reserved
2	Device ID, '1' for SLIMP3
3	Firmware rev, eg 0x11 for version 1.1
4..11	reserved
12..17	MAC address

This packet is sent to the broadcast address on port 1069. The client sends this during startup to discover the server's IP address, after the client has configured his own IP interface. The server replies with a unicast 'D' packet - see above. Firmware v2.2 identifies itself as 1.1.

'h' - Say hello

Field	Value/Description	0	'h' as in "hello"	1	Device ID, '1' for SLIMP3	Firmware rev,

The SLIMP3 sends a hello packet to the server when it first starts up, and whenever it receives a hello from the server. This is for the server to learn that a new client is ready, or for the server to query a client to see if here's still there.

NB whilst this may be what the client sends, the server is only interested in the MAC address, and ignores the device id and firmware completely.

'i' - IR code

Field	Value/Description	0	'i' as in "IR"	1	0x002..5	player's time since startup in ticks

'2' - i2c response

Field	Value/Description	0	'2' as in "i2c"	1..17	reserved	18...	string of bytes in response

'a' - ack response

Field	Value/Description	0	'a' as in ack	1..5	reserved	6..7	write pointer	8..9	read poi

This packet is sent in response to an mpeg data packet. Write pointer is ignored by the server (this makes sense, because the client no longer maintains the write pointer). The read pointer is the position in the buffer that the decoder is reading from. The sequence number is the sequence number from the packet being acknowledged. Streaming and Buffering

The SLIMP3's buffer chip is a 128K x 8 (1Mbit) SRAM. It is presented to the server as a 64K x 16 circular buffer. The server sends packets to the client, with the address in the buffer where they are to be stored, and with a sequence identifier. The client acks each packet as it receives it. The server ensures that each packet has been acknowledged by the client, and will resend any packets that are not acknowledged promptly. Once the server has filled the client's buffer it will send zero length data packets, and will monitor the read pointer returned in the client's acknowledgement, sending further packets as necessary to keep the buffer full.

The read pointer is maintained by the client, and is the position from which the decoder is currently reading. The server can order this read pointer to be reset to the beginning of the buffer by sending a control code of "3" to the client in a data packet. These control codes also control whether the decoder is running or not (see 'm' write MPEG data).

At the start of each track, the server will send a number of packets with the control set to "3", in order to allow the client to build up a small buffer before starting decoding. The control with then switch to "0". "1" is used to pause the player.

The SLIMP3's buffer chip is a 128K x 8 (1Mbit) SRAM. It is presented to the server as a 64K x 16 circular buffer. The SLIMP3 maintains two pointers, called rptr and wptr, which track the position at which the DMA controller is reading (out to the decoder) and writing (in from the network). When a new stream is started, the SLIMP3 begins by initializing an empty buffer, with rptr == wptr == 0. It then begins requesting data from the server, starting from 0. The server replies with some data, which the SLIMP3 writes into the buffer. Once the buffer reaches 50% capacity, the decoder is started,

and the rptr begins to increment. The SLIMP3 continues requesting data until the buffer is 90% full. Once the buffer reaches this "almost full" state, the SLIMP3 continually checks buffer usage, requesting more data only when the buffer usage drops below the "almost full" level.

Timeouts due to a lost packet are handled by the client. If the SLIMP3 does not receive a response from the server after 100ms, then it sends the request again.

Other protocols supported by the player

- ICMP

ICMP echo (ping): The SLIMP3 will send an ICMP response and display a message indicating where the ping came from.

ICMP unreachable: This indicates that the server machine is up, but the server process is not running. The SLIMP3 will display a message to the effect of "10.0.0.1: service unreachable - are you sure the SLIMP3 server is running?"

Other ICMP messages are displayed on the screen in a similar manner, but they are otherwise ignored (eg ICMP redirect does nothing, but the player will still tell you, so you know to fix your gateway setting).

- UDP echo

The player will echo any packet sent to it on UDP port 7. This could be used to obtain an RTT estimate. Unlike ping, it does not cause anything to be displayed on the screen.

- DHCP

As of firmware version 1.1, the SLIMP3 can use DHCP to discover its own IP address.

Retrieved from "http://wiki.slimdevices.com/index.php?title=SLIMP3_client_protocol&oldid=10500"

Category: Development

Copyright © 2016 Logitech. All rights reserved.

[Terms of Use](#) | [Privacy & Security](#) | [Register Your Product](#)

