EPSON
EXCEED YOUR VISION

S1V3034x Series Message Protocol Specification

S1V3034x
S1V3S344
S1V3G340

SEIKO EPSON CORPORATION

Rev. 1.23

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

All brands or product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

©SEIKO EPSON CORPORATION 2010, All rights reserved.

Table of Contents

1. Introduction 1

1.1 Scope 1

1.2 Document Structure 1

1.3 Terminology 1

2. Feature Summary 2

2.1 Hardware Feature Summary 2

2.2 Hardware Interface 2

2.3 Standby Mode 2

3. Message Protocol 4

3.1 Basic Specification 5

3.2 Message Structure 7

3.3 Serial Communications Interface 8

3.3.1 Clock Synchronous Serial Communication 9

3.3.2 UART Communications 9

3.3.3 I2C Communication 11

3.3.4 MSGRDY and Communication Methods 13

3.3.5 Checksum Function 15

4. Messages 16

4.1 Introduction 16

4.2 Message Identifier Summary 17

4.3 Error Codes 19

4.4 System Messages 22

4.4.1 Overview 22

4.4.2 Message Flows 22

4.4.2.1 System Initialization 22

4.4.3 System Initialization Message Description 24

4.4.3.1 ISC_RESET_REQ 24

4.4.3.2 ISC_RESET_RESP 25

4.4.3.3 ISC_TEST_REQ 26

4.4.3.4 ISC_TEST_RESP 27

4.4.3.5 ISC_VERSION_REQ 28

4.4.3.6 ISC_VERSION_RESP 29

4.4.3.7 ISC_ERROR_IND 30
4.4.3.8 ISC_MSG_BLOCKED_RESP 31
4.5 UART Messages 32
4.5.1 Overview 32
4.5.2 Message Flow 32
4.5.2.1 Host Message Receive Ready Notification 32
4.5.2.2 UART Communication Settings 34
4.5.3 Message Description 35
4.5.3.1 ISC_UART_CONFIG_REQ 35
4.5.3.2 ISC_UART_CONFIG_RESP 36
4.5.3.3 ISC_UART_RCVRDY_IND 37
4.6 Power Management Message 38
4.6.1 Overview 38
4.6.2 Message Flow 38
4.6.2.1 Standby Mode Entry 38
4.6.2.2 Standby Mode Exit 39
4.6.3 Message Descriptions 40
4.6.3.1 ISC_PMAN_STANDBY_ENTRY_REQ 40
4.6.3.2 ISC_PMAN_STANDBY_ENTRY_RESP 41
4.6.3.3 ISC_PMAN_STANDBY_EXIT_IND 42
4.7 Audio Messages 43
4.7.1 Overview 43
4.7.2 Message Flow 43
4.7.2.1 Audio Output Setting 43
4.7.2.2 Audio Volume Setting 44
4.7.2.3 Audio Mute 45
4.7.2.4 Audio Pause IND 46
4.7.3 Message Descriptions 47
4.7.3.1 ISC_AUDIO_CONFIG_REQ 47
4.7.3.2 ISC_AUDIO_CONFIG_RESP 48
4.7.3.3 ISC_AUDIO_VOLUME_REQ 49
4.7.3.4 ISC_AUDIO_VOLUME_RESP 50
4.7.3.5 ISC_AUDIO_MUTE_REQ 51
4.7.3.6 ISC_AUDIO_MUTE_RESP 52
4.7.3.7 ISC_AUDIO_PAUSE_IND 53
4.8 Streamed Playback Messages 54

4.8.1 Overview 54
4.8.2 Message Flow 54
4.8.2.1 Audio Decoder Setting 54
4.8.2.2 Audio Decoding 55
4.8.2.3 Streamed Playback End Immediately 55
4.8.2.4 Streamed playback End 56
4.8.2.5 Streamed Playback Pause 56
4.8.2.6 Streamed Playback Pause Release 57
4.8.2.7 Streamed Playback Message Flow 57
4.8.3 Message Description 60
4.8.3.1 ISC_AUDIODEC_CONFIG_REQ 60
4.8.3.2 ISC_AUDIODEC_CONFIG_RESP 61
4.8.3.3 ISC_AUDIODEC_DECODE_REQ 62
4.8.3.4 ISC_AUDIODEC_DECODE_RESP 63
4.8.3.5 ISC_AUDIODEC_READY_IND 64
4.8.3.6 ISC_AUDIODEC_PAUSE_REQ 65
4.8.3.7 ISC_AUDIODEC_PAUSE_RESP 66
4.8.3.8 ISC_AUDIODEC_STOP_REQ 67
4.8.3.9 ISC_AUDIODEC_STOP_RESP 68
4.8.3.10 ISC_AUDIODEC_ERROR_IND 69
4.9 Sequenced Playback Messages (only Supported by Products with Built-in ROM) 70
4.9.1 Overview 70
4.9.2 Message Flow 72
4.9.2.1 Sequenced Playback Setup 72
4.9.2.2 Sequencer START 72
4.9.2.3 Sequencer PAUSE 73
4.9.2.4 Sequenced Playback Pause Release 73
4.9.2.5 Sequencer FINALISE IMMEDIATELY 74
4.9.2.6 Sequencer FINALISE 75
4.9.2.7 Sequencer STATUS IND 75
4.9.2.8 Sequenced Playback Message Flow 76
4.9.3 Message Descriptions 78
4.9.3.1 ISC_SEQUENCER_CONFIG_REQ 78
4.9.3.2 ISC_SEQUENCER_CONFIG_RESP 79
4.9.3.3 ISC_SEQUENCER_START_REQ 80
4.9.3.4 ISC_SEQUENCER_START_RESP 81
4.9.3.5 ISC_SEQUENCER_STOP_REQ 82

4.9.3.6 ISC_SEQUENCER_STOP_RESP 83
4.9.3.7 ISC_SEQUENCER_PAUSE_REQ 84
4.9.3.8 ISC_SEQUENCER_PAUSE_RESP 85
4.9.3.9 ISC_SEQUENCER_STATUS_IND 86
4.9.3.10 ISC_SEQUENCER_ERROR_IND 87
4.10 General-purpose Output Control Message (S1V3S344 and S1V3G340 only) 88
4.10.1 Overview 88
4.10.2 Message Flow 88
4.10.2.1 General-purpose Output Port Control 88
4.10.3 Message Descriptions 89
4.10.3.1 ISC_GPOSW_IND 89
4.11 Flash Access Message (S1V3S344 and S1V3G340 only) 90
4.11.1 Overview 90
4.11.2 Message Flow 91
4.11.2.1 Flash Access Mode Setting 91
4.11.3 Message Descriptions 92
4.11.3.1 ISC_SPISW_IND 92
5. Error Processing 93
5.1 Return from Error 93
5.1.1 Return from Fatal Error 93
5.1.2 Return from Non-fatal Error during Streamed Playback 94
5.1.3 Return from Non-Fatal Error during Sequenced Playback 94
5.1.4 Recovering from Other Errors 95
5.1.5 Message Blocking 96
5.2 IND Message 101
5.2.1 Reception of IND and RESP during Transmission of REQ 101
5.2.1.1 IND during Transmission of ISC_RESET_REQ 101
5.2.1.2 IND during Transmission of REQ Other than ISC_RESET_REQ 102
5.2.2 Concurrence of the IND and the RESP 103
5.2.2.1 Concurrence of ISC_RESET_RESP and IND 103
5.2.2.2 Concurrence of Messages Other than ISC_RESET_RESPs and IND 103
6. Use Case of Message Protocol 105
7. Sample Program Specifications 116
7.1 Overview 116
7.2 Obtaining Sample Programs 116

7.3 Types of Sample Programs 116
7.4 File Configuration 117
7.4.1 Main Program Files 117
7.4.2 S1V3034x Control API Function Definition Files 117
7.4.3 Message Files 118
7.4.4 Other Source Files 119
7.5 Main Program Specifications 120
7.5.1 main_streaming.c 120
7.5.2 main_sequencer.c 122
7.5.3 main_power_management.c 123
7.5.4 main_streaming_simple.c 125
7.5.5 main_sequencer_simple.c 126
7.6 S1V3034x Control API Function Specifications 127
7.6.1 SPI_Initialize 127
7.6.2 SPI_SendReceiveByte 128
7.6.3 SPI_SendMessage 129
7.6.4 SPI_ReceiveMessage 130
7.6.5 SPI_SendMessage_simple 131
7.6.6 SPI_ReceiveMessage_simple 132
7.6.7 GPIO_ControlChipSelect 133
7.6.8 GPIO_ControlStandby 134
7.6.9 GPIO_ControlMute 135
8. Appendix 136
8.1 Confirming Communication Link 136
8.2 Examples of SPI Register Specifications 137
9. Quick Start 151
9.1 Streaming Playback 152
9.2 Sequence Playback 153
9.3 Immediate Ending of Streaming Playback 154
9.4 Immediate Ending of Sequence Playback 155
Revision History 156

1. Introduction

1.1 Scope

This document specifies the message protocol used to control/configure and transfer data to/from the S1V3034x.

1.2 Document Structure

Section 2 summarises the general features of the S1V3034x and the hardware resources.

Section 3 summarises the basic specification of the message protocol used to communicate with S1V3034x and the outline of the hardware interface.

Section 4 describes the details of the message supported by S1V3034x.

Section 5 describes the details of the return procedure when the error occurs in S1V3034x.

1.3 Terminology

In this specification the following terminology is defined:

Audio: voice or audio prompt output by the device.

Audio data: Epson original format encoded voice data.

EOV: abbreviation for the Epson Original Voice

EOV file format: the EOV audio data container file format.

Audio Decoding: process that generates PCM data from audio data.

Audio output: PCM data output from the S1V3034x.

Audio playback: process that combines audio decode and audio output. There are two forms of audio playback; streamed playback and sequenced playback

Streaming playback: Playback format in which audio is played back while audio data is sent from the host to the S1V3034x.

Sequence playback: Playback format in which audio is played back from audio data stored in the internal ROM of the S1V3034x.

Phrase file: audio data files sequenced for playback in sequenced playback mode as a voice guidance phrase.

S1V3034x: Generic name for devices including the S1V30345 to S1V30340 and the S1V3S344 and S1V3G340.

The S1V3S344 and S1V3G340, respectively, have internal flash memory and external SPI flash memory.

L: the logical value 0.
H: the logical value 1.

2. Feature Summary

2.1 Hardware Feature Summary

The S1V3034x supports the following function.

- Decode of files in the Epson Original Voice (EOV) file format
- Streamed playback of EOV audio
- Sequenced playback of EOV audio
- Error detection function using 8-bit checksum
- General-purpose output ports*1

*1: Refer to Table 2.1 for specifics of supported functions.

2.2 Hardware Interface

The S1V3034x supports the following hardware interfaces:

- Serial communication interface
- 16-bit DA converter

The serial communications interface supports bi-directional data transfers between the S1V3034x and the host. Communications can be in clock synchronized serial, UART, or I2C formats.

The 16-bit DA converter supports audio output from the S1V3034x.

2.3 Standby Mode

S1V3034x supports a standby mode which stops the system clock in order to reduce power consumption.

Table 2.1 Function comparison table

Function type			S1V3034x	S1V3S344	S1V3G340
Internal voice data memory *2	ROM	128 Kbytes	○	–	–
		384 Kbytes	○	–	–
		640 Kbytes	○	–	–
	FLASH	512 Kbytes	–	○	–
External voice data memory control function	External SPI flash		–	–	○ (Max. 16 Mbytes)
Host interface	Clock-synchronized serial interface		○	○	○
	Asynchronous (UART)*3		○	○	○
	I2C		○	○	○
Voice playback function	Streaming		○	○	○
	Sequence		○	○	○*4
	Standby mode		○	○	○
General-purpose output pin	GPO		–	○ (x9)	○ (x7)

*2: Playback time: 128 bytes (approx. 1 minute/16 kbps), 384 bytes (approx. 3 minutes/16 kbps), 512 bytes (approx. 4 minutes/16 kbps), 640 bytes (approx. 5 minutes/16 kbps)

*3: Supported only when the system clock frequency is 32.768 kHz.

*4: When using external SPI flash

*5: Also refer to *S1V3034x Series Guidebook*.

3. Message Protocol

The S1V3034x operates as a companion device. A serial communication interface, consisting of the SCKS, SIS, SOS, and NSCSS pins, is used to communicate with the host. One of the following three communication formats can be selected for use:

- Clock synchronous serial (using SCKS, SIS, SOS, NSCSS)
- UART (using SIS, SOS)
- I2C (using SCKS, SIS)

The host device can configure, control and stream data to S1V3034x using ISC(Inter System Communication) messages.

The message protocol defines the mechanism for the sending and receiving of ISC message, the valid message sequences and the contents of the ISC messages which may be sent.

This model of host and S1V3034x communication is presented in Figure 3.1.

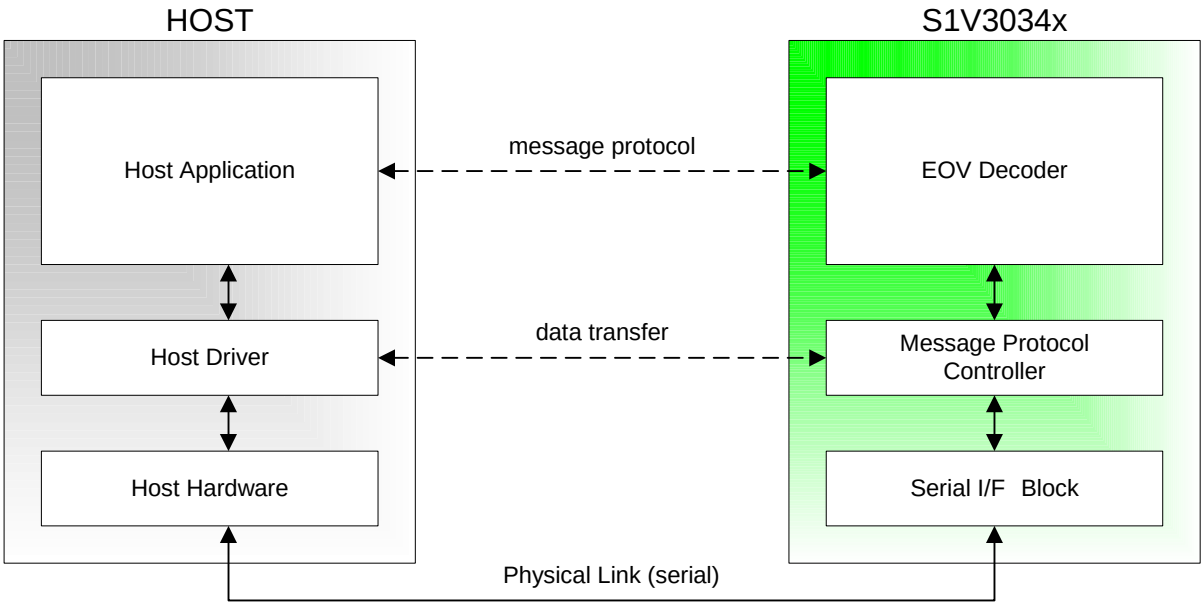


Figure 3.1 Host-S1V3034x communication

3.1 Basic Specification

The Basic specification of the message protocol and its relationship with S1V3034x are as follows.

- The S1V3034x operates by receiving messages from the host.
- A message is composed of a header part of fixed length, containing in the first byte the length of the message, followed by a data part of variable-length.
- There are three types of messages defined:
  1) REQuest
     A message type transmitted from host to S1V3034x. It is used for the control of, configuration of and audio data transfer to S1V3034x.
  2) RESPonse
     A message type transmited from host to S1V3034x. It is used to notify the host that S1V3034x has received the REQ from the host.
  3) INDication
     A message type transmitted from S1V3034x to host. It is used for notifying the host of events other than those related to REQ message reception. S1V3034x may also transmit the IND to the host while the host is transmitting a REQ because S1V3034x can transmit the IND on its ouput line regardless of whether transmission of a REQ is taking place or not.

     Note that the following three messages are sent to the device from the host:

     - ISC_UART_RCVRDY_IND
       - Used to receive messages from the S1V3034x when using UART.
     - ISC_GPOSW_IND (with S1V3S344 and S1V3G340 only)
       - Used to control general-purpose output ports.
     - ISC_SPISW_IND (with S1V3S344 and S1V3G340 only)
       - Used to set the device to flash access mode.

     The IND message is not accompanied by a RESP message.
- REQ and RESP are always transmitted and received as a pair.
- **The host must wait for a RESP from S1V3034x after transmitting a REQ. The host must not transmit a new REQ until the RESP to it is received.**

Figure 3.2 outlines the general message flow for these types:

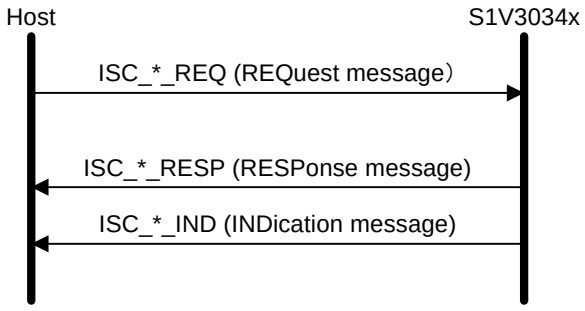


Figure 3.2 General message flow

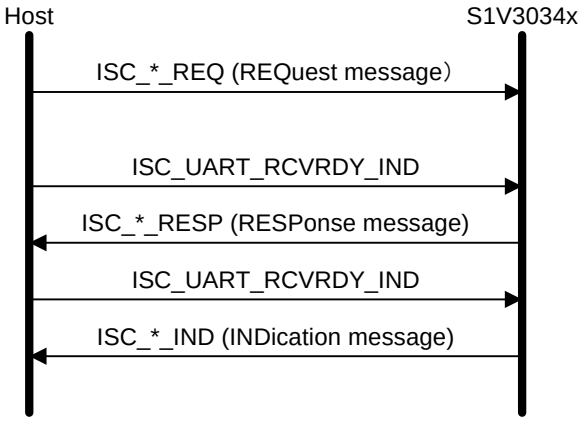


Figure 3.3 UART communication message flow

3.2 Message Structure

The ISC message consists of a variable-length data part and a header part of fixed length.

Table 3.1 shows the definition of these parts. The S1V3034x can receive messages of up to 4,095 bytes in length. All ISC messages must comply with this length restriction. **Note that correct operation cannot be guaranteed for messages exceeding the length restriction.**

0x00 of at least 1 byte must be sent between messages. This is called a padding word. A padding word of at least 3 bytes must be sent when sending REQ messages using the I2C format. A 2-byte padding word is sent when receiving RESP and IND messages.

0xAA must be sent before the first byte of the message to indicate the message start position to the receiving device. This is called the message start command.

S1V3034x interprets the data after 0x00 and 0xAA as a REQ message.

Message from S1V3034x similarly start with a padding word of length one byte or more followed by the message start command.

After recognizing 0x00 and 0xAA, the S1V3034x will not recognize a new message until it receives data the number of bytes of which is consistent with the message length. Consequently, even if a message contains data having the same value as that of the message start command or message ID, this will not cause false operation.

Table 3.1 Structure of message

	Byte	Explanation
Padding word	-2	0x00
Message start command	-1	0xAA
Header part	0 (LSB)	Message length (number of bytes including header)
	1 (MSB)	
	2 (LSB)	Message ID (The data included in the payload is specified.)
	3 (MSB)	
Data part	4	Variable-length data payload

3.3 Serial Communications Interface

Communications between the S1V3034x and host use the SCKS, SIS, SOS, and NSCSS pins. Select one of the following three communication formats:

- Clock synchronous serial
- UART
- I2C

The pins and data output sequence used will depend on the communication format. Connect as shown in the table below. The communication format is selected via pins SHISEL0 and SHISEL1. Note that UART can be selected only when the external clock frequency supplied is 32.768 kHz.

Table 3.2 Communication formats and pins used

Communication format	SCKS	SIS	SOS	NSCSS	Data output sequence
Clock synchronous serial	Serial clock input	Data input	Data output	Slave selection input	MSB first
UART	H-level input	Data input	Data output	L-level input	LSB first
I2C	Serial clock input	Data input/ output	-	L-level input	MSB first

Table 3.3 Communication format selection

CLKSEL	SHISEL0	SHISEL1	Communication format
0 (32.768kHz)	0	0	Clock synchronous serial
		1	I2C
	1	*	UART
1 (12.288MHz)	0	0	Clock synchronous serial
		1	I2C
	1	*	-

3.3.1 Clock Synchronous Serial Communication

The S1V3034x behaves as a slave device with the clock synchronous serial format. The host must output the clock signal to SCKS. MSB-first 8-bit word transfer is used for data transfers. SCKS non-active is logical 1.

The first SIS and SOS bits are output after the first clock edge of SCKS once NSCSS has been asserted, then remain effective until the second clock edge. For specific details of data setup time and hold time, refer to the *S1V3034x series hardware specifications*.

The host outputs the message to be sent to the S1V3034x to the SIS and receives the message from the S1V3034x from the SOS. These are independent, enabling messages to be received and sent simultaneously. However, note that the host cannot send a REQ message until a RESP message has been received from the S1V3034x after the previous REQ message is sent. Only REQ and IND message pairs can be sent and received simultaneously.

The S1V3034x begins sending a message once the clock signal is input to SCKS. For this reason, IND message transmission begins when an error occurs inside the S1V3034x, even when the clock signal is input to SCKS for the host to send messages.

Figure 3.4 Host - S1V3034x clock synchronous serial connection image

3.3.2 UART Communications

Data transfers are LSB-first 8-bit word transfers in the UART format. SIS and SOS initial values are logical 1.

UART communication is initiated by setting 1Bit Time (16 divisions of baud clock) period SIS (SOS) to L. Data is then sent one bit at a time for each 1Bit Time. A parity bit can be set to be added after the data section has been sent. Data transfer is ended by setting the data line to H within 1 to 2BitTime intervals after the parity bit has been set. We recommend performing sampling around midway through BitTime to ensure data consistency when receiving data.

The host outputs the message to be sent to the S1V3034x to the SIS and receives the message from the S1V3034x from the SOS.

To enable reception of messages from the S1V3034x, notification must be sent first to the S1V3034x to indicate that the host can receive messages. The S1V3034x sends the ISC_UART_RCVRDY_IND message (described later) to the S1V3034x to perform this notification. The S1V3034x begins sending the message after ISC_UART_RCVRDY_IND is received.

Figure 3.5 Host - S1V3034x UART communication connection image

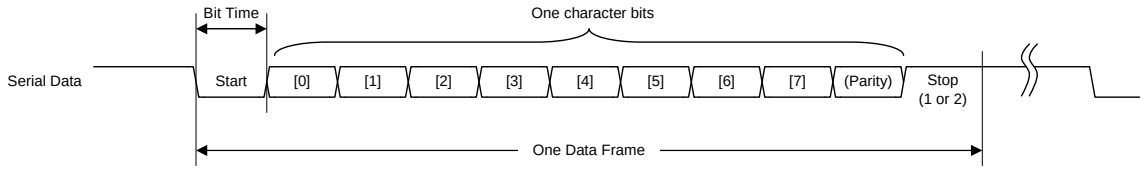


Figure 3.6 UART data format

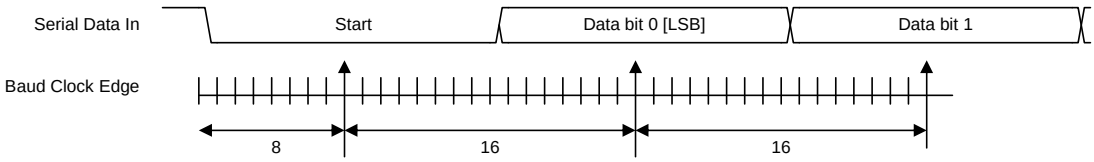


Figure 3.7 UART data sampling timing

3.3.3 I2C Communication

Data transfers are MSB-first 8-bit word transfer when I2C format is used. SCKS non-active and SIS initial values are both logical 1.

I2C communication begins when the host changes SIS from H to L while SCKS is H. This is called the start condition. Except when the start condition is satisfied, SIS can normally be changed only when SCKS is L. Once the start condition is satisfied, the host sends the S1V3034x 7-bit slave ID (0110110) to the SIS. The next bit should be set to L if the host is sending data, and H if the host is receiving data. Setting the SIS to L after receipt of data for the device receiving data constitutes an Ack. Setting the SIS to H constitutes an Nack.

REQ message transmission

When sending a REQ message, the host can start REQ message transmission as specified in this manual following the 3-byte padding word (0x00) and 1-byte message start command (0xAA). The S1V3034x returns Ack each time 8 data bits are received.

For host to end REQ message transmission, SCKS changes SIS from L to H during the H interval after the S1V3034x returns Ack. This is called the stop condition. Except when the stop condition is satisfied, SIS can normally be changed only when SCKS is L.

RESP and IND message receipt

For RESP and IND messages, RESP and IND message transmission as specified in this manual can be started following the 2-byte padding word (0x00) and 1-byte message start command (0xAA) from the S1V3034x.

For the host to end RESP and IND message receipt, Nack should be returned after 8 bits of data have been received. Finally, the stop condition must be satisfied.

However, please transmit IND message sent to the S1V3034x from the host according to the above-mentioned “REQ message transmission” and “Figure 3.9 I2C REQ message timing chart”.

The maximum transfer rate is 83.3 kHz in 12C format, when SCKS and SIS are initialized within 480 ns. If more than 480 ns is required due to load capacity and pull-up resistance, the maximum transfer rate will fall below 83.3 kHz. We recommend sequenced playback for I2C format.

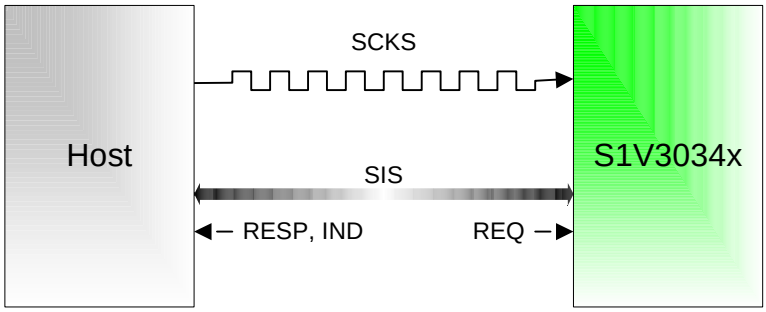


Figure 3.8 Host - S1V3034x I2C connection image

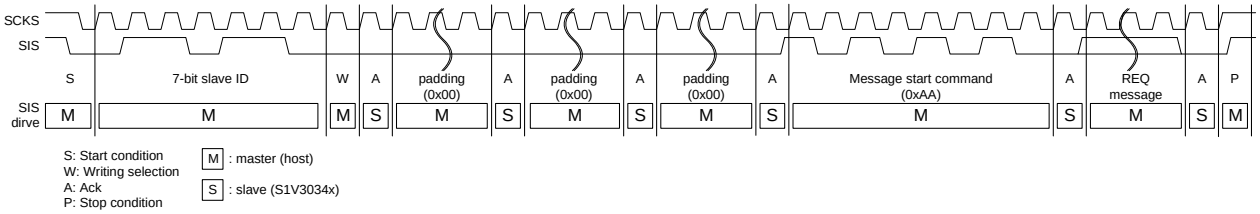


Figure 3.9 I2C REQ message timing chart

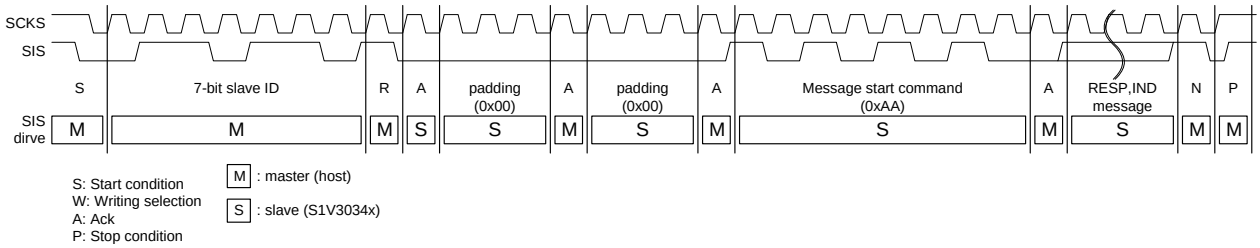


Figure 3.10 I2C RESP/IND message timing chart

3.3.4 MSGRDY and Communication Methods

The S1V3034x includes a MSGRDY pin for notifying the host that preparations are complete for sending a RESP or IND message.

To send a RESP message, MSGRDY is asserted as soon as preparations are complete for sending a RESP message inside the S1V3034x. Output of the padding word sent at the start of a message begins a minimum of t1 after MSGRDY begins. MSGRDY shuts down a minimum of t2 after the first padding word bit is output.

MSGRDY is also asserted using similar timing when sending indication messages.

MSGRDY assertion can be disabled while the host is sending a request. **In this manual, the communication methods between the host and S1V3034x are called full-duplex when the MSGRDY interrupt signal is asserted while the host is sending a message and semi-duplex in other cases.**

Select the communication method using ISC_TEST_REQ/RESP. With the full-duplex method, one MSGRDY is asserted for every message at any time when the S1V3034x sends a message. With the semi-duplex method, MSGRDY is not asserted while the host is sending a message, even if the S1V3034x sends a message. If the host is not sending a message, one MSGRDY is asserted for each message, in the same way as with the full-duplex method.

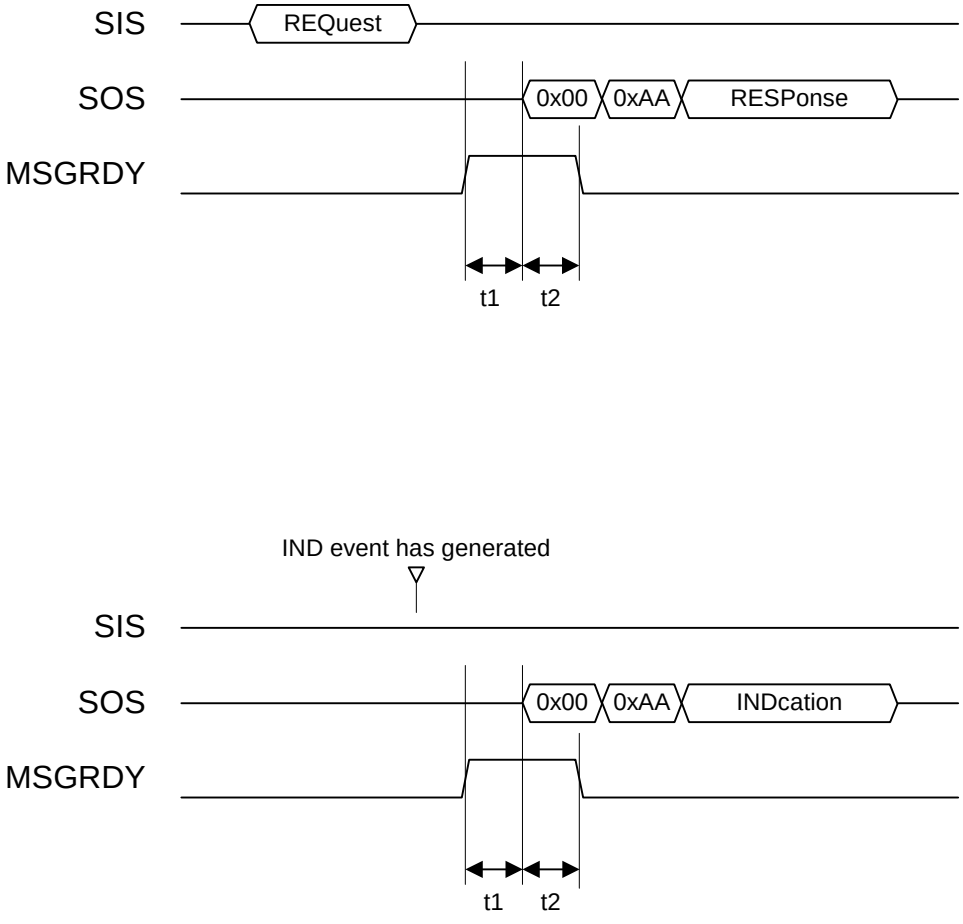
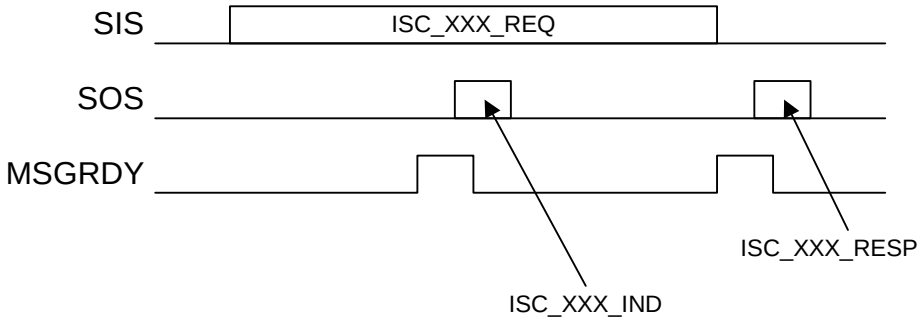


Figure 3.11 MSGRDY timing chart

Code	Item	Min.	Max.	Units
t_1	Time after MSGRDY starts until first bit of padding word (0x00) is output to SIS.	Tscks x 3	-	µs
t_2	Time until MSGRDY ends after first bit of padding word (0x00) is output to SIS.	Tscks x 6	-	µs

Full-duplex method



Semi-duplex method

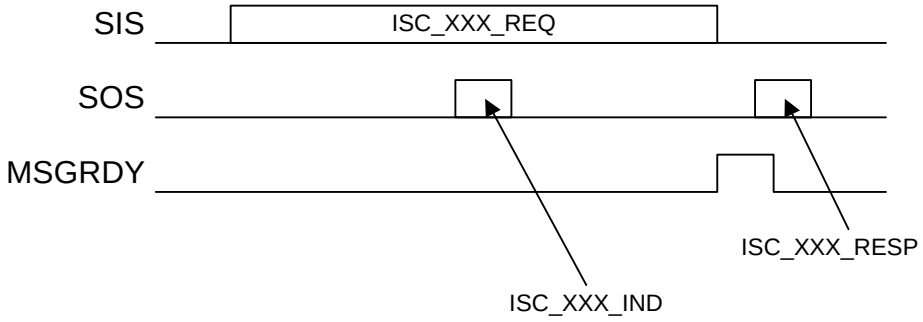


Figure 3.12 Communication methods

3.3.5 Checksum Function

The S1V3034x features a checksum function for detecting transfer data errors arising from noise in communication lines between the host and itself. This function can be enabled or disabled using ISC_TEST_REQ.

When the function is enabled, the host should add checksum data expressing the sum of the data sent as one byte at the end of the message sent. (Use the message length specified in this manual.) The S1V3034x will not operate correctly if the message length used includes checksum data.

The S1V3034x sums data sent from the host for each byte and detects data errors by comparing this against the final checksum data.

If an error is detected, the S1V3034x immediately sends ISC_ERROR_IND to notify the host that a checksum error has occurred.

On receiving ISC_ERROR_IND, the host should initialize the system using ISC_RESET_REQ/RESP.

Checksum data is not added to messages sent by the S1V3034x to the host, even when the checksum function is enabled.

The checksum data consists of the last 8 bits of the result of summing all transmission data.

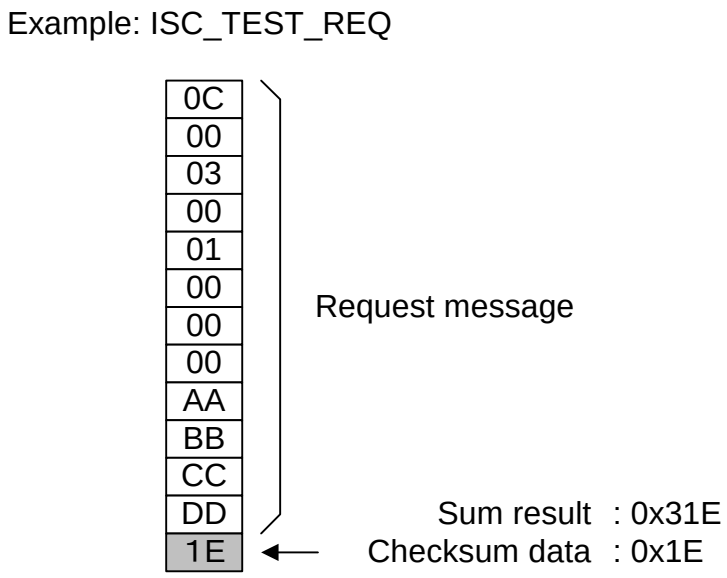


Figure 3.13 Checksum data

4. Messages

4.1 Introduction

Messages are described in detail, grouped according to function. The summary section for each message group describes the individual functions, and the message flow section describes the general message flow.

The host can generally transmit a REQ message at any time. However, after transmitting a REQ message, the host always waits for a RESP message to the REQ message. The host cannot transmit a new REQ message until it receives the RESP message.

If the host transmits a REQ message before receiving a RESP message to the previous REQ message, the S1V3034x will not transmit an error message.

The S1V3034x transmits an error message containing a fatal error code, ISC_ERROR_IND, in the following cases:

- The S1V3034x does not support the received message ID. (0x80E0)
- A parity error occurs during UART communications. (0x8000)
- A checksum error occurs with the checksum function enabled. (0x8FFF)

The S1V3034x transmits an error message containing a non-fatal error code, ISC_AUDIODEC/SEQUENCER_ERROR_IND, when an error occurs during audio playback.

The following events can generate an error during audio playback:

- The beginning of audio data cannot be recognized. (0x5100)
- An audio data CRC (cyclic redundancy checksum) error occurs. (0x5101)
- Audio decoding generates unexpected data. (0x5102)
- The bit rate information in the audio data is incorrect. (0x4078)

When a REQ message sent from the host contains a value not supported by the S1V3034x, it transmits a RESP message indicating a non-fatal error code. The S1V3034x also transmits a RESP message indicating a non-fatal error code if its state is such that it cannot accept a REQ message—for example, ISC_AUDIODEC_PAUSE_REQ or ISC_PMAN_STANDBY_ENTRY_REQ.

4.2 Message Identifier Summary

Table 4.1 summarises the message ID of each message supported by S1V3034x.

Table 4.1 Message identifier summary

Message	Purpose	ID	length	Reference
System message				
ISC_RESET_REQ	Software reset	0x0001	0x0006	Table 4.4
ISC_RESET_RESP		0x0002	0x0004	Table 4.5
ISC_TEST_REQ	Sets the key code & communication method Checksum function on/off	0x0003	0x000C	Table 4.6
ISC_TEST_RESP		0x0004	0x0006	Table 4.7
ISC_VERSION_REQ	Version & support information	0x0005	0x0004	Table 4.8
ISC_VERSION_RESP		0x0006	0x0014	Table 4.9
ISC_ERROR_IND	Fatal error notification	0x0000	0x0006	Table 4.10
ISC_MSG_BLOCKED_RESP	REQ blocked	0x0007	0x0008	Table 4.11
UART message				
ISC_UART_CONFIG_REQ	UART setting	0xFFFF	0x0008	Table 4.12
ISC_UART_CONFIG_RESP		0xFFFE	0x0004	Table 4.13
ISC_UART_RCVRDY_IND	Host message receive ready notification	0xFFFC	0x0004	Table 4.14
Audio message				
ISC_AUDIO_CONFIG_REQ	Audio output set-up	0x0008	0x000C	Table 4.18
ISC_AUDIO_CONFIG_RESP		0x0009	0x0006	Table 4.19
ISC_AUDIO_VOLUME_REQ	Set volume(analog gain)	0x0010	0x0006	Table 4.20
ISC_AUDIO_VOLUME_RESP		0x0011	0x0006	Table 4.21
ISC_AUDIO_MUTE_REQ	Mute of audio output	0x000C	0x0006	Table 4.22
ISC_AUDIO_MUTE_RESP		0x000D	0x0006	Table 4.23
ISC_AUDIO_PAUSE_IND	Pause of audio output	0x007C	0x0004	Table 4.24
Power management message				
ISC_PMAN_STANDBY_ENTRY_REQ	Standby mode entry	0x0064	0x0004	Table 4.15
ISC_PMAN_STANDBY_ENTRY_RESP		0x0065	0x0006	Table 4.16
ISC_PMAN_STANDBY_EXIT_IND	Standby mode exit	0x0066	0x0004	Table 4.17

Streaming playing message				
ISC_AUDIODEC_CONFIG_REQ	Set-up of audio decoder	0x006B	0x0010	Table 4.25
ISC_AUDIODEC_CONFIG_RESP		0x006C	0x0006	Table 4.26
ISC_AUDIODEC_DECODE_REQ	Transfer and start of playback of audio data	0x006D	variable	Table 4.27
ISC_AUDIODEC_DECODE_RESP		0x006E	0x0006	Table 4.28
ISC_AUDIODEC_READY_IND	Indicates ready to accept further data	0x006F	0x0011	Table 4.29
ISC_AUDIODEC_PAUSE_REQ	Temporarily stop (pause) audio decoder output	0x0070	0x0008	Table 4.30
ISC_AUDIODEC_PAUSE_RESP		0x0071	0x0006	Table 4.31
ISC_AUDIODEC_STOP_REQ	Stop of audio Decoding and output	0x0072	0x0006	Table 4.32
ISC_AUDIODEC_STOP_RESP		0x0073	0x0014	Table 4.33
ISC_AUDIODEC_ERROR_IND	Non-fatal audio decoder error	0x007B	0x0006	Table 4.34
Sequence playing message				
ISC_SEQUENCER_CONFIG_REQ	Set-up of sequenced playback.	0x00C4	variable	Table 4.35
ISC_SEQUENCER_CONFIG_RESP		0x00C5	0x0006	Table 4.36
ISC_SEQUENCER_START_REQ	Begin sequenced playback	0x00C6	0x0006	Table 4.37
ISC_SEQUENCER_START_RESP		0x00C7	0x0006	Table 4.38
ISC_SEQUENCER_STOP_REQ	Stop sequenced playback	0x00C8	0x0004	Table 4.39
ISC_SEQUENCER_STOP_RESP		0x00C9	0x0006	Table 4.40
ISC_SEQUENCER_PAUSE_REQ	Pause sequenced playback	0x00CA	0x0006	Table 4.41
ISC_SEQUENCER_PAUSE_RESP		0x00CB	0x0006	Table 4.42
ISC_SEQUENCER_STATUS_IND	Completion of decode of each audio file and completion of sequenced playback	0x00CC	0x0006	Table 4.43
ISC_SEQUENCER_ERROR_IND	Playing sequence error	0x00CD	0x0006	Table 4.44
General-purpose output control message				
ISC_GPOSW_IND	General-purpose output control	0xFF01	0x0006	Table 4.45
Flash access message				
ISC_SPISW_IND	Flash access mode setting	0xFF00	0x0004	Table 4.46

4.3 Error Codes

Error codes generated by the S1V3034x are partitioned as follows:

0x0000 = No Error

0x0001 to 0x3FFF = Reserved (internal) non-fatal error

0x4000 to 0x7FFF = Non-reserved (user) non-fatal error

0x8000 to 0xFFFF = Fatal error

In the normal course of operation, the host should not receive a reserved error from S1V3034x.

Non-fatal errors can be viewed as warnings, from which the system recovers, and have no adverse effect on system behaviour.

Non-fatal errors are communicated to the host processor via the appropriate RESP message, or, in an application specific error IND (ISC_AUDIODEC_ERROR_IND for example).

A RESP message that includes an error code indicates that the REQed action could not be completed.

Fatal errors cannot be recovered from without resetting the device. Fatal errors are communicated to the host using the ISC_ERROR_IND message. The following reset mechanisms are supported:

1) Hardware Reset, using the S1V3034x NRESET pin.

2) Software reset that uses ISC_RESET_REQ message

Please refer to 4.4.3.1 for the scope of the initialization carried out by these resets.

The following tables summarise all non-fatal and fatal errors that may be triggered by the host.

Table 4.2 Non-fatal error code summary

Error code(hex)	Associated Messages	Meaning
General error codes		
0x4001	N/A	Reserved
0x4002	N/A	Reserved
0x4003	N/A	Reserved
0x4004	ISC_TEST_REQ	The key code necessary for descrambling has already been registered.
0x4005	N/A	Reserved
0x4006	N/A	Reserved
Error codes related to Audio		
0x4020	ISC_AUDIO_CONFIG_REQ	The audio setting is invalid.
0x4021	ISC_AUDIO_CONFIG_REQ	A selected configuration parameter is outside the allowed range.
	ISC_AUDIO_VOLUME_REQ	
0x4022	N/A	Reserved
0x4023	N/A	Reserved
0x4024	N/A	Reserved
0x4025	N/A	Reserved
0x4026	N/A	Reserved
0x4027	N/A	Reserved
0x4028	ISC_AUDIO_CONFIG_REQ	Incompatible data output routing setting
0x4029	ISC_AUDIO_CONFIG_REQ	Incompatible sampling frequency setting
	(ISC_AUDIODEC_CONFIG_REQ)	
0x402A	N/A	Reserved
0x402B	N/A	Reserved
0x402C	N/A	Reserved
Error codes related to Audio decoder setting		
0x4060	ISC_AUDIODEC_CONFIG_REQ	Input data is invalid.
	ISC_AUDIODEC_DECODE_REQ	
0x4061	N/A	Reserved
0x4062	N/A	Reserved
0x4063	ISC_AUDIODEC_PAUSE_REQ	Pause REQ while paused.
0x4064	ISC_AUDIODEC_PAUSE_REQ	Pause release REQ while playing
0x4065	N/A	Reserved
0x4066	N/A	Reserved
0x4067	N/A	Reserved
0x4068	N/A	Reserved
0x4069	N/A	Reserved
0x406A	N/A	Reserved
0x406B	N/A	Reserved
0x406C	N/A	Reserved

0x406D	N/A	Reserved
0x406E	N/A	Reserved
0x406F	N/A	Reserved
0x4070	N/A	Reserved
0x4071	N/A	Reserved
0x4072	N/A	Reserved
0x4073	N/A	Reserved
0x4074	N/A	Reserved
0x4075	N/A	Reserved
0x4076	N/A	Reserved
0x4077	All audio decoder messages	Error in audio message sequence
0x4078	ISC_AUDIODEC_ERROR_IND/ ISC_SEQUENCER_ERROR_IND	Bitrate specification error
Error codes related to Standby mode		
0x40C0	N/A	Reserved
0X40C1	ISC_PMAN_STANDBY_EXIT_RESP	Standby entry not yet complete.
Error codes related to Sequencer		
0x4180	N/A	Unexpected message
0x4181	ISC_SEQUENCER_CONFIG_REQ	Illegal configuration
0x4182	ISC_SEQUENCER_PAUSE_REQ	Pause failure (pause requested while already paused or requested unpause while already playing back).
0x4183	ISC_SEQUENCER_CONFIG_REQ	File type not supported
Error codes related to Audio decoding		
0x5100	ISC_AUDIODEC_ERROR_IND/ ISC_SEQUENCER_ERROR_IND	The file header cannot be acquired.
0x5101	ISC_AUDIODEC_ERROR_IND/ ISC_SEQUENCER_ERROR_IND	Cyclic Redundancy Check has detected an error in the audio data.
0x5102	ISC_AUDIODEC_ERROR_IND/ ISC_SEQUENCER_ERROR_IND	Unexpected data while the Decoder is operating.

Table 4.3 Fatal error code summary

Error Code (hex)	Associated Messages	Meaning
Serial communication interface error codes		
0x80E0	All messages	Unsupported Message ID
0x8000	All messages	UART communication error
0x8FFF	All messages	Checksum error

4.4 System Messages

4.4.1 Overview

System Initialisation Messages are used for the following functions:

- Software reset
- Registration of the key code
- Configuration of the communication mechanism
- Reporting the version number and supported functions
- Checksum function on/off

4.4.2 Message Flows

4.4.2.1 System Initialization

S1V3034x supports system reset via the ISC_RESET_REQ/RESP message sequence in addition to a hardware reset via the NRESET pin.

The ISC_TEST_REQ message contains the key code used to enable de-scrambling of the audio data. After resetting the hardware, the host must wait time t1 for system initialisation then send the ISC_TEST_REQ message and wait to receive the ISC_TEST_RESP message. The ISC_TEST_REQ message can also be used to set the communication method.

The ISC_VERSION_REQ/RESP message is used to acquire the hardware version number and details of the features that are supported by the system.

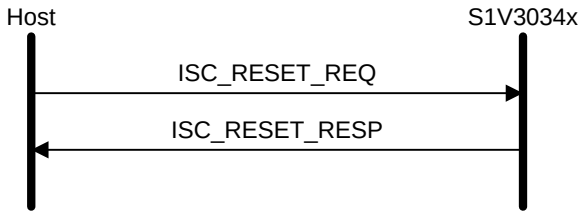


Figure 4.1 Software reset message flow

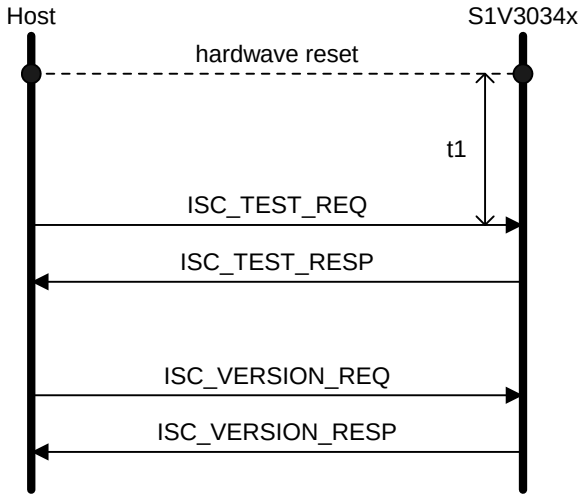


Figure 4.2 System initialization message flow

Sign	Parameter	Min	Max	Unit
t1	Time from hardware reset to being able to communicate with the S1V3034x.	120	-	ms
Note1	There are no restrictions on the transmission of padding bytes during the period t1.			

4.4.3 System Initialization Message Description

4.4.3.1 ISC_RESET_REQ

Table 4.4 ISC_RESET_REQ

Direction	Host to S1V3034x	
Purpose	Used to execute software reset. This is the only valid message after a fatal error.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0001 - ISC_RESET_REQ
3	msg_id (msb)	
4	boot_id	0x00: Reset is executed.
5	reserved	0x00

ISC_RESET_REQ does not re-initialise the serial interface or audio volume (as specified via the ISC_AUDIO_CONFIG_REQ and/or ISC_AUDIO_VOLUME_REQ messages). Please use the hardware reset via the NRESET pin if a full reset of the the S1V3034x is desired.

The audio output buffer contents are not cleared by the ISC_RESET_REQ message but the buffer is re-initialised and cleared on the completion of playback.

The scope of the initialization carried out in response to a ISC_RESET_REQ message and the status of each configuration setting are shown below:

Component			**Status after ISC_RESET_REQ**
Serial interface		Invalid	No change
Playback flow	Error code	Valid	Cleared
	Streaming playback	Valid	Must be re-configured using ISC_AUDIODEC_CONFIG_REQ
	Sequence playback	Valid	Must be re-configured using ISC_SEQUENCER_CONFIG_REQ
Register setting	ISC_TEST_REQ	Valid	Configured for Half duplex (msg_ready_enable = 0x0000)
			Checksum OFF (checksum enable = 0x0000)
			Configured with no keycode (key = 0x00000000)
	ISC_AUDIO_MUTE_REQ	Valid	Mute disabled (audio_mute_enable = 0x0000)
	ISC_AUDIODEC_PAUSE_REQ	Valid	Pause disabled (pause_enable = 0x0000)
	ISC_SEQUENCER_PAUSE_REQ	Valid	Pause disabled (enable_pause = 0x0000)
Audio output	Audio data input buffer	Valid	Cleared
	Audio output buffer	Invalid	No change. However this is re-initialised on the completion at the end of playback.
	volume	Invalid	No change

4.4.3.2 ISC_RESET_RESP

Table 4.5 ISC_RESET_RESP

Direction	S1V3034x to host	
Purpose	Used to notify host that ISC_RESET_REQ has been received. When the transmission of this message is completed, S1V3034x executes an immediate reset.	
Byte	**Field**	**Value**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0x0002 - ISC_RESET_RESP
3	msg_id (msb)	

4.4.3.3 ISC_TEST_REQ

Table 4.6 ISC_TEST_REQ

Direction	Host to S1V3034x	
Purpose	The key field is used to specify the key used for the unscrambling of audio data. The checksum_enable field is used to enable/disable the checksum function. (See “3.3.5 Checksum Function.”) The msg_ready_enable field is used to set the communication method of the device with the host. (See “3.3.4 MSGRDY and Communication Methods.”)	
Byte	**Field**	**Value**
0	length (lsb)	0x000C
1	length (msb)	
2	msg_id (lsb)	0x0003 - ISC_TEST_REQ
3	msg_id (msb)	
4	checksum_enable (lsb)	0x0000: Checksum function disabled 0x0001: Checksum function enabled
5	checksum_enable (msb)	
6	msg_ready_enable (lsb)	0x0000: Half duplex transmission 0x0001: Full duplex transmission
7	msg_ready_enable (msb)	
8	key (lsb)	Insert EPSON provided key code.
9	key	
10	key	
11	key (msb)	

4.4.3.4 ISC_TEST_RESP

Table 4.7 ISC_TEST_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_TEST_REQ has been received. An error code is appended and is transmitted with the response if the REQ content is invalid.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0004 - ISC_TEST_RESP
3	msg_id (msb)	
4	registration_success (lsb)	0x0000: No Error >0x0000: Error Code
5	registration_success (msb)	

4.4.3.5 ISC_VERSION_REQ

Table 4.8 ISC_VERSION_REQ

Direction	Host to S1V3034x	
Purpose	Used to acquire the hardware version of S1V3034x and information on supported codecs.	
Byte	**Field**	**Value**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0x0005 - ISC_VERSION_REQ
3	msg_id (msb)	

4.4.3.6 ISC_VERSION_RESP

Table 4.9 ISC_VERSION_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_VERSION_REQ has been received. Information on the S1V3034x hardware version and supported codecs is appended.	
Byte	**Field**	**Value**
0	length (lsb)	0x0014
1	length (msb)	
2	msg_id (lsb)	0x0006 - ISC_VERSION_RESP
3	msg_id (msb)	
4	hw_id_int	Hardware version identifier integer part
5	hw_id_frac	Hardware version identifier decimal part
6	fw_version_int	Reserved
7	fw_version_frac	Reserved
8-11	fw_features	Supported decoder codecs: The bit field is defined as follows: 0x00000001: Reserved 0x00000002: Reserved 0x00000004: Reserved 0x00000008: Reserved 0x00000010: Reserved 0x00000040: Reserved 0x00000080: Reserved 0x00000100: Reserved 0x00000200: Reserved 0x00000400: Reserved 0x00000800: Reserved 0x00001000: Reserved 0x00004000: EOV Decoding 0x10000000: Descrambler
12-19	reserved	0x00

4.4.3.7 ISC_ERROR_IND

Table 4.10 ISC_ERROR_IND

Direction	S1V3034x to host	
Purpose	Used to notify the host of all fatal errors. After ISC_ERROR_IND, the only message that may be successfully received by S1V3034x is ISC_RESET_REQ. (See “5. Error Processing.”)	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0000 - ISC_ERROR_IND
3	msg_id (msb)	
4	error_code (lsb)	Error code
5	error_code (msb)	

4.4.3.8 ISC_MSG_BLOCKED_RESP

Table 4.11 ISC_MSG_BLOCKED_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that a REQ has been blocked by S1V3034x. When there is a violation of the message protocol defined in this specification, the REQ is blocked. ISC_MSG_BLOCKED_RESP is transmitted instead of the RESP that is associated with the blocked REQ. (See “5. Error Processing.”)	
Byte	**Field**	**Value**
0	length (lsb)	0x0008
1	length (msb)	
2	msg_id (lsb)	0x0007 – ISC_MSG_BLOCKED_RESP
3	msg_id (msb)	
4	blocked_msg_id (lsb)	Message ID of blocked REQ
5	blocked_msg_id (lsb)	
6	error_code (lsb)	Error code
7	error_code (msb)	

4.5 UART Messages

4.5.1 Overview

UART messages perform the following functions:

- Indicates completion of host message receipt preparations
- Sets UART communication

4.5.2 Message Flow

4.5.2.1 Host Message Receive Ready Notification

When the host receives messages in UART communications with the S1V3034x, notification must be issued to the S1V3034x to indicate that messages can be received. Notification that the host is ready to receive messages is sent to the S1V3034x using ISC_UART_RCVRDY_IND. Thus, the host sends ISC_UART_RCVRDY_IND and receives a RESP message from the S1V3034x each time a REQ message is sent.

The host must also send ISC_UART_RCVRDY_IND to the S1V3034x when receiving IND messages from the S1V3034x, in the same way as for RESP messages.

On receiving ISC_UART_RCVRDY_IND, the S1V3034x begins sending any messages awaiting transmission to the host (the S1V3034x sends nothing if no messages exist).

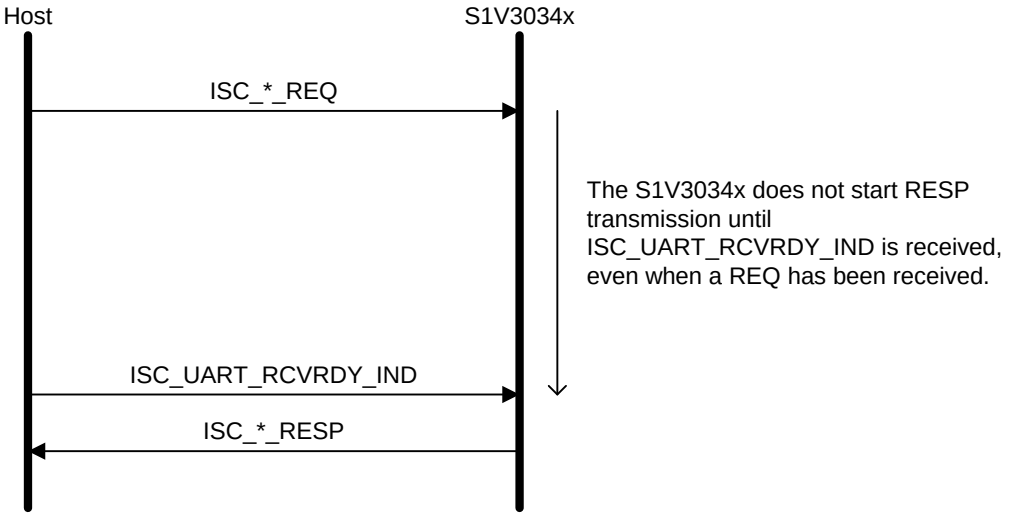


Figure 4.3 REQ/RESP in UART communication

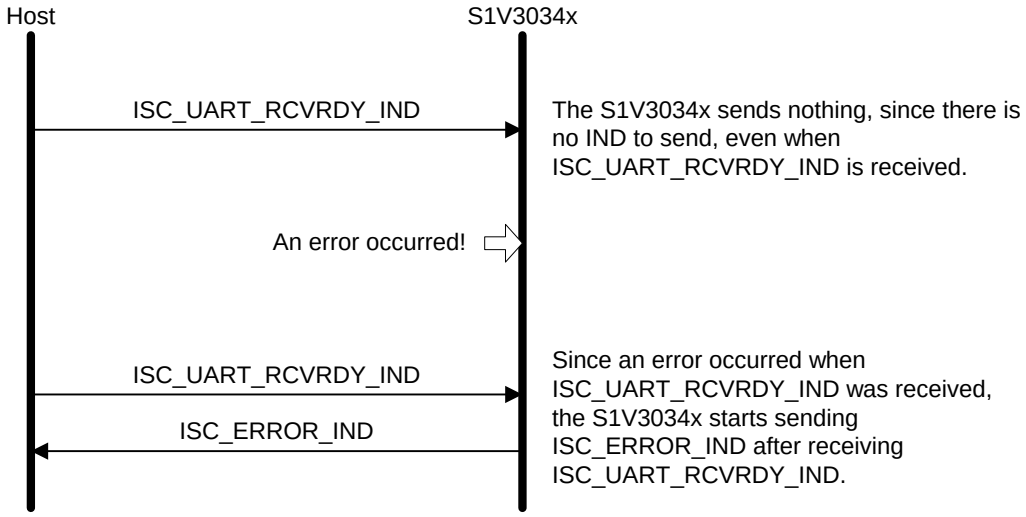


Figure 4.4 IND in UART communication (example: ISC_ERROR_IND)

4.5.2.2 UART Communication Settings

UART communication is set using ISC_UART_CONFIG_REQ/RESP.

The following items can be set using ISC_UART_CONFIG_REQ/RESP.

- Parity
- Stop bit
- Baud rate

When UART communication settings have been changed using ISC_UART_CONFIG_REQ, ISC_UART_RCVRDY_IND should be sent using the post-setting change communication conditions from the t1 communication condition setting change period (communication prohibited) onward.

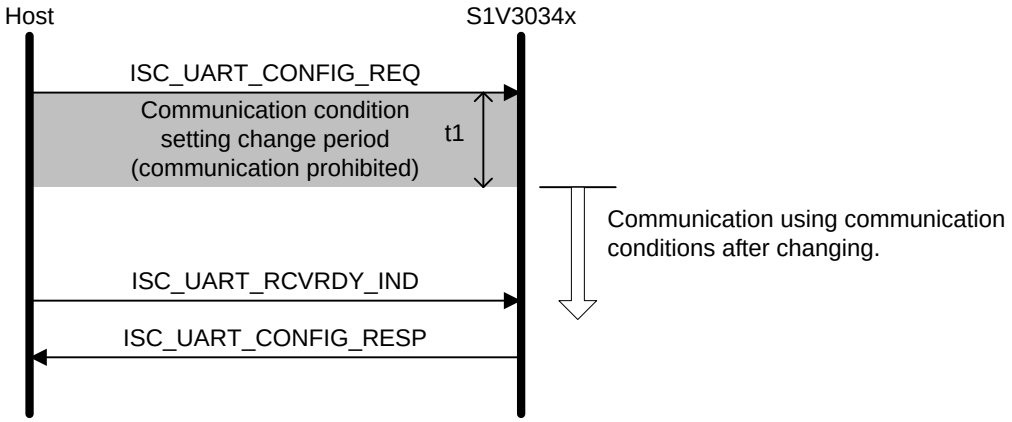


Figure 4.5 UART communication setting message flow

Code	Parameter	Min	Max	Unit
t1	Period in which communication is prohibited after host has sent ISC_UART_CONFIG_REQ	-	100	μs

4.5.3 Message Description

4.5.3.1 ISC_UART_CONFIG_REQ

Table 4.12 ISC_UART_CONFIG_REQ

Direction	Host to S1V3034x	
Purpose	Used for UART communication setting.	
Byte	**Field**	**Value**
0	length (lsb)	0x0008
1	length (msb)	
2	msg_id (lsb)	0xFFFF -ISC_UART_CONFIG_REQ
3	msg_id (msb)	
4-7	uart_setting	7-0[DLL]: Deviser latch 0x05: 460800bps 0x0A: 230400bps 0x14: 115200bps 0x28: 57600bps 0x3C: 38400bps 0x78: 19200bps 0xF0: 9600bps (default) 15-8[DLH]: 0x00 16[STOP]: Stop bit length 0: 1bit (default) 1: 2bit 17[PEN]: Parity enable 0: No parity (default) 1: Parity 18[EPS]: Even parity selection 0: Odd parity (default) Logical 1 is sent or checked for odd number of data word and parity bits. 1: Even parity Logical 1 is sent or checked for even number of data word and parity bits. 19-31: All 0

4.5.3.2 ISC_UART_CONFIG_RESP

Table 4.13 ISC_UART_CONFIG_RESP

Direction	S1V3034x to host	
Purpose	Used to indicate receipt of ISC_UART_CONFIG_REQ.	
Byte	**Field**	**Value**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0xFFFE -ISC_UART_CONFIG_RESP
3	msg_id (msb)	

4.5.3.3 ISC_UART_RCVRDY_IND

Table 4.14 ISC_UART_RCVRDY_IND

Direction	Host to S1V3034x	
Purpose	Used to indicate completion of preparations for receiving host messages. This message is the only message sent from the host to the S1V3034x among IND messages.	
Byte	**Field**	**Value**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0xFFFC -ISC_UART_CONFIG_RESP
3	msg_id (msb)	

4.6 Power Management Message

4.6.1 Overview

The host can cause the S1V3034x to enter standby mode using the ISC_PMAN_STANDBY_ENTRY_REQ/RESP sequence.

When S1V3034x is in standby mode, the system clock of S1V3034x is stopped. The standby mode can be cleared by asserting the STBYEXIT pin.

4.6.2 Message Flow

4.6.2.1 Standby Mode Entry

For the S1V3034x to enter standby mode, the STDBY_EXIT pin must first be set to L. ISC_PMAN_STANDBY_ENTRY_REQ is then sent. The S1V3034x sends a response with no error code if switching to standby mode is possible. Setting STDBYEXIT to H in this state switches the S1V3034x to standby mode.

When the host receives ISC_PMAN_STANDBY_ENTRY_RESP with a no error code, STBYEXIT must be set to H to switch the S1V3034x to standby mode.

If the S1V3034x cannot be switched to standby mode, an error response is sent with an error code appended. The S1V3034x does not switch to standby mode in this case.

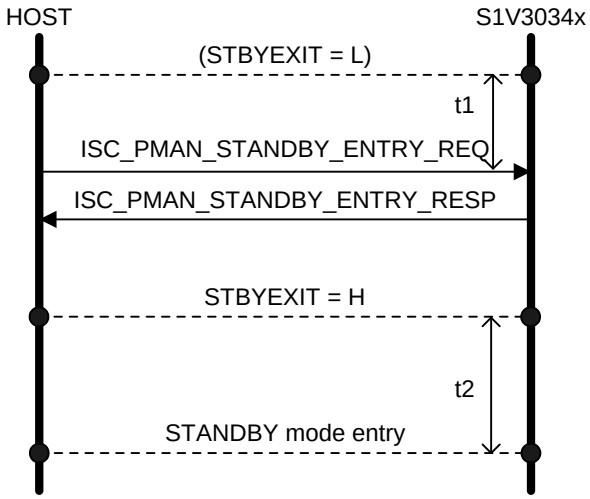


Figure 4.6 Standby mode entry message flow

Sign	Parameter		Min	Max	Unit
t1	Time for ISC_PMAN_STANDBY_ENTRY_REQ to be transmitted after STBYEXIT pin is deasserted		50	-	µs
t2	Time for standby mode to be entered after STBYEXIT pin is asserted	CLOCK: 32.768kHz	-	800	µs
		CLOCK: 12.288MHz	-	100	µs
Note 1	Sending padding bytes during period t1 is permitted however the host must not send any other messages during this period.				

4.6.2.2 Standby Mode Exit

The S1V3034x standby mode can be exited by setting STBYEXIT to L. When STBYEXIT is set to L, the S1V3034x switches from standby mode to normal mode after time t3. The S1V3034x then notifies the host that it has switched to normal mode using ISC_PMAN_STANDBY_EXIT_IND.

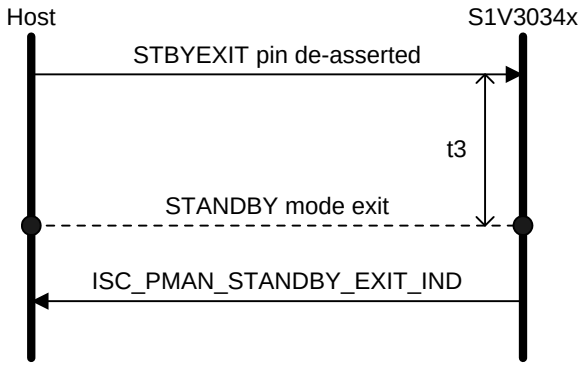


Figure 4.7 Standby mode exit message flow

Sign	Parameter	Min	Max	Unit
t3	Time for MSGRDY pin to be asserted after STBYEXIT pin is deasserted	-	120	ms
Note 1	Transmission of padding bytes during period t3 is permitted.			

4.6.3 Message Descriptions

4.6.3.1 ISC_PMAN_STANDBY_ENTRY_REQ

Table 4.15 ISC_PMAN_STANDBY_ENTRY_REQ

Direction	Host to S1V3034x	
Purpose	Used to REQ standby mode entry.	
Byte	**Field**	**Value**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0x0064 –ISC_PMAN_STANDBY_ENTRY_REQ
3	msg_id (msb)	

4.6.3.2 ISC_PMAN_STANDBY_ENTRY_RESP

Table 4.16 ISC_PMAN_STANDBY_ENTRY_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that ISC_PMAN_STANDBY_ENTRY_REQ has been received. If it is possible to enter standby mode, S1V3034x does so after transmitting ISC_PMAN_STANDBY_RESP with no error. If standby mode entry is not permitted, an error code is appended in the transmitted ISC_PMAN_STANDBY_ENTRY_RESP.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0065 -ISC_PMAN_STANDBY_ENTRY_RESP
3	msg_id (msb)	
4	pman_standby_entry_success (lsb)	0x0000: No error >0x0000: Error code
5	pman_standby_entry_success (msb)	

4.6.3.3 ISC_PMAN_STANDBY_EXIT_IND

Table 4.17 ISC_PMAN_STANDBY_EXIT_IND

Direction	S1V3034x to host	
Purpose	Used to notify the host that the S1V3034x has exited form standby mode and is now in normal mode.	
Byte	**Field**	**Value**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0x0066 - ISC_PMAN_STANDBY_EXIT_IND
3	msg_id (msb)	

4.7 Audio Messages

4.7.1 Overview

Audio messages are used to access the following functions.

- Audio output path and sample rate setting
- Initial and real-time setting of volume

4.7.2 Message Flow

4.7.2.1 Audio Output Setting

The audio sample rate and audio output path must be setup using the ISC_AUDIO_CONFIG_REQ/RESP. An intial audio volume is also selected with this message.

The value set is specified by the absolute value within the range of 0x00 (mute) and 0x01(-48dB) to -0x43(+18dB). If the value set is outwith this range, the audio output is muted, and volume setting with ISC_AUDIO_VOLUME_REQ is disabled and the volume cannot be corrected. In that case, the ISC_AUDIO_CONFIG_REQ shoud be used to once again set the volume to a valid value.

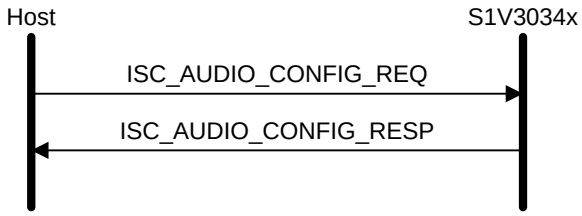


Figure 4.8 Audio INITIALISE message flow

4.7.2.2 Audio Volume Setting

ISC_AUDIO_VOLUME_REQ/RESP is used for real-time setting of the audio volume during audio playback.

An absolute value for the volume is specified by the ISC_AUDIO_CONFIG_REQ and the ISC_AUDIO_VOLUME_REQ is used to specify offsets to this relative value.

The range of valid values for the volume is 0x00 (mute), and 0x01(-48dB) to -0x43(+18dB). The host should select a volume within this range. Please specify a signed 8bit positive number if you want to increase the current volume value. If on the other hand a decrease in volume is required, please specify the offset as a signed 8-bit negative number. When a value is set outwith the allowed range, the audio is muted, and cannot again be corrected with ISC_AUDIO_VOLUME_REQ. In that case, please set the volume to a valid value once again by using the ISC_AUDIO_CONFIG_REQ message.

The host is forbidden to transmit the ISC_AUDIO_VOLUME_REQ command while in the data transfer stage or the audio output standby stage of streamed playback (Figure 4.19).

If volume changes are required after audio data transfer has begun, please set the volume during the interval between the transmission of the ISC_AUDIODEC_DECODE_REQ and the next ISC_AUDIODEC_READY_IND.

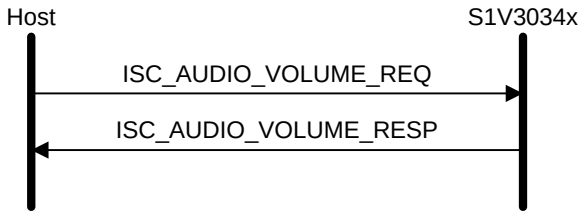


Figure 4.9 Audio volume real-time setting

4.7.2.3 Audio Mute

Audio Output can be muted using the ISC_AUDIO_MUTE_REQ/RESP Message Flow.

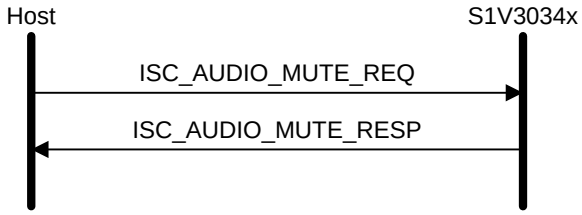


Figure 4.10 Audio MUTE message flow

However, the mute setting is released by ISC_AUDIODEC_STOP_REQ or ISC_SEQUENCER_STOP_REQ even if the host does not unmute the audio with the ISC_AUDIO_MUTE_REQ/RESP sequence.

Therefore, whenever audio playback ends for normal streaming playback, the mute setting is released. In the same way, the mute setting is released when the ISC_*_STOP_REQ message is used to terminate audio playback before it has been completed.

The host must not transmit the ISC_AUDIO_MUTE_REQ command while in the data transfer stage or audio output standby stage during streamed playback. (Figure 4.19).

To ensure correct operation, if mute is dersired after audio data transfer has begun, the host must send Mute commands during the interval between the sending of the ISC_AUDIO_DECODE_REQ and receiving the next ISC_AUDIODEC_READY_IND.

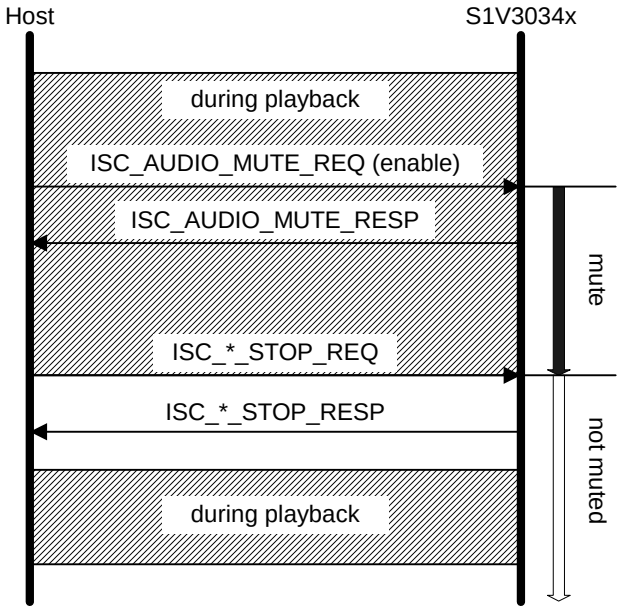


Figure 4.11 Mute setting released by ISC_*_STOP_REQ

4.7.2.4 Audio Pause IND

The S1V3034x notifies the host that audio output has been stopped using ISC_AUDIO_PAUSE_IND.

S1V3034x transmits ISC_AUDIO_PAUSE_IND in the following situations.

1) When S1V3034x has been directed to pause by the host.
2) When streaming playback ends normally.
3) When audio data is not trransmitted in time by the host, and voice output becomes interrupted.

The ISC_AUDIODEC_PAUSE_IND message is transmitted in the following situations.

- When audio playback is cancelled by the host.
- When sequenced playback ends.

However, if 2) or 3) occurs at the same time as the S1V3034x is receiving ISC_AUDIODEC_STOP_REQ, the ISC_AUDIODEC_PAUSE_IND is transmitted after the ISC_AUDIODEC_STOP_RESP. For detailed information, see “5.2.2.2 Concurrence of Messages Other than ISC_RESET_RESPs and IND.”

Figure 4.12 Audio pause IND flow

4.7.3 Message Descriptions

4.7.3.1 ISC_AUDIO_CONFIG_REQ

Table 4.18 ISC_AUDIO_CONFIG_REQ

Direction	Host to S1V3034x	
Purpose	Used to configure the audio output.	
Byte	**Field**	**Values**
0	length (lsb)	0x000C
1	length (msb)	
2	msg_id (lsb)	0x0008 - ISC_AUDIO_CONFIG_REQ
3	msg_id (msb)	
4	reserved	0x00
5	audio_gain	0x00: Mute 0x01: -48dB 0x02: -47dB . . 0x43: +18dB
6	reserved	0x00
7	audio_sample_rate	0x03: 16kHz 0x09: Don’t care
8	reserved	0x00
9	reserved	0x00
10	reserved	0x00
11	reserved	0x00

VOLUME	**CONFIG**	**gain [dB]**
Offsets to volumes in this range may be applied via increases and decreases relative to current volume.	0x00	mute
	0x01	-48
	0x02	-47
	:	:
	0x2F	-2
	0x30	-1
	0x31	0
	0x32	1
	0x33	2
	:	:
	0x42	17
	0x43	18
forbidden	other	mute

4.7.3.2 ISC_AUDIO_CONFIG_RESP

Table 4.19 ISC_AUDIO_CONFIG_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_AUDIO_CONFIG_REQ has been received. An error code is appended and is transmitted with the response if the REQ content is invalid.	
Byte	**Field**	**Values**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0009 - ISC_AUDIO_CONFIG_RESP
3	msg_id (msb)	
4	audio_config_success (lsb)	0x0000: No Error >0x0000: Error Code
5	audio_config_success (msb)	

4.7.3.3 ISC_AUDIO_VOLUME_REQ

Table 4.20 ISC_AUDIO_VOLUME_REQ

Direction	Host to S1V3034x	
Purpose	Used for real-time setting of volume	
Byte	**Field**	**Values**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0010 - ISC_AUDIO_VOLUME_REQ
3	msg_id (msb)	
4	audio_gain_inc (lsb)	2's complement value Gain increment/decrement in dBs *Note 1
5	audio_gain_inc (msb)	

VOLUME	**CONFIG**	**gain [dB]**
Offsets to volumes in this range may be applied via increases and decreases relative to current volume.	0x00	mute
	0x01	-48
	0x02	-47
	:	:
	0x2F	-2
	0x30	-1
	0x31	0
	0x32	1
	0x33	2
	:	:
	0x42	17
	0x43	18
forbidden	other	mute

Note 1: The S1V3034x lacks a function for reading out the volume set value. Volume set values should be controlled by the host.

4.7.3.4 ISC_AUDIO_VOLUME_RESP

Table 4.21 ISC_AUDIO_VOLUME_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_AUDIO_VOLUME_REQ has been received. An error code is appended and is transmitted with the response if the REQ content is invalid.	
Byte	**Field**	**Values**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0011 - ISC_AUDIO_VOLUME_RESP
3	msg_id (msb)	
4	audio_gain_success (lsb)	0x0000: No Error >0x0000: Error Code
5	audio_gain_success (msb)	

4.7.3.5 ISC_AUDIO_MUTE_REQ

Table 4.22 ISC_AUDIO_MUTE_REQ

Direction	Host to S1V3034x	
Purpose	Used to digitally mute the audio output	
Byte	**Field**	**Values**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x000C - ISC_AUDIO_MUTE_REQ
3	msg_id (msb)	
4	audio_mute_enable (lsb)	0x0000: disable mute 0x0001: enable mute
5	audio_mute_enable (msb)	

4.7.3.6 ISC_AUDIO_MUTE_RESP

Table 4.23 ISC_AUDIO_MUTE_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_AUDIO_MUTE_REQ has been received. An error code is appended and is transmitted with the response if the REQ content is invalid.	
Byte	**Field**	**Values**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x000D - ISC_AUDIO_MUTE_RESP
3	msg_id (msb)	
4	audio_mute_success (lsb)	0x0000: No Error >0x0000: Error Code
5	audio_mute_success(msb)	

4.7.3.7 ISC_AUDIO_PAUSE_IND

Table 4.24 ISC_AUDIO_PAUSE_IND

Direction	S1V3034x to host	
Purpose	Indicates that the audio output is paused	
Byte	**Field**	**Values**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0x007C - ISC_AUDIO_PAUSE_IND
3	msg_id (msb)	

4.8 Streamed Playback Messages

4.8.1 Overview

Streamed playback messages are used to access the following functions.

(1) Setting of the audio file format

(2) Transmission of audio data

(3) Streamed playback

(4) Pause and stop of streamed playback

The playback method in which audio data is played back while being transferred from the host to the S1V3034x via the serial communications interface is referred to as **streaming playback**.

4.8.2 Message Flow

4.8.2.1 Audio Decoder Setting

The ISC_AUDIODEC_CONFIG_REQ/RESP message is used to setup the audio decoder.

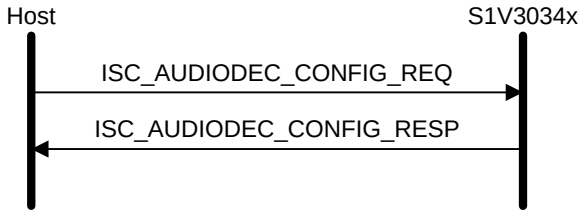


Figure 4.13 Audio decoder setup message flow

4.8.2.2 Audio Decoding

The ISC_AUDIODEC_DECODE_REQ/RESP message is used to send data to the audio data buffers. The ISC_AUDIODEC_READY_IND is used to indicate whether the audio data buffers can accept further data. The host transmits blocks of audio data to the ISC_AUDIODEC_DECODE_REQ. When the host has finished sending the first block of data, S1V3034x automatically begins audio decoding. When the amount of decoded data reaches 256 samples(16ms), audio output begins.

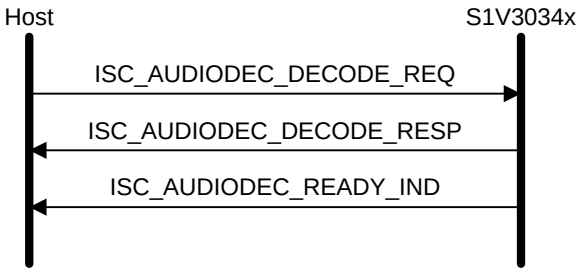


Figure 4.14 Audio Decoding Message flow

4.8.2.3 Streamed Playback End Immediately

ISC_AUDIODEC_STOP_REQ/RESP is used to immediately terminate streamed playback. The S1V3034x immediately ends streaming playback if ISC_AUDIODEC_STOP_REQ is received twice, and the audio data buffer contents are also cleared. **ISC_AUDIODEC_STOP_REQ/RESP must be sent and received twice in order to end streaming playback.** At this time, ISC_AUDIO_PAUSE_IND is not transmitted. However, ISC_AUDIO_PAUSE_IND might be transmitted after the RESP if S1V3034x receives the REQ after a break in the audio due to failure by the host to meet its real-time constarints or if S1V3034x streamed playback ends normally at the same instant as the ISC_AUDIODEC_STOP_REQ is received. For detailed information, see “5.2.2.2 Concurrence of Messages Other than ISC_RESET_RESPs and IND.”

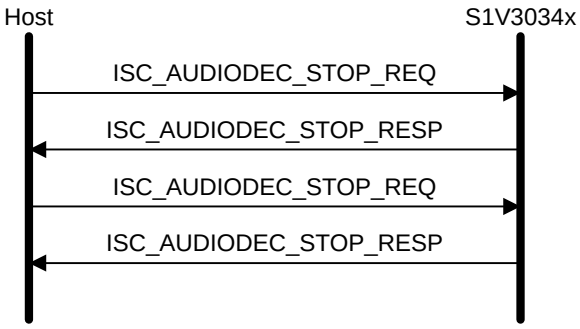


Figure 4.15 Immediate streamed playback end message flow

4.8.2.4 Streamed playback End

The S1V3034x is able to detect the end of an audio file automatically. When streaming playback ends, ISC_AUDIO_PAUSE_IND is sent to the host to indicate streaming playback has ended. The host may then end streamed playback using the ISC_AUDIODEC_STOP_REQ/RESP sequence. In this case, there is no problem if ISC_AUDIODEC_STOP_REQ/RESP is sent and received twice, in the same way as for immediate ending of streaming playback.

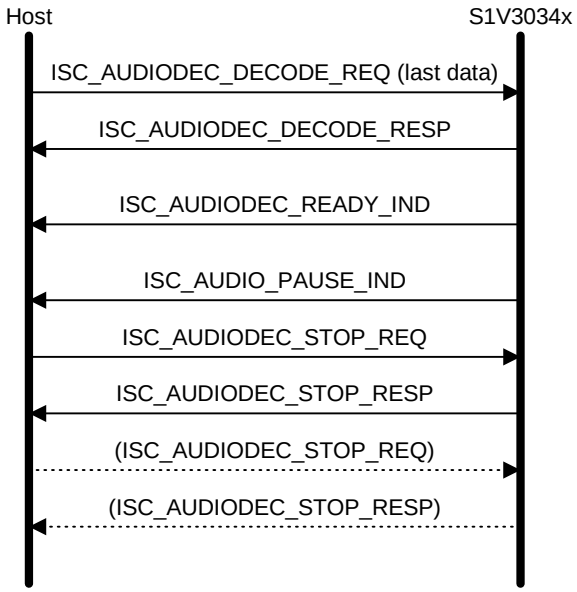


Figure 4.16 Streamed playback end message flow

4.8.2.5 Streamed Playback Pause

The host can pause the audio output during streaming playback using the ISC_AUDIODEC_PAUSE_REQ/RESP message.

When the audio output is paused, S1V3034x transmits the ISC_AUDIO_PAUSE_IND message.

The host is prohibited from transmitting the ISC_AUDIODEC_PAUSE_REQ under the following conditions:

The host must not transmit the ISC_AUDIODEC_PAUSE_REQ command while in the data transfer stage or audio output standby stage of streamed playback (Figure 4.19).

Please send commands to set pause status in the interval between the sending of the ISC_AUDIODEC_READY_IND and the following ISC_AUDIODEC_DECODE_REQ/RESP.

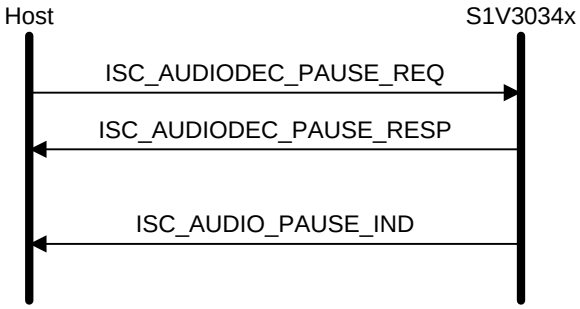


Figure 4.17 Streamed playback pause message flow

4.8.2.6 Streamed Playback Pause Release

The host can restart audio output for paused S1V3034x streaming playback by canceling the pause setting and using ISC_AUDIODEC_PAUSE_REQ/RESP.

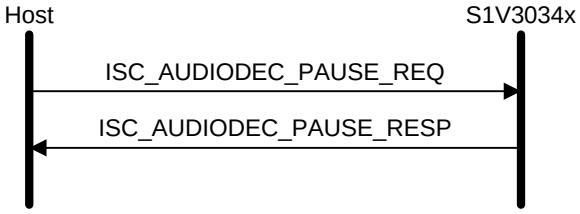


Figure 4.18 Streamed playback pause release message flow

4.8.2.7 Streamed Playback Message Flow

Figure 4.19 is the message flow for streamed playback.

First, audio output is configured using the ISC_AUDIO_CONFIG_REQ/RESP sequence. Before playing back audio for the first time after a reset, please set the audio output configuration setting. This audio configuration need only be done once before the first audio file is played back and is not necessary for subsequent playback of files.

Next, the audio decoder is setup using the ISC_AUDIODEC_CONFIG_REQ / RESP sequence.

Once audio output and decoder setup has been completed, the transmission of audio data starts with the sending of a ISC_AUDIODEC_DECODE_REQ/RESP. After completing the reception of the first of these messages, S1V3034x automatically begins audio decoding. ISC_AUDIODEC_READY_IND is transmitted from S1V3034x when there is enough space in the audio data buffer to accept the next block of audio data. The host must not transmit the next block of audio data before receiving this message. The interval from the sending of an ISC_AUDIODEC_DECODE_REQ to the receipt of a

ISC_AUDIODEC_READY_IND is termed the **data transfer stage**. The data transfer stage is repeated until the following condition is met:

The host has transmitted all audio data.

ISC_AUDIO_PAUSE_IND is transmitted when S1V3034x automatically detects the end of the audio and finishes outputing all transmitted audio. The interval from the sending of the ISC_AUDIODEC_DECODE_REQ transmitting the final block of audio data to the receipt by the host of the ISC_AUDIO_PAUSE_IND that indicates the completion of audio output is termed the **audio output standby stage**. The host terminates streamed playback by sending an ISC_AUDIODEC_STOP_REQ/RESP after the audio output standby stage has ended.

Commands used to alter the status of volume, pause and mute are prohibited during the data transfer stage and audio output standby stage. Therefore, sending of these commands is limited to the interval before the first data transfer stage and that in-between each data transfer stage.

The transmission of messages related to sequenced playback during the streamed playback period is prohibited.

The streamed playback period is assumed to be the interval starting from the host transmiting the ISC_AUDIODEC_CONFIG_REQ to the completion of the transmission of the ISC_AUDIODEC_STOP_RESP by the S1V3034x.

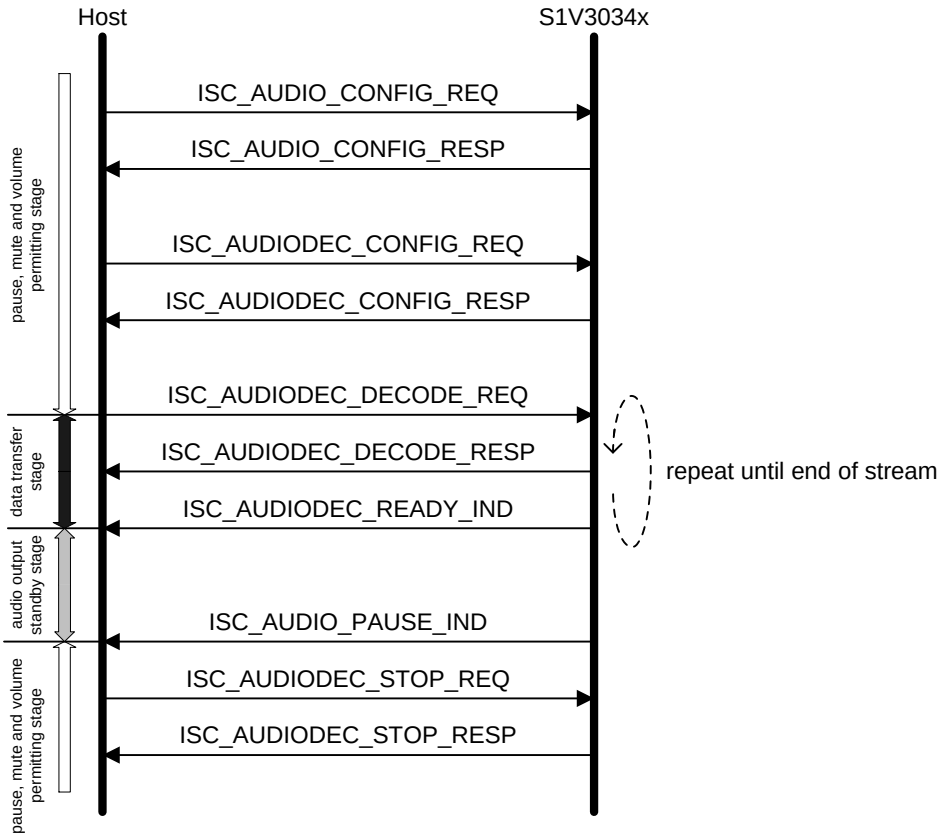


Figure 4.19 Streamed playback message flow

4.8.3 Message Description

4.8.3.1 ISC_AUDIODEC_CONFIG_REQ

Table 4.25 ISC_AUDIODEC_CONFIG_REQ

Direction	Host to S1V3034x	
Purpose	Used to setup the audio decoder.	
Byte	**Field**	**Values**
0	length (lsb)	0x0010 is set.
1	length (msb)	
2	msg_id (lsb)	0x006B - ISC_AUDIODEC_CONFIG_REQ
3	msg_id (msb)	
4	reserved	0x00
5	file_type	0x09: EOV
6	reserved	0x00
7	reserved	0x00
8	sampling_rate	Sampling frequency Example 16kHz: 0x3E80
9		
10		
11		
12	reserved	0x00
13	reserved	0x00
14	reserved	0x00
15	reserved	0x00

4.8.3.2 ISC_AUDIODEC_CONFIG_RESP

Table 4.26 ISC_AUDIODEC_CONFIG_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_AUDIODEC_CONFIG_REQ has been received. An error code is appended and is transmitted with the response if the REQ content is invalid.	
Byte	**Field**	**Values**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x006C -ISC_AUDIODEC_CONFIG_RESP
3	msg_id (msb)	
4	config_success (lsb)	0x0000: No Error >0x0000: Error Code
5	config_success (msb)	

4.8.3.3 ISC_AUDIODEC_DECODE_REQ

Table 4.27 ISC_AUDIODEC_DECODE_REQ

Direction	Host to S1V3034x	
Purpose	Used to transmit the audio data.	
Byte	**Field**	**Value**
0	length (lsb)	VARIABLE *Note 1
1	length (msb)	
2	msg_id (lsb)	0x006D – ISC_AUDIODEC_DECODE_REQ
3	msg_id (msb)	
4	reserved	0x00
5	reserved	0x00
6	reserved	0x00
7	reserved	0x00
	data *Note 1	Audio data

Note 1: Audio data size appended to ISC_AUDIODEC_DECODE_REQ is either 512, 1024, 2048. However, the last block of audio data transmitted may be of a size less than the blocks that were used during the rest of audio playback.

4.8.3.4 ISC_AUDIODEC_DECODE_RESP

Table 4.28 ISC_AUDIODEC_DECODE_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_AUDIODEC_DECODE_REQ has been received. An error code is appended if there is any invalid content in the REQ.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x006E -ISC_AUDIODEC_DECODE_RESP
3	msg_id (msb)	
4	decode_success (lsb)	0x0000: No error >0x0000: Error code
5	decode_success (msb)	

4.8.3.5 ISC_AUDIODEC_READY_IND

Table 4.29 ISC_AUDIODEC_READY_IND

Direction	S1V3034x to host	
Purpose	Used to notify the host that the S1V3034x is ready to receive further audio data.	
Byte	**Field**	**Value**
0	length (lsb)	0x0011
1	length (msb)	
2	msg_id (lsb)	0x006F - ISC_AUDIODEC_READY_IND
3	msg_id (msb)	
4-16	reserved	0x00

4.8.3.6 ISC_AUDIODEC_PAUSE_REQ

Table 4.30 ISC_AUDIODEC_PAUSE_REQ

Direction	Host to S1V3034x	
Purpose	Used to pause the audio output for streamed playback.	
Byte	**Field**	**Value**
0	length (lsb)	0x0008
1	length (msb)	
2	msg_id (lsb)	0x0070 - ISC_AUDIODEC_PAUSE_REQ
3	msg_id (msb)	
4	pause_enable (lsb)	0x0000: Cancel pause 0x0001: Enable pause
5	pause_enable (msb)	
6	reserved	0x00
7	reserved	0x00

4.8.3.7 ISC_AUDIODEC_PAUSE_RESP

Table 4.31 ISC_AUDIODEC_PAUSE_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_AUDIODEC_PAUSE_REQ has been received. An error code is appended if there is any invalid content in the REQ.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0071 - ISC_AUDIODEC_PAUSE_RESP
3	msg_id (msb)	
4	pause_success	0x0000: No error >0x0000: Error code
5	pause_success	

4.8.3.8 ISC_AUDIODEC_STOP_REQ

Table 4.32 ISC_AUDIODEC_STOP_REQ

Direction	Host to S1V3034x	
Purpose	Used to immediately end streamed playback, and to clear the contents of the audio data buffer.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x0072 - ISC_AUDIODEC_STOP_REQ
3	msg_id (msb)	
4	reserved	0x00
5	reserved	0x00

4.8.3.9 ISC_AUDIODEC_STOP_RESP

Table 4.33 ISC_AUDIODEC_STOP_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_AUDIODEC_STOP_REQ has been received.	
Byte	**Field**	**Value**
0	length (lsb)	0x0014
1	length (msb)	
2	msg_id (lsb)	0x0073 - ISC_AUDIODEC_STOP_RESP
3	msg_id (msb)	
4	stop_success (lsb)	0x0000: No error >0x0000: Error code
5	stop_success (msb)	
6-19	reserved	0x00

4.8.3.10 ISC_AUDIODEC_ERROR_IND

Table 4.34 ISC_AUDIODEC_ERROR_IND

Direction	S1V3034x to host	
Purpose	Used to notify the host of a non-fatal error in streamed playback.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x007B - ISC_AUDIODEC_ERROR_IND
3	msg_id (msb)	
5	error_code (lsb)	Error code
6	error_code (msb)	

4.9 Sequenced Playback Messages (only Supported by Products with Built-in ROM)

4.9.1 Overview

Sequenced playback is supported only by product versions containing built-in ROM.

- Configure the order (sequence) of audio data playback and set the silence duration between audio files.
- Sequenced playback
- Pause and stop of sequenced playback

The audio data is obtained from the customer beforehand and is stored in the built-in ROM of S1V3034x.

The host selects audio files from this ROM, and directs the order in which these are played back by S1V3034x.

After the host directs S1V3034x to start playback, S1V3034x continuously automatically plays the specified audio.This playback method is termed **sequenced playback**.

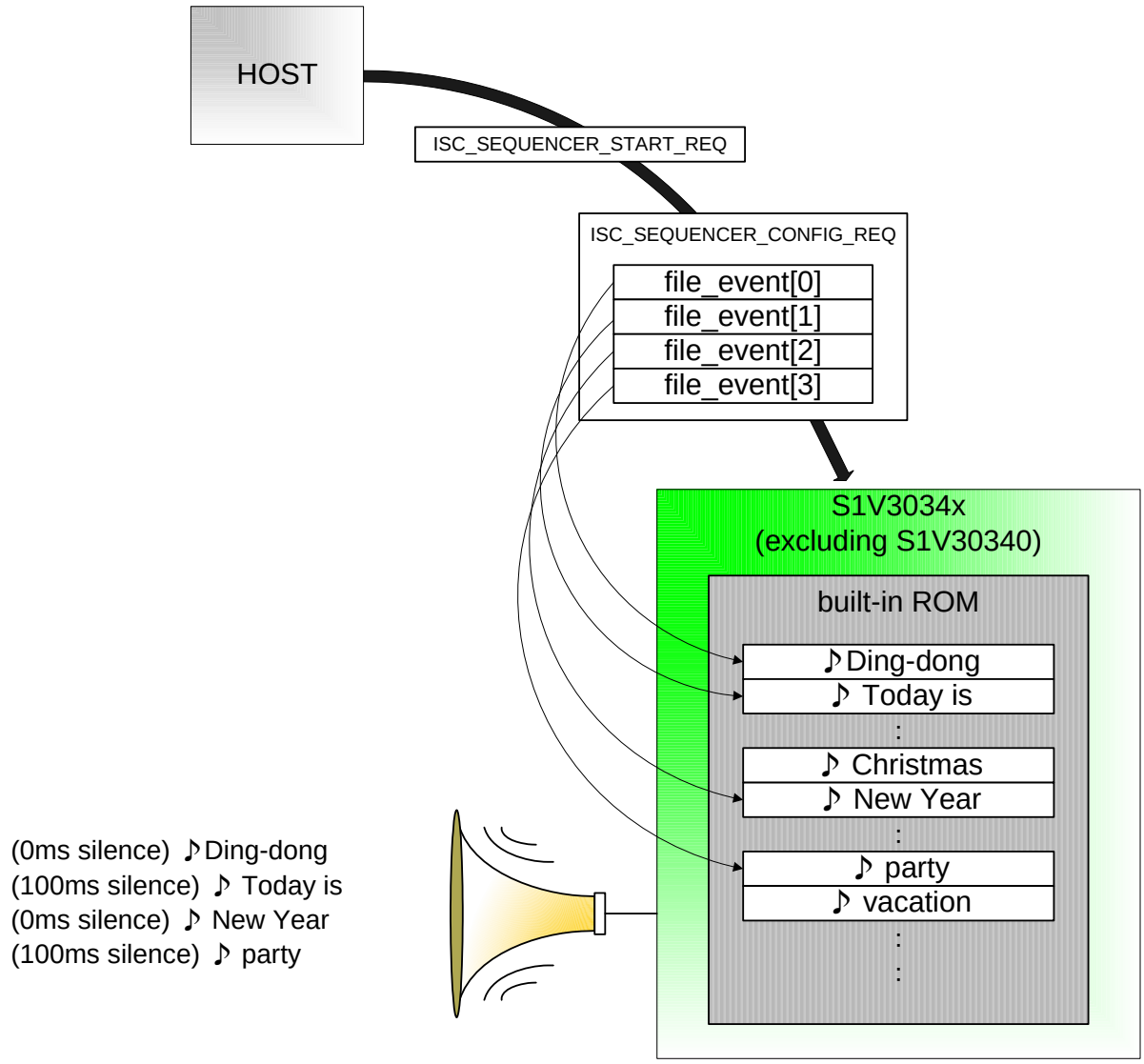


Figure 4.20 Image of sequence playing

4.9.2 Message Flow

4.9.2.1 Sequenced Playback Setup

ISC_SEQUENCER_CONFIG_REQ/RESP is used for the mandatory setup of sequenced playback. The items set-up are as follows.

- Number of times the full sequence is played back (it is possible to playback the sequence repeatedly).
- Number of audio files sequenced
- Phrase
- Audio files to playback in sequence
- Duration of silence interval before playing each audio file
- The playback order

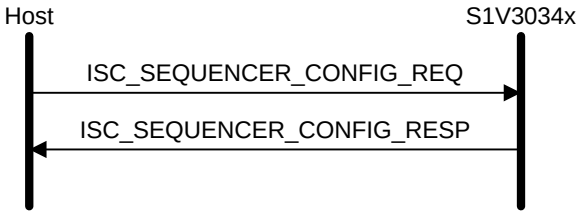


Figure 4.21 Sequence setting message flow

4.9.2.2 Sequencer START

The ISC_SEQUENCER_START_REQ/RESP message is used to start playback of the configured sequence.

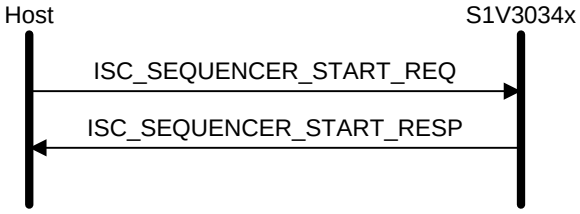


Figure 4.22 Sequence START message flow

4.9.2.3 Sequencer PAUSE

The ISC_SEQUENCER_PAUSE_REQ/RESP message is used to pause and un-pause the currently active decode sequence. When audio ouput stops, S1V3034x transmits ISC_AUDIO_PAUSE_IND.

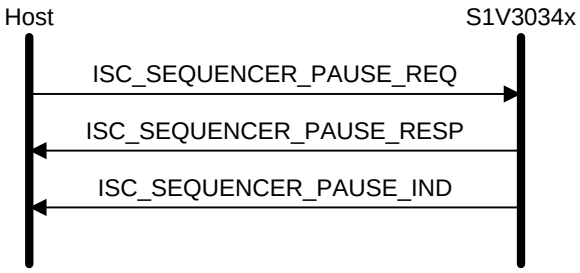


Figure 4.23 Sequencer PAUSE message flow

4.9.2.4 Sequenced Playback Pause Release

The host can restart audio output for paused S1V3034x streaming playback by canceling the pause setting and using ISC_AUDIODEC_PAUSE_REQ/RESP.

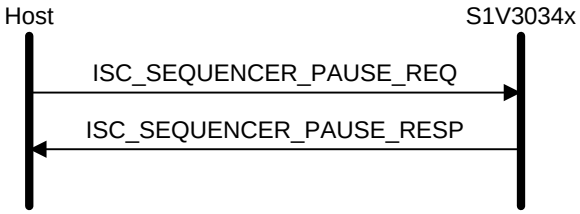


Figure 4.24 Sequenced playback PAUSE release message flow

4.9.2.5 Sequencer FINALISE IMMEDIATELY

The ISC_SEQUENCER_STOP_REQ/RESP message is used to immediately stop and finalise the currently active decode sequence. The S1V3034x immediately ends sequence playback if ISC_SEQUENCER_STOP_REQ is received twice. **ISC_SEQUENCER_STOP_REQ/RESP must be sent and received twice in order to immediately end sequence playback.**

At this time ISC_AUDIO_PAUSE_IND or ISC_SEQUENCER_STATUS_IND is not transmitted. However, ISC_SQUENCER_STATUS_IND may be transmitted after the RESP if S1V3034x receives the REQ at the same instant it is preparing to send this messsage. Please refer to “5.2.2.2 Concurrence of Messages Other than ISC_RESET_RESPs and IND” for details.

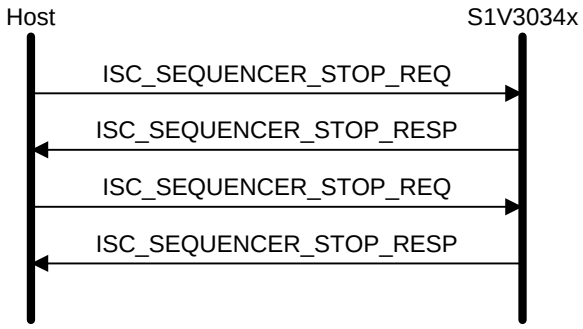


Figure 4.25 Immediate ending of sequenced playback message flow

4.9.2.6 Sequencer FINALISE

The S1V3034x can detect the end of audio playback automatically. When playback of an entire sequence has ended, ISC_SEQUENCER_STATUS_IND is sent to the host to indicate sequence playback has ended. The host can terminate sequenced playback using ISC_SEQUENCER_STOP_REQ/RESP. In this case, there is no problem if ISC_SEQUENCER_STOP_REQ/RESP is sent and received twice, in the same way as for immediate ending of sequence playback.

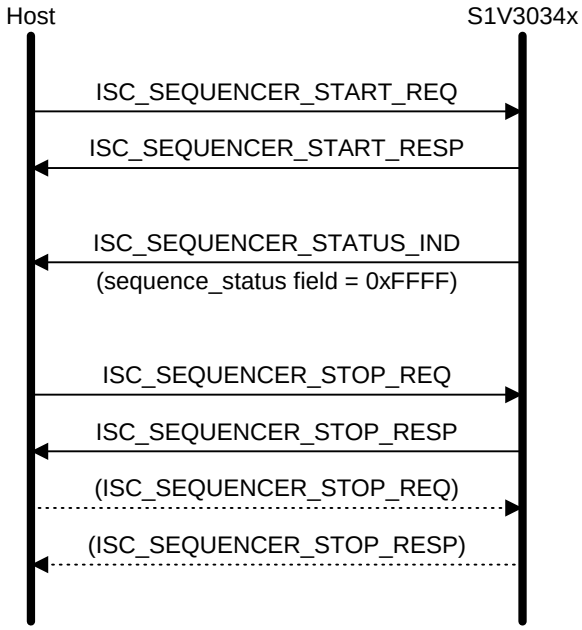


Figure 4.26 Sequence playback end message flow

4.9.2.7 Sequencer STATUS IND

The ISC_SEQUENCER_STATUS_IND message is sent by S1V3034x when the sequence is finished, and optionally (as specified in the ISC_SEQUENCER_START_REQ) when the decode of each individual sequenced file is complete.

Figure 4.27 Sequencer STATUS IND message flow

4.9.2.8 Sequenced Playback Message Flow

The message flow for sequenced playback is shown in Figure 4.28.

If not already configured, audio output must be configured before sequenced playback begins. Then use the ISC_SEQUENCER_CONFIG_REQ/RESP to setup the audio data playback sequence and the duration of the silent intervals inserted before each sequenced audio file. The maximum number of audio files phrases can be played back at a time is 64.

The silence duration is limited to the range 0 or 20-2047ms which may be specified in 1ms steps. The number of times that the playback sequence is repeated can be set in the play_count field. The range of this is limited to 0x0001-0xFFFE with infinite repetition possible through the setting of this value to 0xFFFF. Note that repeat playback is not supported when multiple phrases are specified. *Note 1

Sequenced playback may then be started using ISC_SEQUENCER_START_REQ/RESP.

At this time, it is possible to select whether S1V3034x will notify the host of the end of the decoding of each sequenced file.

When notification of the end of each sequenced file is enabled, S1V3034x transmits ISC_SEQUENCER_STATUS_IND when decode of each audio file ends and a value that increments every time the end of a sequenced file reached is appended to its sequencer_status_field. The value of this field at the end of the decoding of the first audio file is 0. **When the last audio file of the sequence has been output, the value of sequencer_status field is 0xFFFF**. The ISC_SEQUENCER_STATUS_IND transmitted when the last audio output is completed is transmitted regardless of the selection of the notification option. The host finalises sequenced playback by using ISC_SEQUENCER_STOP_REQ/RESP. The subsequent sequenced playback cannot begin until sequenced playback has been finalised. *Note 2

If the REQ-RESP protocol is adhered to; cancel, pause, mute and volume setting may be requested at any time.

If playback is halted by sending and receiving ISC_SEQUENCER_STOP_REQ/RESP twice during infinite repeat with the play_count field at 0xFFFF, the system should be reset using ISC_RESET_REQ before using sequence playback with new settings.

The message flow of the 3 audio scenarios specified is shown in Figure 4.28.

The transmission of messages related to streamed playback during the sequenced playback period is prohibited.

The sequenced playback period starts from the host transmiting ISC_SEQUENCER_CONFIG_REQ and ends with the completion of the transmission of the ISC_SEQUENCER_STATUS_IND (sequence end notification) by the S1V3034x.

Note 1: The number of phrases can be set to 1 by Seiko Epson if repeat playback is required for multiple phrases. It may not be possible to use 1 for certain phrase formats. Please consult for details.

Note 2: When host sets enable_status_ind field to 0x0001 at ISC_SEQ_START_REQ, host should receive ISC_SEQ_STATUS_IND before completion of the next EOV file's playback.

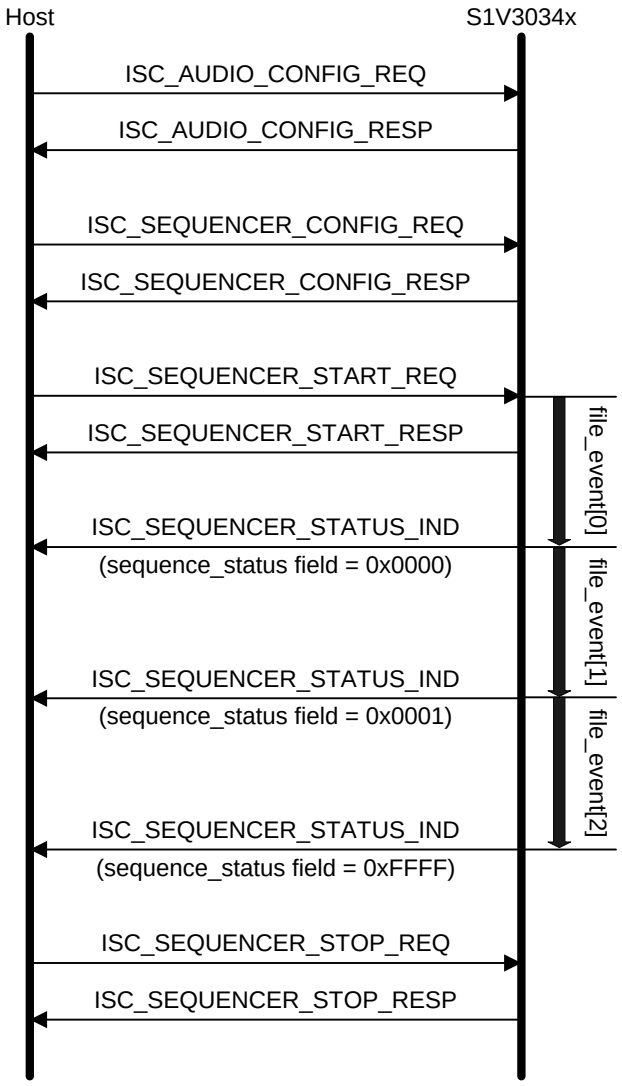


Figure 4.28 Sequenced playback message flow (enable_status_ind = 0x0001)

4.9.3 Message Descriptions

4.9.3.1 ISC_SEQUENCER_CONFIG_REQ

Table 4.35 ISC_SEQUENCER_CONFIG_REQ

Direction	Host to S1V3034x	
Purpose	Used to set the sequence to be played back.	
Byte	**Field**	**Value**
0	length (lsb)	8 + number of phrases × 8
1	length (msb)	
2	msg_id (lsb)	0x00C4 - ISC_SEQUENCER_CONFIG_REQ
3	msg_id (msb)	
4	play_count (lsb)	0xFFFF: Infinite sequence playback 0x0001-0xFFFE: Number of times sequence is played back. *Note 1
5	play_count (msb)	
6	num_files (lsb)	Number of sequence playback files 0x01-0x40
7	num_files (msb)	
8..15	file_event[0]	file_event 0
16..23	file_event[1]	file_event 1
…	…	…
512…519	file_event[63]	file_event 63

File_event structure(Use it with ISC_SEQUENCER_CONFIG_REQ.)

0	delay_ms (lsb)	Delay time inserted before playback of sequence file (inserted after playback of previous sequenced file is complete). 0x0000: No delay 0x0014-0x07FF: 20 – 2047ms *Note 2
1	delay_ms (msb)	
2	descramble (lsb)	Reserved
3	descramble (msb)	
4	file_type (lsb)	0x0000: Reserved 0x0001: Reserved 0x0002: Reserved 0x0003: EOV
5	file_type (msb)	
6	filename(lsb)	sequence file *Note 3
7	filename(msb)	

Note 1: The play_count field must be set to 1 if the num_files field is specified as 2 or more. Repeat playback is not supported when multiple phrases are specified. The number of phrases can be set to 1 by Seiko Epson if repeat playback is required for multiple phrases. It may not be possible to use 1 for certain phrase formats. Please consult for details.

Note 2: The last 13 bits will be set as a mute interval if a value larger than 0x07FFF is set in the delay_ms field.

Note 3: For files named by their number, the order in which these are stored in ROM corresponds to the file numbering.

4.9.3.2 ISC_SEQUENCER_CONFIG_RESP

Table 4.36 ISC_SEQUENCER_CONFIG_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_SEQUENCER_CONFIG_REQ has been received. An error code is appended if invalid content was detected in the REQ.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x00C5 –ISC_SEQUENCER_CONFIG_RESP
3	msg_id (msb)	
4	config_success (lsb)	0x0000: No error >0x0000: Error
5	config_success (msb)	

4.9.3.3 ISC_SEQUENCER_START_REQ

Table 4.37 ISC_SEQUENCER_START_REQ

Direction	Host to S1V3034x	
Purpose	Start the sequence that was specified by the ISC_SEQUENCER_CONFIG_REQ.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x00C6 – ISC_SEQUENCER_START_REQ
3	msg_id (msb)	
4	enable_status_ind (lsb)	0x0000: ISC_SEQUENCER_STATUS_IND is not sent. 0x0001: ISC_SEQUENCER_STATUS_IND is sent. *Note 1, 2
5	enable_status_ind (msb)	

Note 1: The last ISC_SEQUENCER_STATUS_IND message is sent, whenever the sequence ends regardless of the enable_status_ind setting.

Note 2: When host sets enable_status_ind field to 0x0001, host should receive ISC_SEQ_STATUS_IND before completion of the next EOV file's playback.

4.9.3.4 ISC_SEQUENCER_START_RESP

Table 4.38 ISC_SEQUENCER_START_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_SEQUENCER_START_REQ has been received.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x00C7 – ISC_SEQUENCER_START_RESP
3	msg_id (msb)	
4	start_success (lsb)	0x0000: No error >0x0000: Error
5	start_success (msb)	

4.9.3.5 ISC_SEQUENCER_STOP_REQ

Table 4.39 ISC_SEQUENCER_STOP_REQ

Direction	Host to S1V3034x	
Purpose	Used to immediately end sequenced playback.	
Byte	**Field**	**Value**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0x00C8 – ISC_SEQUENCER_STOP_REQ
3	msg_id (msb)	

4.9.3.6 ISC_SEQUENCER_STOP_RESP

Table 4.40 ISC_SEQUENCER_STOP_RESP

Direction	Device to host	
Purpose	Used to notify the host that ISC_SEQUENCER_STOP_REQ has been received.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x00C9 – ISC_SEQUENCER_STOP_RESP
3	msg_id (msb)	
4	stop_success (lsb)	0x0000: No error >0x0000: Error
5	stop_success (msb)	

4.9.3.7 ISC_SEQUENCER_PAUSE_REQ

Table 4.41 ISC_SEQUENCER_PAUSE_REQ

Direction	S1V3034x to host	
Purpose	Used to pause the audio output during sequenced playback.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x00CA – ISC_SEQUENCER_PAUSE_REQ
3	msg_id (msb)	
4	enable_pause (lsb)	0x0000: Disable pause 0x0001: Enable pause
5	enable_pause (msb)	

4.9.3.8 ISC_SEQUENCER_PAUSE_RESP

Table 4.42 ISC_SEQUENCER_PAUSE_RESP

Direction	S1V3034x to host	
Purpose	Used to notify the host that the ISC_SEQUENCER_PAUSE_REQ has been received. An error code is appended if there is any invalid content in the REQ.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x00CB – ISC_SEQUENCER_PAUSE_RESP
3	msg_id (msb)	
4	pause_success (lsb)	0x0000: No error >0x0000: Error
5	pause_success (msb)	

4.9.3.9 ISC_SEQUENCER_STATUS_IND

Table 4.43 ISC_SEQUENCER_STATUS_IND

Direction	S1V3034x to host	
Purpose	Used to notify the host of the completion of the decode of each file in the sequence or of the completion of the sequenced playback.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x00CC – ISC_SEQUENCER_STATUS_IND
3	msg_id (msb)	
4	sequence_status (lsb)	0xFFFF: End of sequence. >0x0000: File index *Note 1 (Range from 0 to 62 specified by ISC_SEQUENCER_CONFIG_REQ, ending with 0xFFFF. Note that range is from 0 to 63 if play_count is 2 or more.)
5	sequence_status (msb)	

Note 1: The file index indicates that processing is complete for the file sequence set in the sequencer output.

4.9.3.10 ISC_SEQUENCER_ERROR_IND

Table 4.44 ISC_SEQUENCER_ERROR_IND

Direction	S1V3034x to host	
Purpose	Used to notify the host of a non-fatal error in sequenced playback.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0x00CD – ISC_SEQUENCER_ERROR_IND
3	msg_id (msb)	
4	error_code (lsb)	Error code
5	error_code (msb)	

4.10 General-purpose Output Control Message (S1V3S344 and S1V3G340 only)

4.10.1 Overview

The S1V3S344 features nine general-purpose output ports, while the S1V3G340 features seven. These ports can be controlled by the host using ISC_GPOSW_IND.

4.10.2 Message Flow

4.10.2.1 General-purpose Output Port Control

The host can control general-purpose output ports by sending ISC_GPOSW_IND to the S1V3S344 or S1V3G340. The S1V3S344 or S1V3G340 changes the level of the general-purpose output ports as specified in the data received when ISC_GPOSW_IND is received.

Figure 4.29 General-purpose output port control message flow

4.10.3 Message Descriptions

4.10.3.1 ISC_GPOSW_IND

Table 4.45 ISC_GPOSW_IND

Direction	Host to S1V3S344 or S1V3G340	
Purpose	Used to control general-purpose output ports.	
Byte	**Field**	**Value**
0	length (lsb)	0x0006
1	length (msb)	
2	msg_id (lsb)	0xFF01 -ISC_GPOSW_IND
3	msg_id (msb)	
4	gpo_set (lsb)	Setting bit assigned to pins to 0 outputs L, while setting to 1 outputs H. [0]: GP00 (initial value 0) [1]: GP01 (initial value 0) [2]: GP02 (initial value 0) [3]: GP03 (initial value 0) [4]: GP04 (initial value 0) [5]: GP05 (initial value 0) [6]: GP06 (initial value 0) [7]: GP07 (initial value 0) [8]: GP08 (initial value 0)*Note 1 [15:9]: reserved
5	gpo_set (msb)	

Note 1: GP07 and GP08 are not present in S1V3G340.

4.11 Flash Access Message (S1V3S344 and S1V3G340 only)

4.11.1 Overview

The S1V3S344 and S1V3G340 enable the host to access flash memory directly via a serial interface. This is known as flash access mode. The host can set the S1V3S344 or S1V3G340 to flash access mode using ISC_SPIDW_IND.

Flash access mode is used to write voice data to flash memory. For information on how to write voice data, refer to the *Flash Access Specification*.

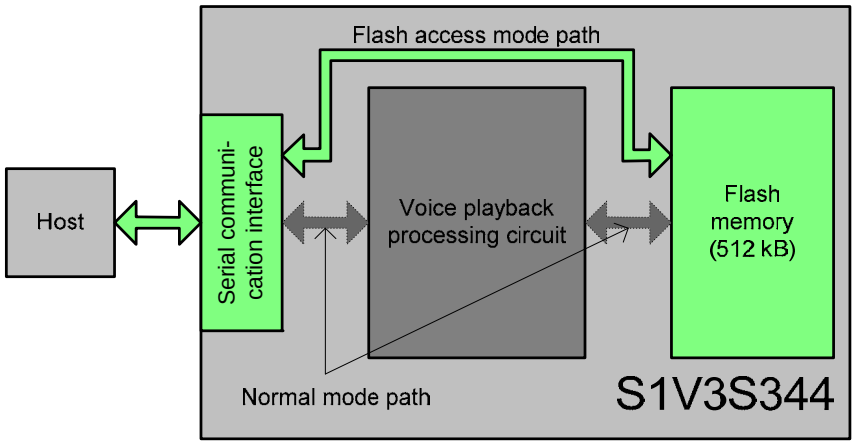


Figure 4.30 S1V3S344 flash access mode image

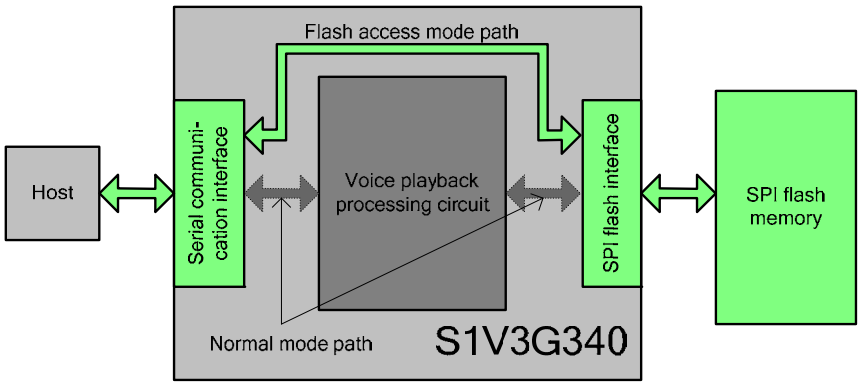


Figure 4.31 S1V3G340 flash access mode image

4.11.2 Message Flow

4.11.2.1 Flash Access Mode Setting

The host can set the device to flash access mode by sending ISC_SPISW_IND to the S1V3S344 or S1V3G340. On receiving ISC_SPISW_IND, the S1V3S344 or S1V3G340 switches from normal mode to flash access mode, and the flash memory is connected to the serial communication interface.

Resetting from Flash access mode to normal mode requires a hardware reset using NRESET.

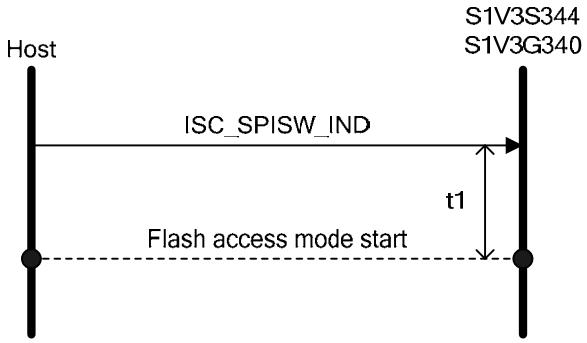


Figure 4.32 Flash access mode message flow

Sign	Parameter	Min	Max	Unit
t1	Time from receipt of ISC_SPISW_IND by S1V3S344 or S1V3G340 to start of flash access mode.	-	5.0	µs

4.11.3 Message Descriptions

4.11.3.1 ISC_SPISW_IND

Table 4.46 ISC_SPISW_IND

Direction	Host to S1V3S344 or S1V3G340	
Purpose	Used to set S1V3S344 or S1V3G340 to flash access mode. *Note 1	
Byte	**Field**	**Value**
0	length (lsb)	0x0004
1	length (msb)	
2	msg_id (lsb)	0xFF00 -ISC_SPISW_IND
3	msg_id (msb)	

Note 1: Once set to flash access mode, all signals input to the serial communication interface are transmitted directly to flash memory. For information on how to write, read, and erase data to or from flash memory, refer to the *Flash Access Specification*.

5. Error Processing

5.1 Return from Error

When an error occurs, the action taken depends upon whether the error is fatal or non-fatal. If the error is fatal then the host is notified through the transmission of a ISC_ERROR_IND. If the error is non-fatal, streamed playback errors are notified through the transmission of ISC_AUDIODEC_ERROR_IND and sequenced playback errors are notified through the transmission of ISC_SEQUENCER_ERROR_IND. This section explains the recovery procedure to be used if an error occurs.

5.1.1 Return from Fatal Error

S1V3034x can notify the host of the occurrence of a fatal error through the transmission of a ISC_ERROR_IND message at any time. The error code in this message indicates the cause of the error. If an ISC_ERROR_IND is received by the host, indicating a fatal error, the S1V3034x must be reset in order to recover. There are two methods for resetting the system; hardware reset using the NRESET pin; and software reset using ISC_RESET_REQ/RESP. For information on the range initialized by this reset, see "4.4.3.1 ISC_RESET_REQ."

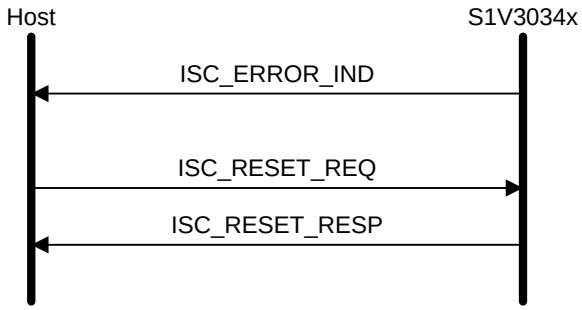


Figure 5.1 Return message flow from fatal error

5.1.2 Return from Non-fatal Error during Streamed Playback

S1V3034x can notify the host of a non-fatal error in streamed playback at any time by transmitting a ISC_AUDIODEC_ERROR_IND. The error code in this message shows the cause of the error. ISC_AUDIODEC_STOP_REQ/RESP must be sent and received twice in order to reset following an error.

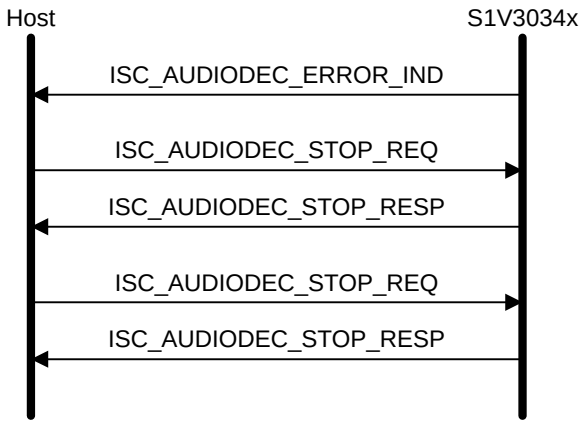


Figure 5.2 Return message flow from non-fatal error during streamed playback

5.1.3 Return from Non-Fatal Error during Sequenced Playback

S1V3034x can notify the host of a non-fatal error in sequenced playback at any time by transmitting a ISC_SEQUENCER_ERROR_IND. The error code in this message shows the cause of the error. ISC_SEQUENCER_STOP_REQ/RESP must be sent and received twice in order to reset following an error.

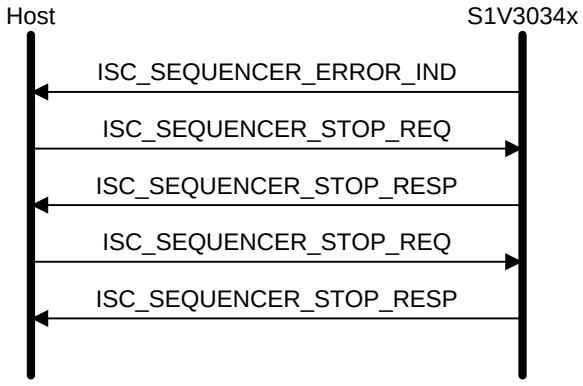


Figure 5.3 Return Message flow from non-fatal error during sequenced playback

5.1.4 Recovering from Other Errors

The S1V3034x transmits a RESP message indicating a non-fatal error code when a REQ message sent from the host contains a value not supported by the S1V3034x. If this error occurs, reset the system by transmitting ISC_RESET_REQ and receiving ISC_RESET_RESP.

The S1V3034x also transmits a RESP message indicating a non-fatal error code if its state is such that it cannot accept a REQ message. The system is reset in different ways, depending on the REQ message, as follows:

1) ISC_TEST_REQ/RESP (0x4004)

   This error occurs if the S1V3034x has already received ISC_TEST_REQ and receives ISC_TEST_REQ once again. The values for the fields are updated to the values last set by ISC_TEST_REQ. However, the key fields are fixed to each user. Make sure that the correct key code is entered.

2) ISC_AUDIO_CONFIG_REQ/RESP or ISC_AUDIO_VOLUME_REQ/RESP (0x4021)

   This error occurs if the S1V3034x volume is set beyond the permissible range. The system is reset by setting the desired volume using ISC_ AUDIO_CONFIG_REQ once again.

3) ISC_AUDIODEC_PAUSE_REQ/RESP (0x4063 or 0x4064)

   This error occurs if a pause is requested using ISC_AUDIODEC_PAUSE_REQ while streamed playback is paused (0x4063) or if a pause release is requested using ISC_AUDIODEC_PAUSE_REQ during streamed playback. A pause should be requested during playback, and a pause release should be requested during a pause.

4) ISC_SEQUENCER_PAUSE_REQ/RESP (0x4182)

   This error occurs if a pause is requested using ISC_ SEQUENCER_PAUSE_REQ while the sequenced playback is paused or if a pause release is requested using ISC_SEQUENCER_PAUSE_REQ during sequenced playback. A pause should be requested during playback, and a pause release should be requested during pause.

5) ISC_PMAN_STANDBY_ENTRY_REQ/RESP

   This error occurs if the S1V3034x cannot enter standby mode. Since the S1V3034x cannot enter standby mode during audio playback, transmit ISC_PMAN_STANDBY_ENTRY_REQ after audio playback has stopped completely.

5.1.5 Message Blocking

S1V3034x notifies the host of an invalid Request message by sending an ISC_MSG_BLOCKED_RESP in the place of the expected response and with the same timing.

The cause of the message protocol violation depends upon the usage scenario which may be any of the following:

1) The host has violated the audio playback protocol defined within this message protocol specification. The violation is observed in one of the following playback scenaroios:

   (1) An error has occurred during streamed playback. In this case the streamed playback message flow is replaced by the Substitute message blocked during streamed playback recovery flow (Refer to Figure 5.4).

   (2) An error has occurred during sequenced playback. In this case the sequenced playback message flow is replaced by the Substitute message blocked during sequenced playback recovery flow (Refer to Figure 5.5).

   (3) A sequence playback message has been transmitted during the streaming playing period. (Refer to Figure 5.6)

   (4) A streamed playback message has been transmitted during the sequence playing period. (Refer to Figure 5.7)

2) A fatal error has occurred and the host has transmitted messages other than a ISC_RESET_REQ before the S1V3034x has been reset. (Refer to Figure 5.8)

3) A non-fatal error has occurred and the host has transmitted messages other than a ISC_(*)_STOP_REQ or ISC_RESET_REQ before the error has been corrected. (Refer to Figure 5.9, Figure 5.10)

In the cases of 2) and 3) above, the error code field appended to ISC_MSG_BLOCKED_RESP contains the same error code as was notified by the ISC_ERROR_IND or ISC_(*)_ERROR_IND. However if another error occurs, the latest error is attached to the ISC_MSG_BLOCKED_RESP.

Note: In the above description (*) represents either AUDIODEC or SEQUENCER

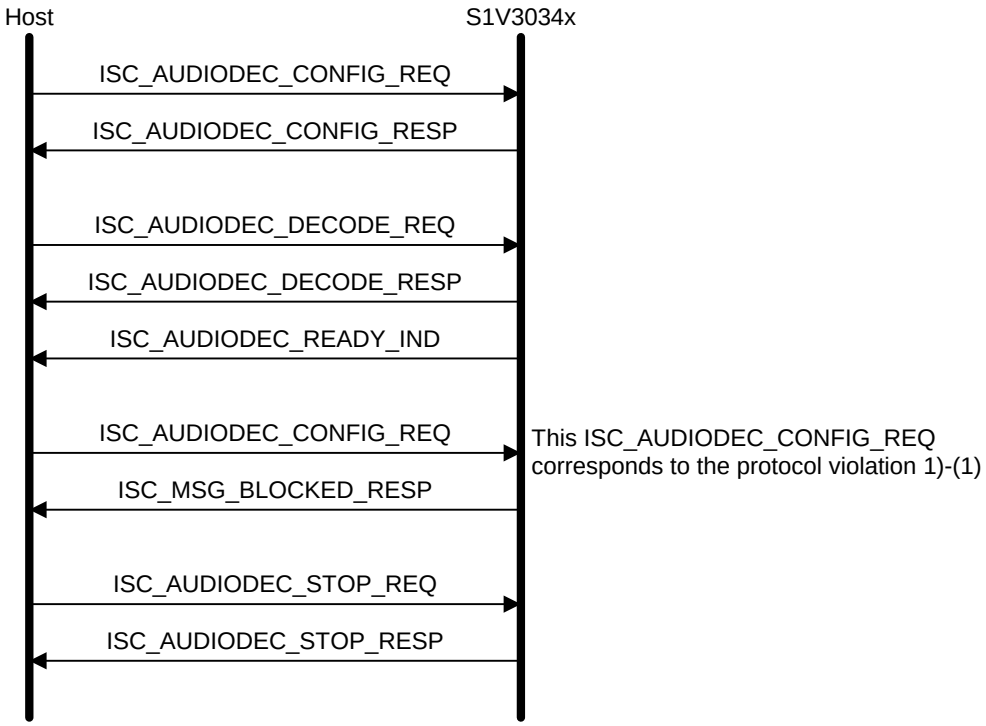


Figure 5.4 Substitute message blocked during streamed playback recovery flow

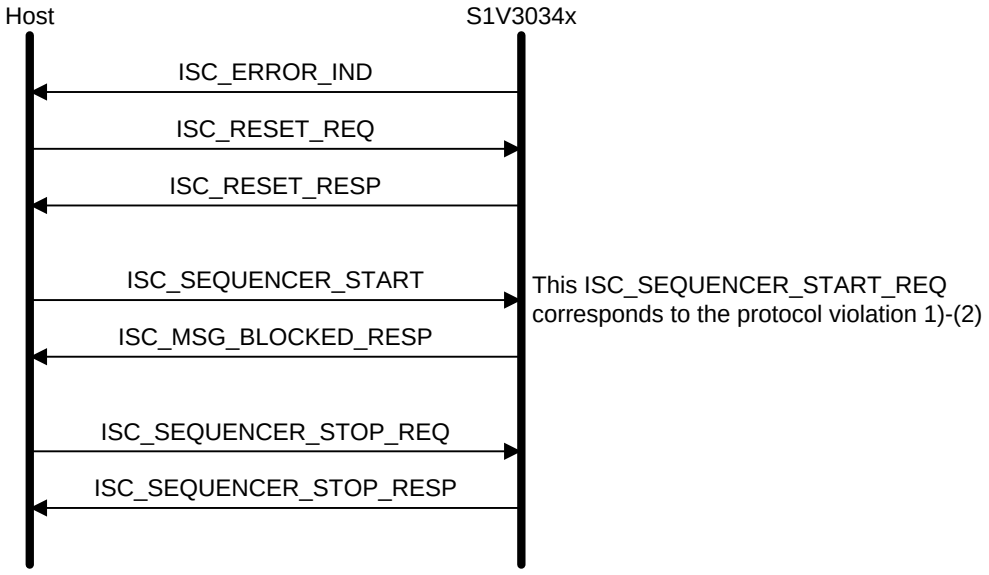


Figure 5.5 Substitute message blocked during sequenced playback recovery flow

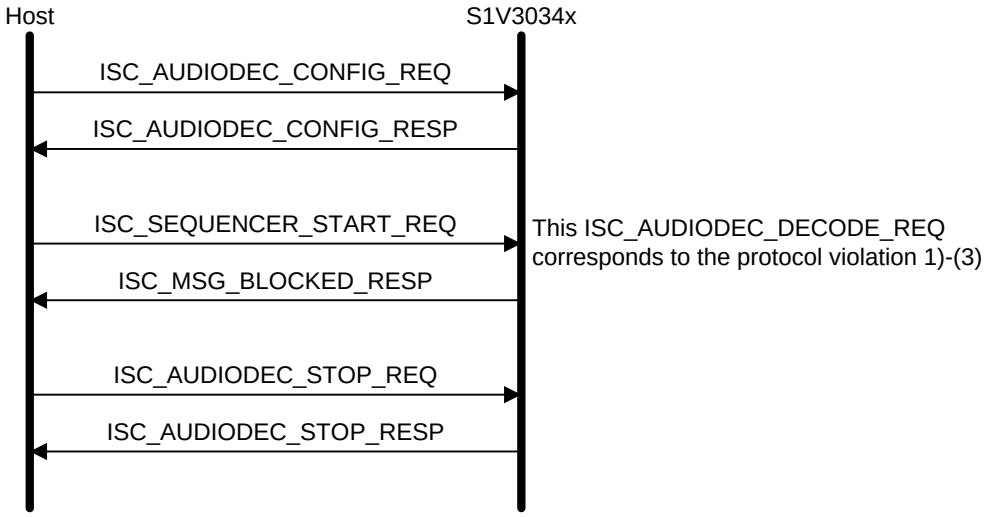


Figure 5.6 It blocks and it return flow by the transmission of the message related to the sequence while the streaming playing.

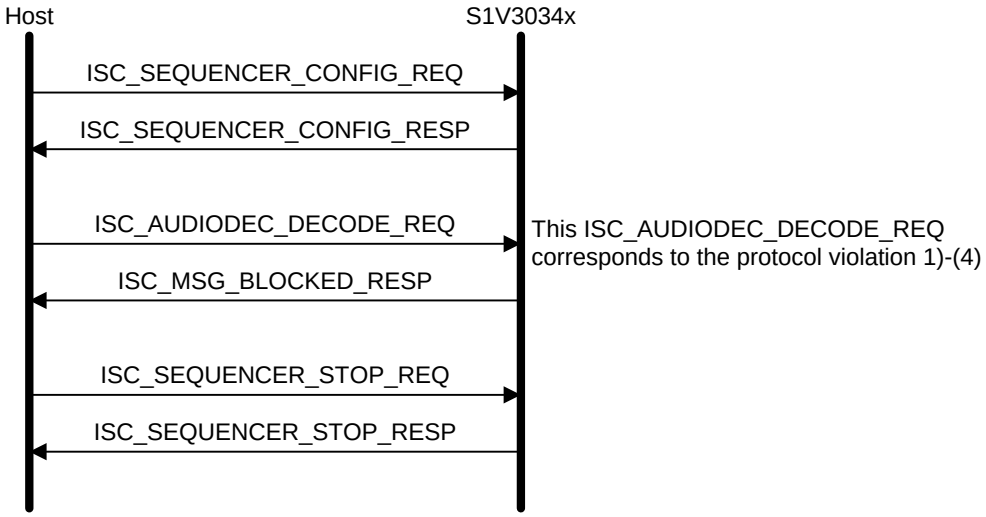


Figure 5.7 It blocks and return flow by the transmission of the message related to the streaming while the sequence playing.

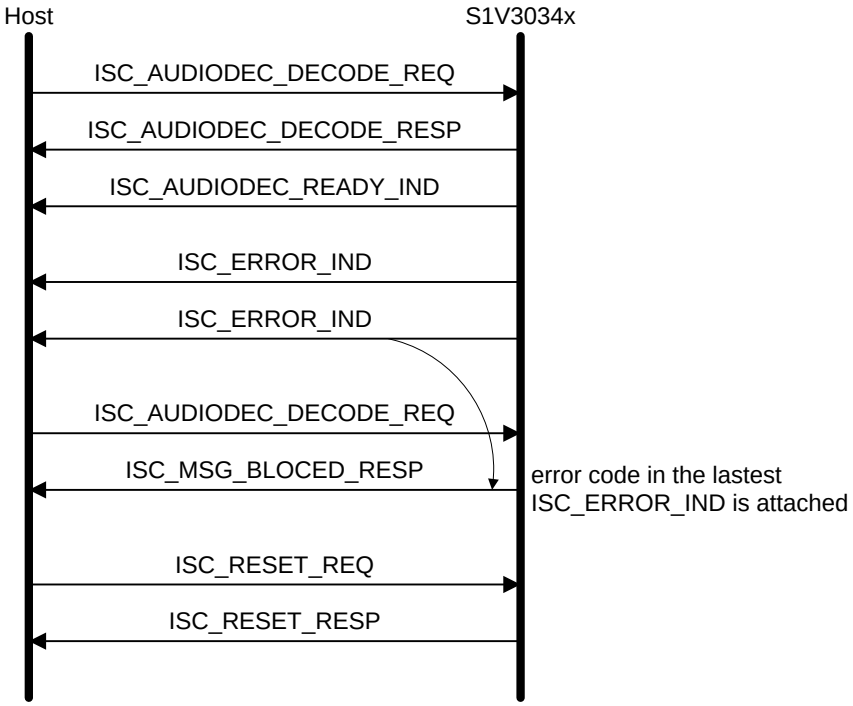


Figure 5.8 Block and return flow from fatal error

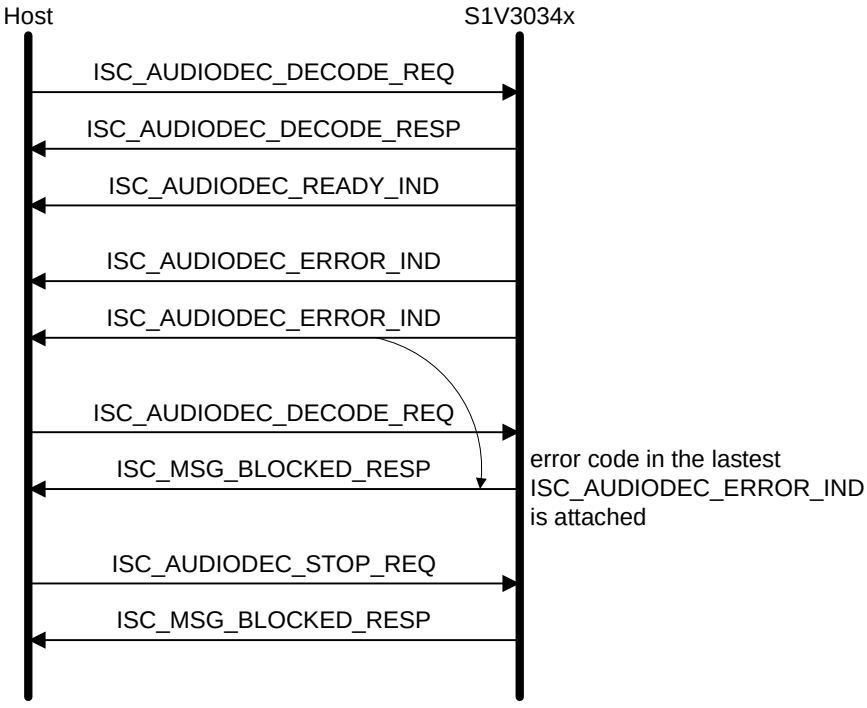


Figure 5.9 Block and return flow from non-fatal error while streaming playing

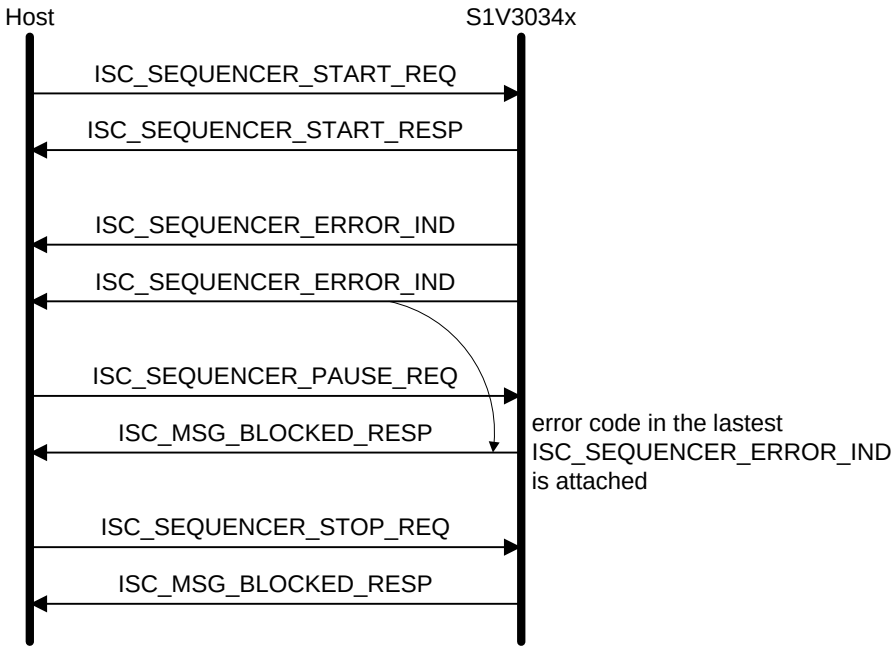


Figure 5.10 Block and return flow from non-fatal error while sequence playing

5.2 IND Message

The host REQ message and S1V3034x IND message may be sent at the same time when communicating with the S1V3034x using clock synchronous serial format, since the data lines send and receive messages independently. RESP and IND messages may also be generated at the same time, as the timing for sending IND messages by the S1V3034x is determined by internal status and is unrelated to the timing for REQ message receipt.

5.2.1 Reception of IND and RESP during Transmission of REQ

5.2.1.1 IND during Transmission of ISC_RESET_REQ

If transmission of an IND message begins while the host is sending ISC_RESET_REQ, the RESP message corresponding to the REQ message is sent with normal timing.

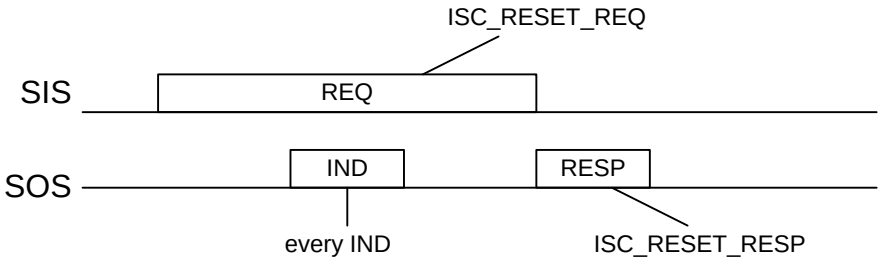


Figure 5.11 IND transmission during ISC_RESET_REQ

5.2.1.2 IND during Transmission of REQ Other than ISC_RESET_REQ

If transmission of any of the following IND messages begins while the host is transmitting any REQ message other than ISC_RESET_REQ, all RESP messages corresponding to the REQ messages are replaced by ISC_MSG_BLOCKED_RESP.

- ISC_ERROR_IND
- ISC_AUDIODEC_ERROR_IND
- ISC_SEQUENCER_ERROR_IND

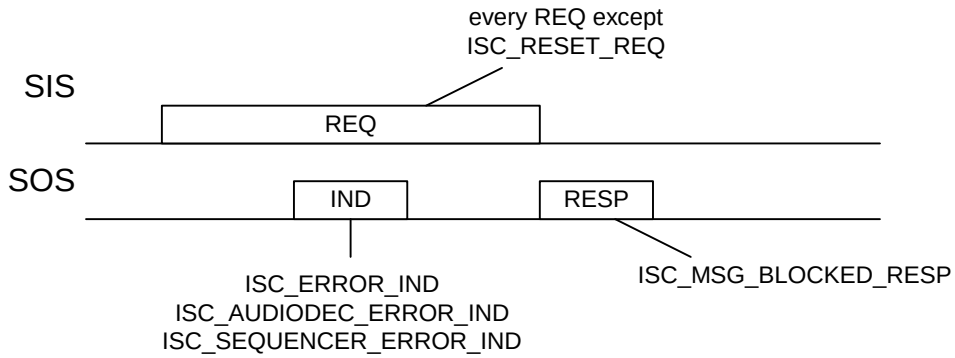


Figure 5.12 Error IND during REQ transmission

If transmission of an IND message other than those above begins under the same conditions, the RESP messages corresponding to the REQ messages are sent by the S1V3034x using normal timing.

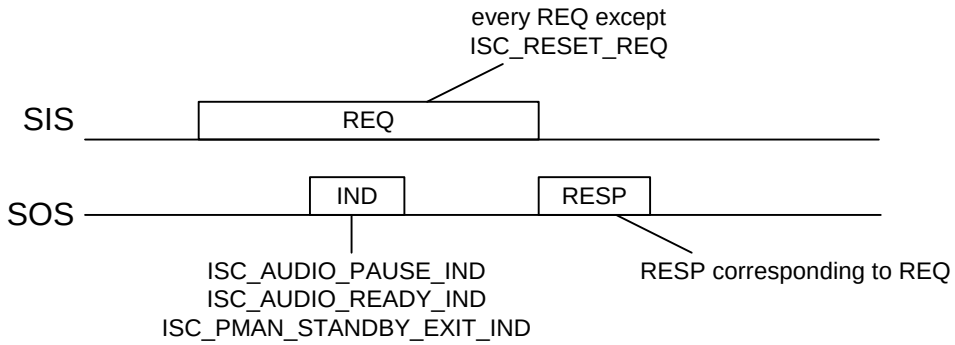


Figure 5.13 Non-error system IND transmitting REQ

5.2.2 Concurrence of the IND and the RESP

5.2.2.1 Concurrence of ISC_RESET_RESP and IND

When an event resulting in an IND message occurs at the same time as a RESET_RESP is ready to be sent, the IND is not sent due to the actioning of the software reset and the RESP is sent as usual.

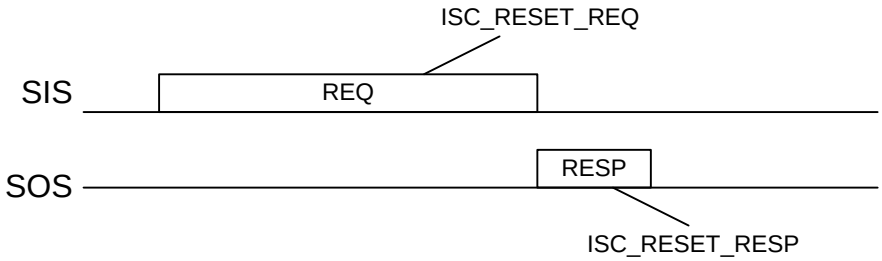


Figure 5.14 Concurrence of ISC_RESET_RESP and IND

5.2.2.2 Concurrence of Messages Other than ISC_RESET_RESPs and IND

When an event sending any of the following IND messages occurs at the same time as a RESP message (other than ISC_RESET_RESP), these IND messages are sent at the same timing usually used to send RESP messages in response to REQ messages. The RESP messages are sent in response to REQ messages once the IND message has been sent.

(1) ISC_ERROR_IND

(2) ISC_AUDIODEC_ERROR_IND

(3) ISC_SEQUENCER_ERROR_IND

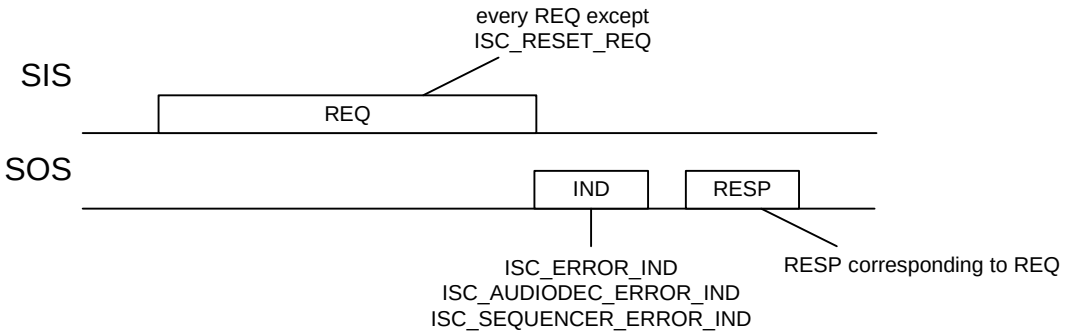


Figure 5.15 Concurrence of RESP and error IND messages

When an event sending an IND message other than those above occurs at the same time as a RESP message (other than ISC_RESET_RESP), the RESP messages in response to REQ messages are sent at the normal timing. IND messages are sent after the RESP messages have been sent.

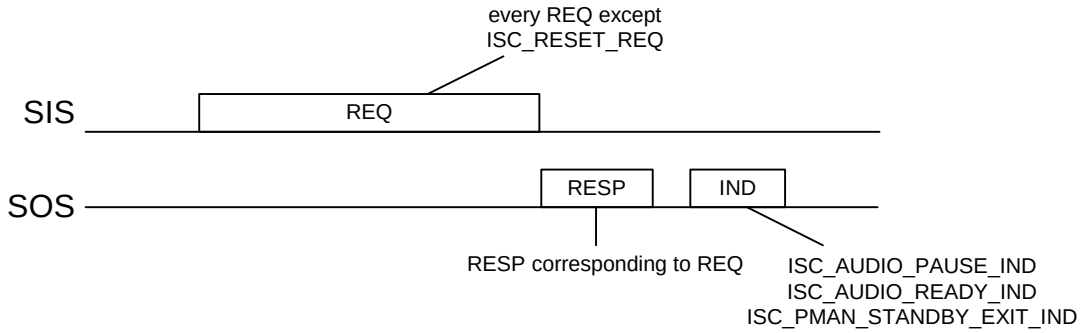


Figure 5.16 Concurrence of RESP and non-system error IND

6. Use Case of Message Protocol

No.	Category	Situation	Note	Refference
1	streaming	Audio playback of 3,712 bytes of data. The volume of data sent at a time is assumed to be 512 bytes.	• Flow from hardware reset to end of general streamed playback.	Figure 6.1
2	streaming	The case defined in no.1 is stopped immediately by using B. After it has been stopped the same audio is re-played from the beginning.	• Immediately end streamed playback • Restart of streamed playback after it has been stopped	Figure 6.2
3	streaming	Audio Playback of 5,511 byte. Mute and pause are requested during playback. The Volume of data sent at a time is assumed to be 512 bytes.	• The mute process in Streamed playback • The pause process in Streamed playback	Figure 6.3
4	streaming	Audio playback of 1,378 byte. The volume of data sent at a time is assumed to be 512 byte.	• The pause, mute and volume permitting state in streamed playback. • Audio output standby stage in streamed playback.	Figure 6.4
5	sequenced	Three audio data files are played back using sequenced playback.	• Flow from hardware reset to end of general sequenced playback. • ISC_SEQUENCER_STATUS_ IND when status IND is made effective.	Figure 6.5
6	sequenced	Similar scenario to no.4 except the status IND at the end of each file is disabled.	• ISC_SEQUENCER_STATUS_ IND when status IND at end of each file is disabled.	Figure 6.6
7	sequenced	One audio data file is played back using sequenced playback. The Play count is assumed to be 3. The status IND at the end of each file is enabled.	• ISC_SEQUENCER_STATUS_ IND when status IND at end of each file is disabled and play count is assumed to be 2.	Figure 6.7
8	sequenced	Three audio data files are played back using sequenced. Mute and pause are requested during streamed playback. Sequenced playback is ended stop at the end of playback.	• The pause process during sequenced playback. • An immediate stop of sequenced playback.	Figure 6.8
9	power management	After entering standby-mode from normal mode, it exits standby to enter normal mode again.	• Enter standby-mode from normal mode. • Exit standby-mode and enter normal mode.	Figure 6.9
10	Streaming playback (UART)	Executes same state as No. 1 using UART.		Figure 6.10

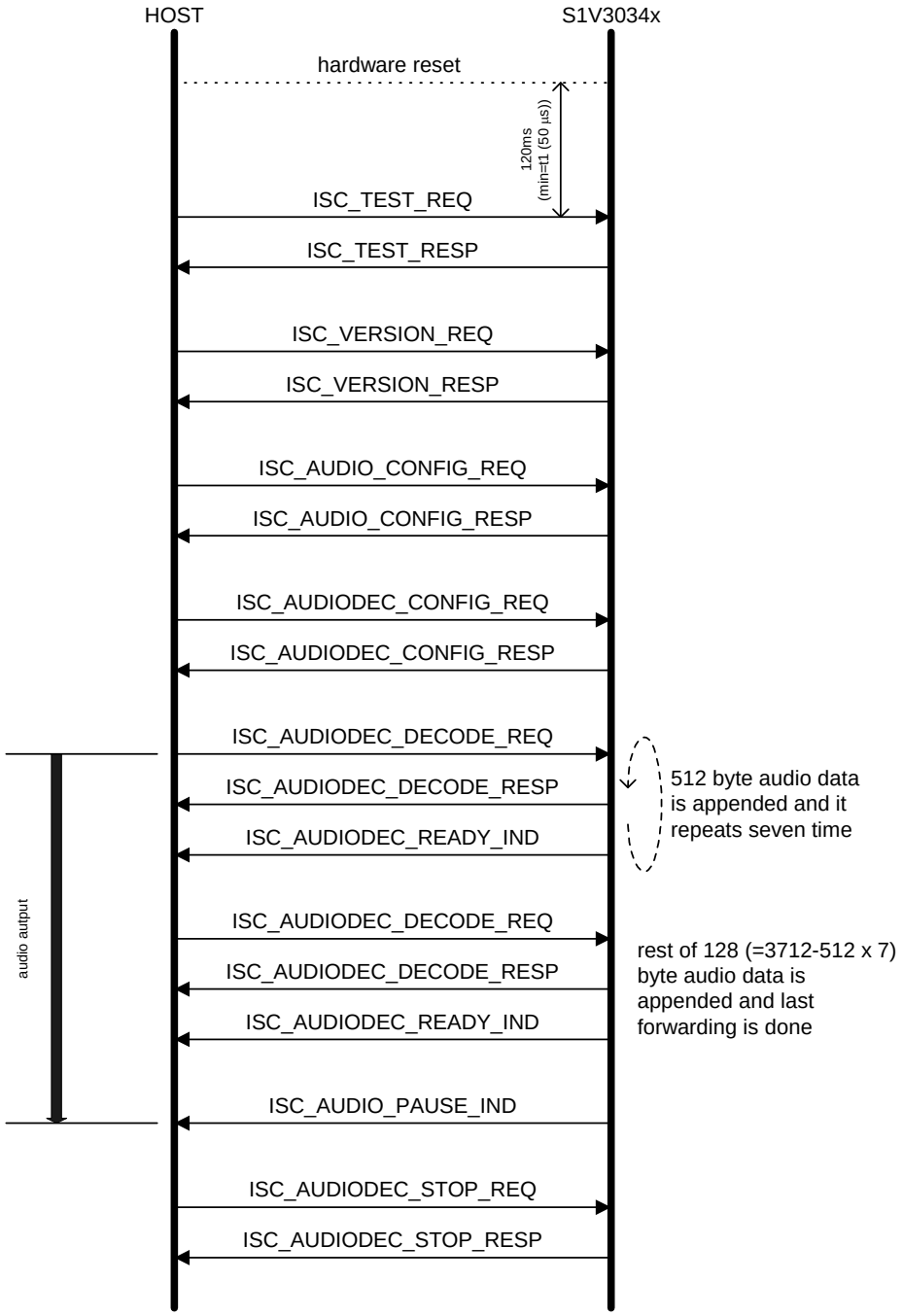


Figure 6.1 Normal message flowchart for streaming playback

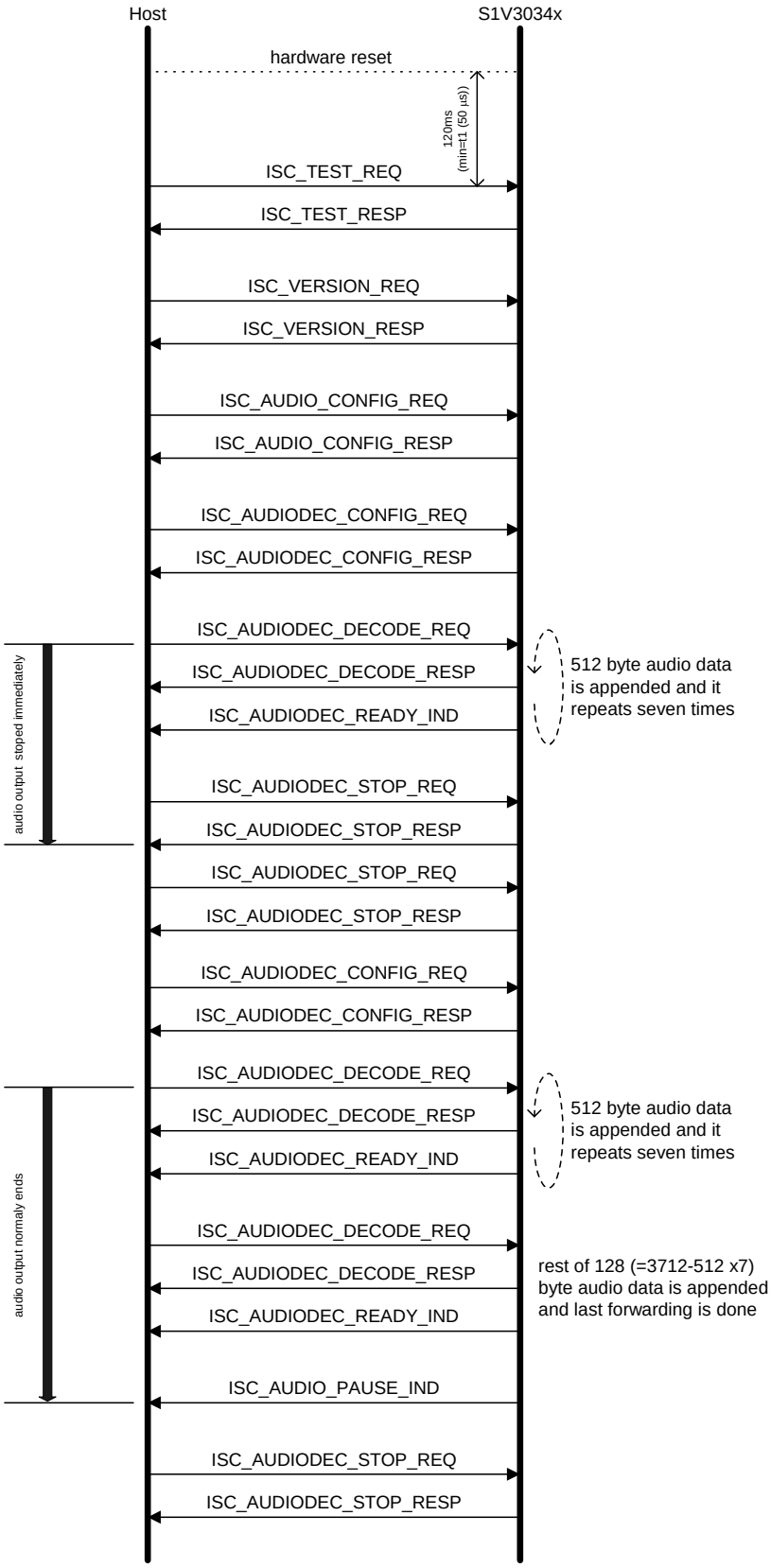


Figure 6.2 Message flowchart for streaming playback at or after immediate halt during streaming playback

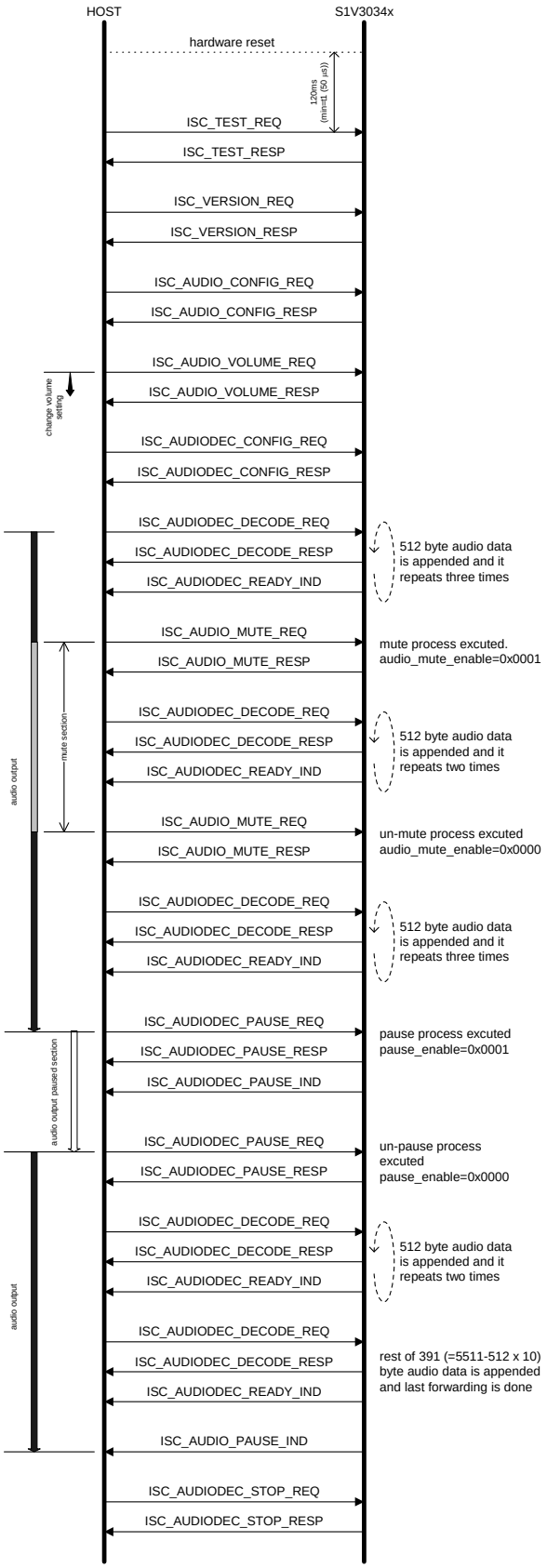


Figure 6.3 Message flowchart for mute and pause during streaming playback

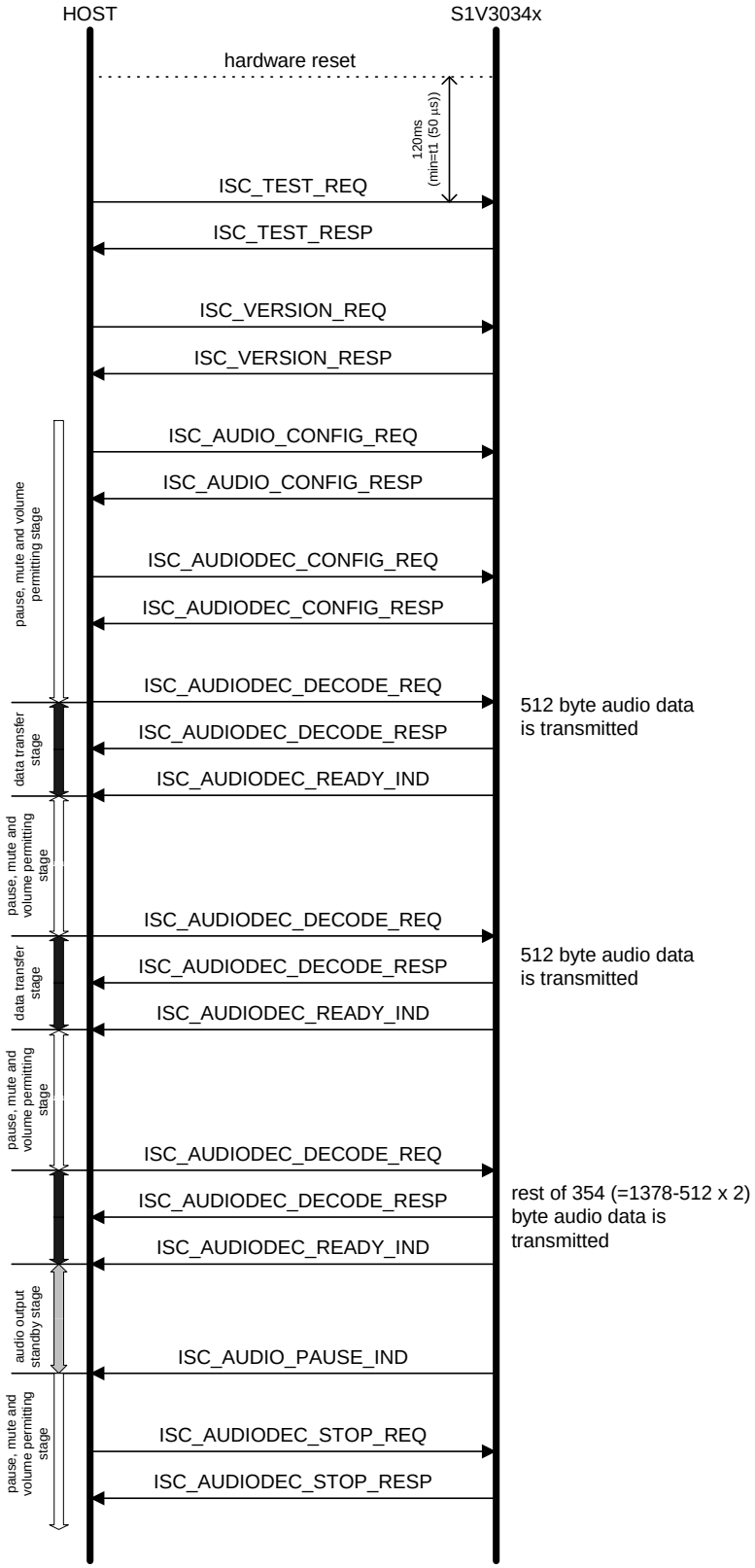


Figure 6.4 Pause and other permitted sections, data transfer sections, and audio output complete sections during streaming playback

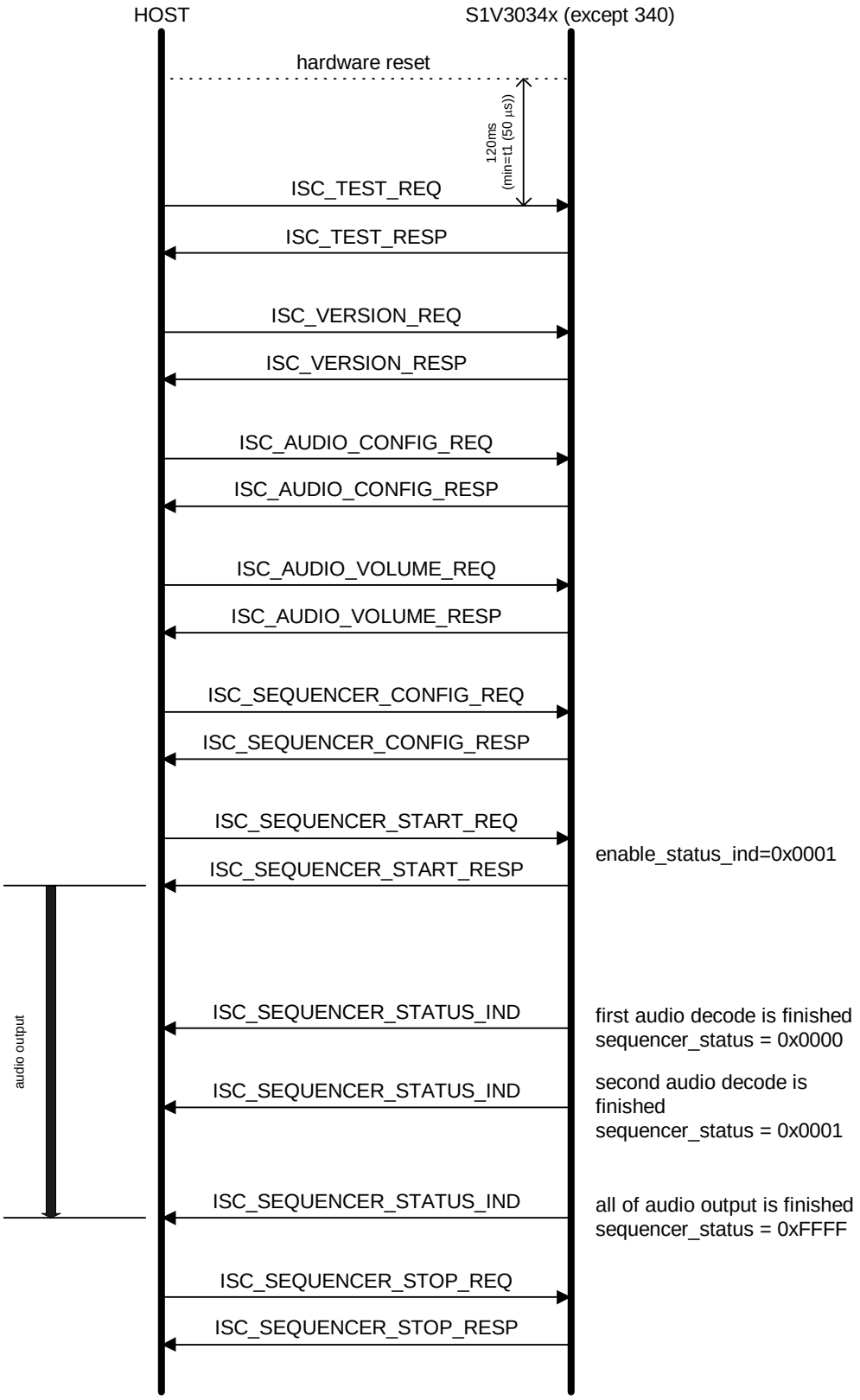


Figure 6.5 Normal message flowchart for sequence playback (status notification enabled)

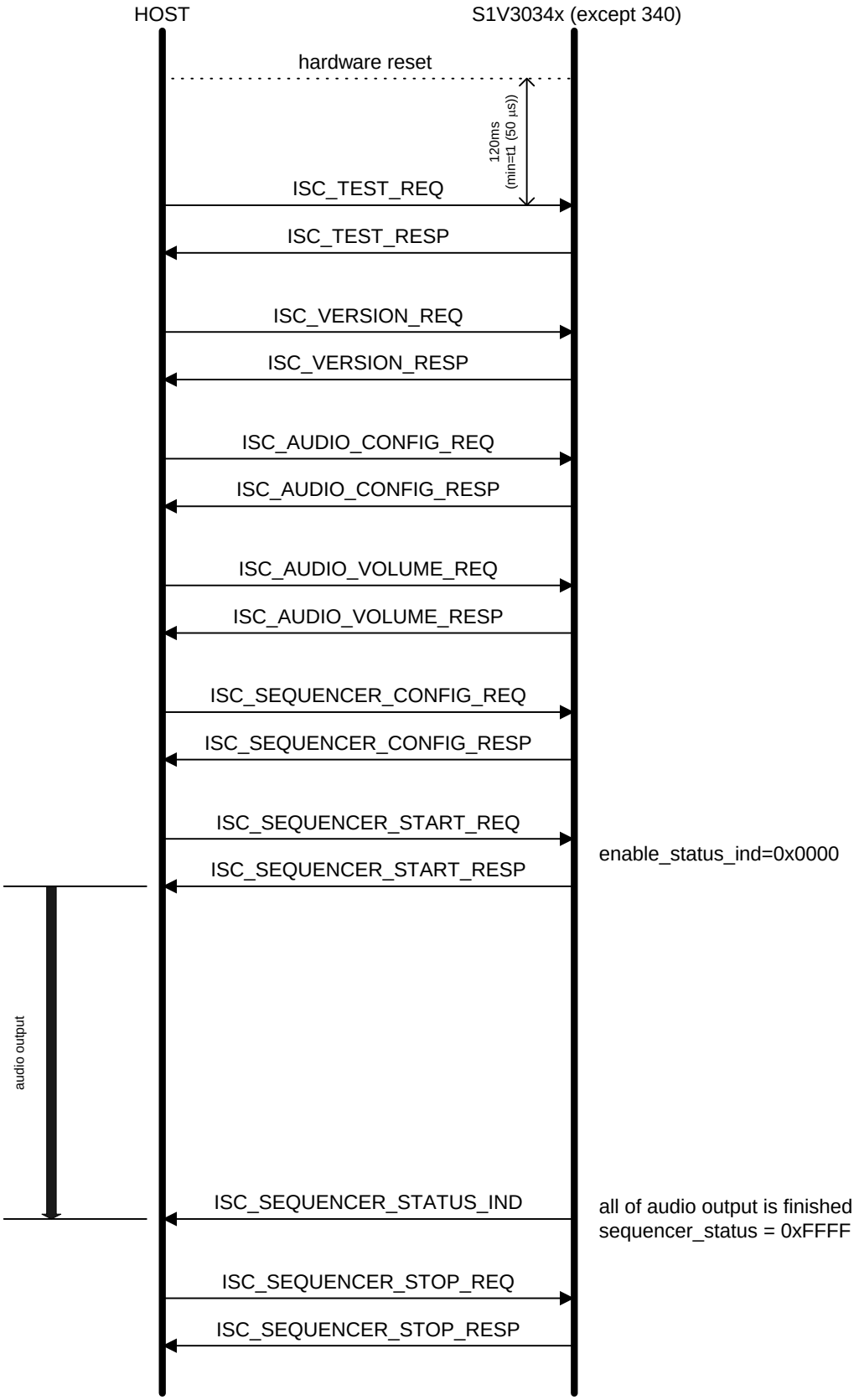


Figure 6.6 Normal message flowchart for sequence playback (status notification disabled)

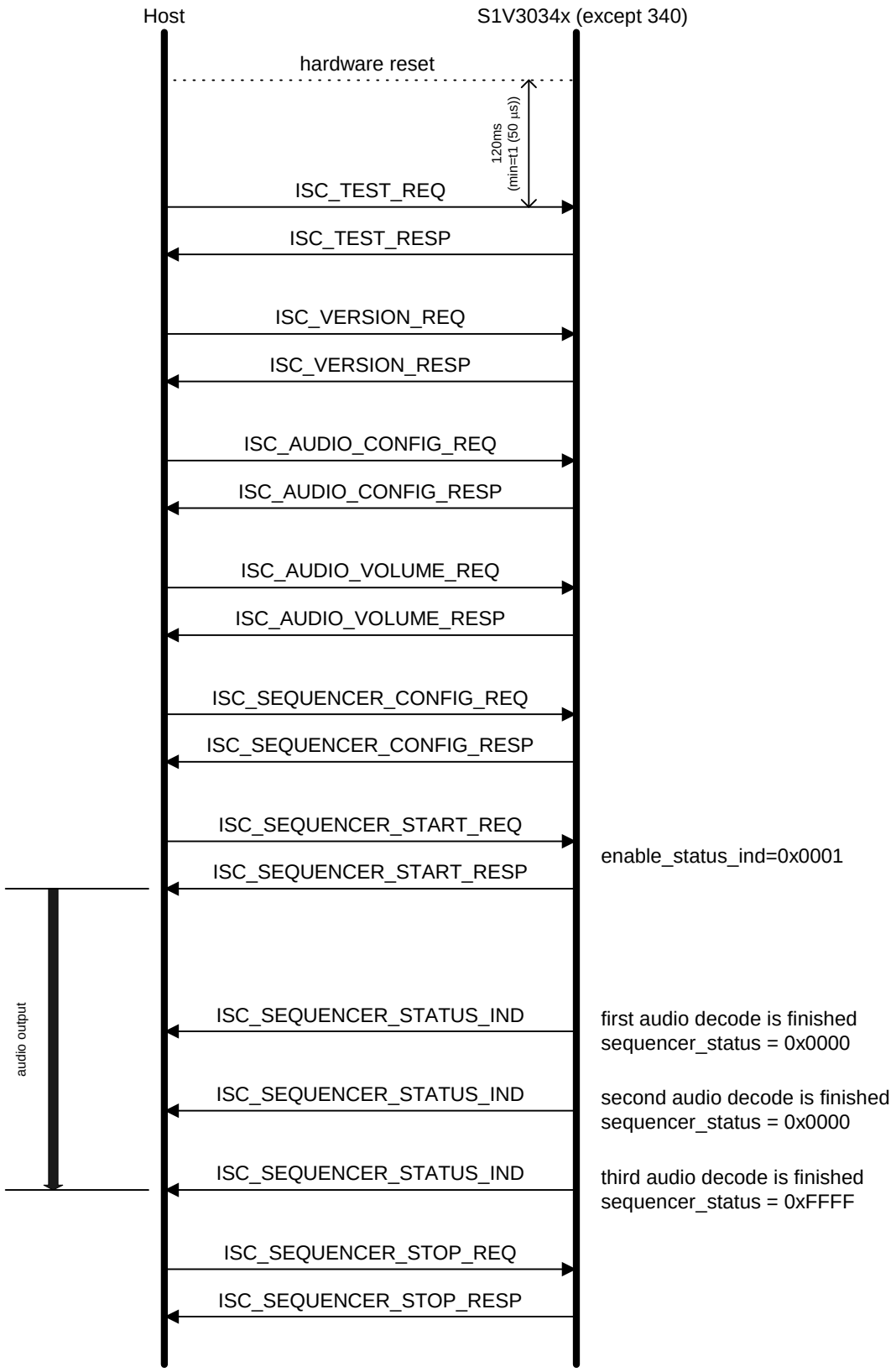


Figure 6.7 Message flowchart for sequence playback when play_count = 3

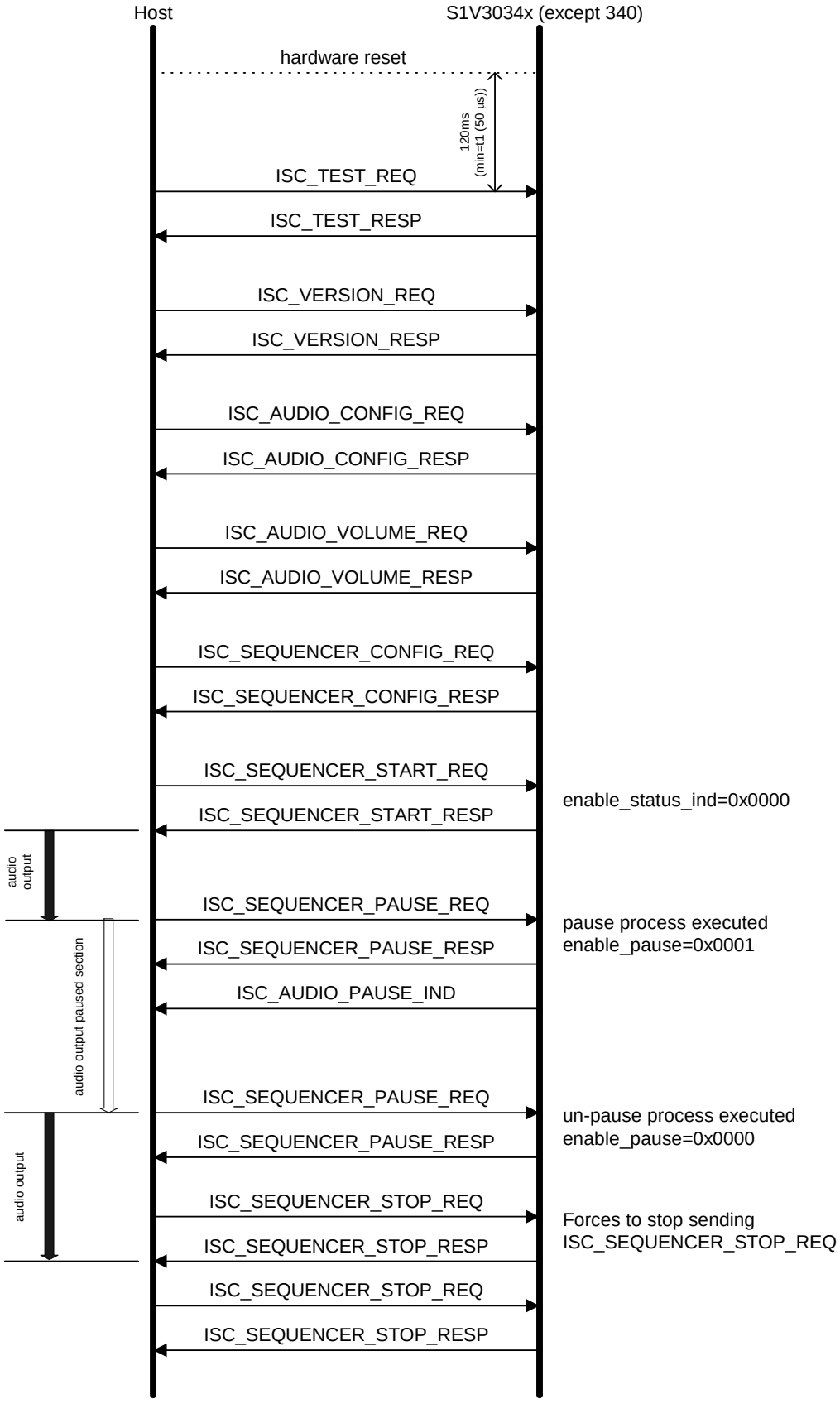


Figure 6.8 Message flowchart for pause and immediate halt during sequence playback

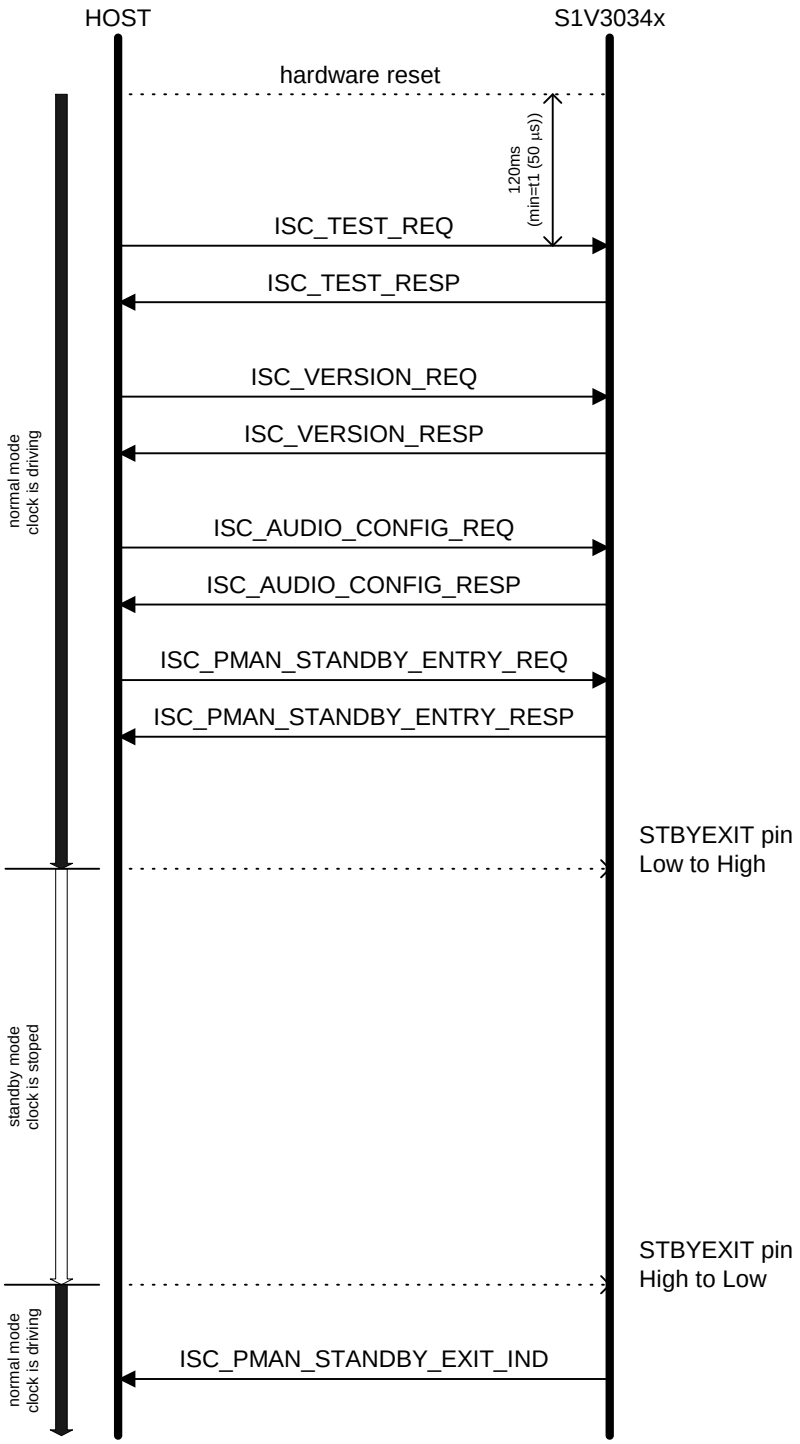


Figure 6.9 Normal message flowchart for power management

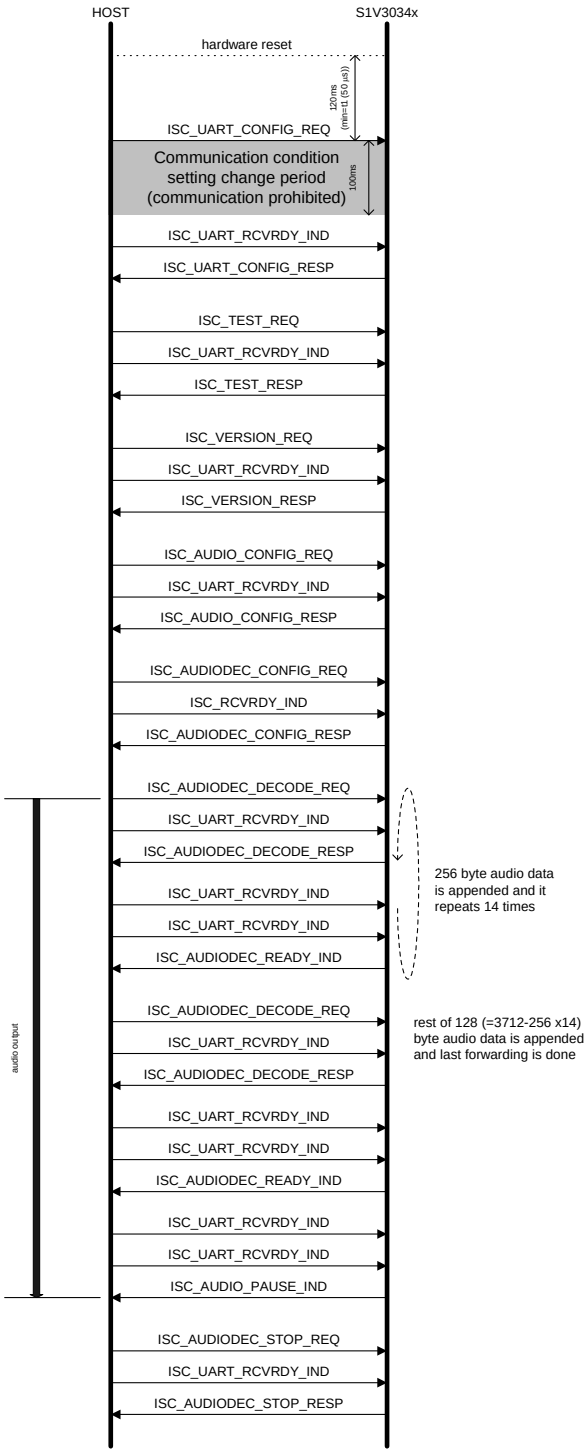


Figure 6.10 Normal message flowchart for streaming playback using UART communication

7. Sample Program Specifications

7.1 Overview

This section describes specifications for sample programs provided to users of the S1V3034x. The sample programs are designed to control the S1V3034x when incorporated into client systems. The following discussion clarifies specifications for the API functions used in the sample programs.

Note: The sample programs were designed for the actual host system used by Seiko Epson to evaluate the control performance of the S1V3034x. For this reason, we do not guarantee that the sample programs will function as intended when used in client systems.

7.2 Obtaining Sample Programs

Download the sample programs described in this section from the Seiko Epson Speech & Audio User's Site. For the Speech & Audio User's Site URL, please refer to the documentation provided with the S1V3034x evaluation kit.

7.3 Types of Sample Programs

Sample programs come in two types: full and abridged versions.

- Full-version sample programs

  General-purpose sample programs for error processing when sending and receiving messages.

- Abridged-version sample programs

  Sample programs focusing on audio playback with simplified message send/receive functions, designed to run on even low-performance host CPUs.

7.4 File Configuration

Sample programs are stored in the following directories:

\normal Stores the source files for full-version sample programs.

\simple Stores the source files for abridged-version sample programs.

\common Stores the common source files for both full and abridged versions.

Note: When a sample program is incorporated into client systems, some source file must be modified to suit the specifications of the client system.

7.4.1 Main Program Files

The main program files describe a series of control programs to implement a streamed playback process, sequenced playback process, and power management process under the control of the host processor.

Table 7.1 shows a list of main program files. Each main program file in the Table 7.1 is a main program containing the “main” function. Select a single main program file.

Table 7.1 List of main program files

Filename	Directory	Description
main_streaming.c	normal	Program for streaming playback
main_sequnecer.c	normal	Program for sequence playback
main_power_management.c	normal	Program for power management process
main_streaming_simple.c	simple	Simplified program for streaming playback
main_sequencer_simple.c	simple	Simplified program for sequence playback

7.4.2 S1V3034x Control API Function Definition Files

The S1V3034x control API function definition files define API functions used to control the S1V3034x.

Table 7.2 lists the API function definition files used for S1V3034x control.

Refer to “7.6 S1V3034x Control API Function Specifications” for detailed API function specifications.

Table 7.2 List of S1V3034x Control API Function Definition Files

Filename	Directory	Description
spi_api.c	common	Source file defining S1V3034x control API functions
spi_api.h	common	Header file declaring S1V3034x control API functions

Note: “spi_api.c” is a control program for SPI on a host system used by Seiko Epson for S1V3034x control evaluations. The client will need to make modifications to suit the client’s specifications before incorporation into the client system.

7.4.3 Message Files

Message files define REQ message array tables. Table 7.3 shows a list of message files.

Table 7.3 List of message files

Filename	Directory	Description
isc_msgs.c	normal	REQ message array table definition file for full-version sample programs (except for ISC_AUDIODEC_DECODE_REQ and ISC_SEQUENCER_CONFIG_REQ)
isc_audiodec_decode_req.c	normal	ISC_AUDIODEC_DECODE_REQ array table definition file for full-version sample programs (dummy file)
isc_sequencer_config_req.c	normal	ISC_SEQUENCER_CONFIG_REQ array table definition file for full-version sample programs (dummy file)
isc_msgs_simple.c	simple	REQ message array table definition file for abridged-version sample programs (except for ISC_AUDIODEC_DECODE_REQ and ISC_SEQUENCER_CONFIG_REQ)
isc_audiodec_decode_req_simple.c	simple	ISC_AUDIODEC_DECODE_REQ array table definition file for abridged-version sample programs (dummy file)
isc_sequencer_config_req_simple.c	simple	ISC_SEQUENCER_CONFIG_REQ array table definition file for abridged-version sample programs (dummy file)

Message files define data (as shown in Table 7.4) and describe the REQ message data sequence, including “0x00 and 0xAA” (i.e., message start padding and start command).

Table 7.4 Message file example (full version)

```
unsigned char aucIscResetReq[ ] = {
   0x00, 0xAA, 0x01, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00,
};
```

The message files for the abridged-version sample programs contain the same data strings as the message files for the full version, except that each string is followed by multiple “0x00” for padding (shown in underlined part in Table 7.5). This “0x00” is used to skip a RESP message to be received after REQ message transmission and to simplify the message send/receive process.

Table 7.5 Message file example (abridged version)

```
unsigned char aucIscResetReq[ ] = {
   0x00, 0xAA, 0x01, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
```

Note: isc_audiodec_decoder_req(_simple).c and isc_sequencer_config(_simple).c in Table 7.3 are dummy files describing empty array tables. When using this sample program, replace them with files of the same name created with the EPSON Speech IC Speech Guide Creation Tool. For more information on the EPSON Speech IC Speech Guide Creation Tool, please refer to the *EPSON Speech IC Speech Guide Creation Tool User Guide*.

7.4.4 Other Source Files

Table 7.6 lists relevant source files other than those described above.

Table 7.6 List of other source files

Filename	Directory	Description
isc_msgs.h	common	Header file defining message length and ID
reg.h	common	Header file defining SPI register map

Note: "reg.h" describes the SPI register map on a host system used by Seiko Epson for S1V3034x control evaluations. The client will need to make modifications to suit the client's SPI specifications before incorporation into the client system. For the specifications of the SPI register on the actual host system used by Seiko Epson, refer to "8.2 Examples of SPI Register Specifications."

7.5 Main Program Specifications

The sample programs include five main programs (see Section 7.4.1).
Each main program implements control functions according to the following procedures.

Note 1: In each main program described below, the ISC_TEST_REQ message array data to be sent in step (6) describes a dummy key code. When using the sample programs, replace the dummy key code in the ISC_TEST_REQ message array data with the client-specific key code provided by us.

Note 2: The sample programs do not entail interrupt processing. If the client's host system requires interrupt processing, the sample programs must be modified to suit the interrupt specifications of the actual host system used.

7.5.1 main_streaming.c

This is the main program used for streaming playback. It initializes the SPI I/F and initiates streaming playback. It also handles sound level processing and pause processing during streaming playback.

(1) Initializes the SPI I/F.

(2) Sets the S1V3034x chip select signal (NSCSS) to Low (assert).

(3) Sets the external audio amplifier mute signal (MUTE) to High (disable).

(4) Guarantees a wait time of 120 ms for the S1V3034x.

(5) Resets the S1V3034x.
(Transmission/reception of ISC_RESET_REQ/RESP)

(6) Sets the key code and communication system.
(Transmission/reception of ISC_TEST_REQ/RESP)

(7) Requests notification of hardware version.
(Transmission/reception of ISC_VERSION_REQ/RESP)

(8) Sets audio output.
(Transmission/reception of ISC_AUIDO_CONFIG_REQ/RESP)

< Start of streaming playback >

(9) Sets streaming playback.
(Transmission/reception of ISC_AUDIODEC_CONFIG_REQ/RESP)

(10) Sends the audio data.
(Transmission/reception of ISC_AUDIODEC_DECODE_REQ/RESP)

(11) Awaits audio data transmission permit message sent from the S1V3034x.
(Reception of ISC_AUDIODEC_READY_IND)

(12) Repeats steps (10) and (11) until the end of the audio data is reached.

(13) Adjusts the sound level during streaming playback in step (12).
(Reception of ISC_ SEQUENCER_STATUS_IND)

(14) Mutes the sound during streaming playback in step (12).
(Transmission/reception of ISC_AUDIO_MUTE_REQ/RESP)

< During mute >

(15) Cancels muting.
(Transmission/reception of ISC_AUDIO_MUTE_REQ/RESP)

(16) Pauses during streaming playback in step (12).
(Transmission/reception of ISC_AUDIODEC_PAUSE_REQ/RESP)
(Reception of ISC_AUDIO_PAUSE_IND)

< During pause >

(17) Cancels pause.
(Transmission/reception of ISC_AUDIODEC_PAUSE_REQ/RESP)

(18) Awaits streaming playback completion message sent from the S1V3034x.
(Reception of ISC_AUDIO_PAUSE_IND)

(19) Performs streaming playback completion processing.
(Transmission/reception of ISC_AUDIODEC_STOP_REQ/RESP)

< Streaming playback completion >

(20) Sets the external audio amplifier mute signal (MUTE) to Low (enable).

(21) Sets the S1V3034x chip select signal (NSCSS) to High (deassert).

7.5.2 main_sequencer.c

This is the main program used for sequence playback. It initializes SPI I/F and initiates sequence playback. It also handles pause processing during sequence playback.

(1) Initializes the SPI I/F.

(2) Sets the S1V3034x chip select signal (NSCSS) to Low (assert).

(3) Sets the external audio amplifier mute signal (MUTE) to High (disable).

(4) Guarantees a wait time of 120 ms for the S1V3034x.

(5) Resets the S1V3034x.
(Transmission/reception of ISC_RESET_REQ/RESP)

(6) Sets the key code and communication system.
(Transmission/reception of ISC_TEST_REQ/RESP)

(7) Requests notification of hardware version.
(Transmission/reception of ISC_VERSION_REQ/RESP)

(8) Sets audio output.
(Transmission/reception of ISC_AUIDO_CONFIG_REQ/RESP)

< Start of sequence playback >

(9) Sets sequence playback.
(Transmission/reception of ISC_SEQUENCER_CONFIG_REQ/RESP)

(10) Starts the sequence playback.
(Transmission/reception of ISC_SEQUENCER_START_REQ/RESP)

(11) Pauses during the sequence playback in step (10).
(Transmission/reception of ISC_SEQUENCER_PAUSE_REQ/RESP)
(Reception of ISC_AUDIO_PAUSE_IND)

< During pause >

(12) Cancels pause.
(Transmission/reception of ISC_SEQUENCER_PAUSE_REQ/RESP)

(13) Awaits sequence playback completion message sent from the S1V3034x.
(Reception of ISC_SEQUENCER_STATUS_IND)

(14) Performs sequence playback completion processing.
(Transmission/reception of ISC_SEQUENCER_STOP_REQ/RESP)

< Sequence playback completion >

(15) Sets the external audio amplifier mute signal (MUTE) to Low (enable).

(16) Sets the S1V3034x chip select signal (NSCSS) to High (deassert).

7.5.3 main_power_management.c

This is the main program used to control power management. It initializes the SPI I/F and controls power management.

(1) Initializes the SPI I/F.

(2) Sets the S1V3034x chip select signal (NSCSS) to Low (assert).

(3) Sets the external audio amplifier mute signal (MUTE) to High (disable).

(4) Guarantees a wait time of 120 ms for the S1V3034x.

(5) Resets the S1V3034x.
(Transmission/reception of ISC_RESET_REQ/RESP)

(6) Sets the key code and communication system.
(Transmission/reception of ISC_TEST_REQ/RESP)

(7) Requests notification of hardware version.
(Transmission/reception of ISC_VERSION_REQ/RESP)

(8) Sets audio output.
(Transmission/reception of ISC_AUIDO_CONFIG_REQ/RESP)

< Start of streaming playback or Start of sequence playback >

…

< Streaming playback completion or Sequence playback completion >

< Start of standby mode >

(9) Sets the external audio amplifier mute signal (MUTE) to Low (enable).

(10) Sets the standby signal (STBYEXIT) to Low (deassert).

(11) Guarantees a wait time of 50 μs for the S1V3034x.

(12) Performs entry to the standby mode.
(Transmission/reception of ISC_PMAN_STANDBY_ENTRY_REQ/RESP)

(13) Sets the standby signal (STBYEXIT) to High (assert).

< During S1V3034x standby mode >

(14) Sets the standby signal (STBYEXIT) to Low (deassert).

(15) Guarantees a wait time of 120 ms for the S1V3034x.

(16) Awaits standby mode completion message sent from the S1V3034x.
(Awaits reception of ISC_PMAN_STANDBY_EXIT_IND)

(17) Sets the external audio amplifier mute signal (MUTE) to High (disable).

< Release of standby mode >

< Start of streaming playback or Start of sequence playback >

…

< Streaming playback completion or Sequence playback completion >

(18) Sets the external audio amplifier mute signal (MUTE) to Low (enable).

(19) Sets the S1V3034x chip select signal (NSCSS) to High (deassert).

7.5.4 main_streaming_simple.c

This is the main program used for streaming playback. It omits error processing to simplify message transmission/reception. It initializes the SPI I/F and initiates streaming playback.

(1) Initializes the SPI I/F.

(2) Sets the S1V3034x chip select signal (NSCSS) to Low (assert).

(3) Sets the external audio amplifier mute signal (MUTE) to High (disable).

(4) Guarantees a wait time of 120 ms for the S1V3034x.

(5) Resets the S1V3034x.
(Transmission of ISC_RESET_REQ)

(6) Sets the key code and communication system.
(Transmission of ISC_TEST_REQ)

(7) Requests notification of hardware version.
(Transmission of ISC_VERSION_REQ)

(8) Sets audio output.
(Transmission of ISC_AUIDO_CONFIG_REQ)

< Start of streaming playback >

(9) Sets streaming playback.
(Transmission of ISC_AUDIODEC_CONFIG_REQ)

(10) Transmits the audio data.
(Transmission of ISC_AUDIODEC_DECODE_REQ)

(11) Awaits audio data transmission permit message sent from the S1V3034x.
(Reception of ISC_AUDIODEC_READY_IND)

(12) Repeats steps (10) and (11) until the end of the audio data is reached.

(13) Awaits streaming playback completion message sent from the S1V3034x.
(Reception of ISC_AUDIO_PAUSE_IND)

(14) Performs streaming playback completion processing.
(Transmission of ISC_AUDIODEC_STOP_REQ)

< Streaming playback completion >

(15) Sets the external audio amplifier mute signal (MUTE) to Low (enable).

(16) Sets the S1V3034x chip select signal (NSCSS) to High (deassert).

7.5.5 main_sequencer_simple.c

This is the main program used for sequence playback. It omits error processing to simplify message transmission/reception. It initializes the SPI I/F and initiates sequence playback.

(1) Initializes the SPI I/F.

(2) Sets the S1V3034x chip select signal (NSCSS) to Low (assert).

(3) Sets the external audio amplifier mute signal (MUTE) to High (disable).

(4) Guarantees a wait time of 120 ms for the S1V3034x.

(5) Resets the S1V3034x.
(Transmission of ISC_RESET_REQ)

(6) Sets the key code and communication system.
(Transmission of ISC_TEST_REQ)

(7) Requests notification of hardware version.
(Transmission/reception of ISC_VERSION_REQ/RESP)

(8) Sets audio output.
(Transmission of ISC_AUIDO_CONFIG_REQ)

< Start of sequence playback >

(9) Sets sequence playback.
(Transmission of ISC_SEQUENCER_CONFIG_REQ)

(10) Starts the sequence playback.
(Transmission of ISC_SEQUENCER_START_REQ)

(11) Awaits sequence playback completion message sent from the S1V3034x.
(Reception of ISC_SEQUENCER_STATUS_IND)

(12) Performs sequence playback completion processing.
(Transmission/reception of ISC_SEQUENCER_STOP_REQ/RESP)

< Sequence playback completion >

(13) Sets the external audio amplifier mute signal (MUTE) to Low (enable).

(14) Sets the S1V3034x chip select signal (NSCSS) to High (deassert).

7.6 S1V3034x Control API Function Specifications

This section describes the specifications for the API functions used to control the S1V3034x.

Note: The API functions described below are control programs for SPI on a host system used by Seiko Epson for S1V3034x control evaluations. The client will need to make the appropriate modifications to API functions to suit the client's specifications before incorporation into the client system.

7.6.1 SPI_Initialize

[Syntax]

void SPI_Initialize (void)

[Function]

Initializes the SPI.

[Input argument]

None

[Output argument]

None

[Return value]

None

[Function description]

This API initializes the SPI registers. This function complies with the host processor SPI specifications used for the Seiko Epson evaluation system and performs the following actions.

(1) Disables SPI.

(2) Disables SPI interrupt settings.

(3) Sets the input/output terminals.

(4) Sets the SPI clock frequency.

(5) Sets the SPI clock mode (polarity and phase).

(6) Sets the SPI to master mode.

(7) Sets the wait cycle between data transfers.

(8) Sets the bit mask for received data.

(9) Enables SPI.

Note: For detailed information, refer to "8.2 Examples of SPI Register Specifications."

7.6.2 SPI_SendReceiveByte

[Syntax]

unsigned char SPI_SendReceiveByte (
unsigned char ucSendData)

[Function]

Sends/receives 1-byte data.

[Input argument]

ucSendData Set 1-byte transmission data.

[Output argument]

None

[Return value]

Returns 1-byte received data.

[Function description]

This API sends/receives 1-byte data to or from the S1V3034x via SPI.

7.6.3 SPI_SendMessage

[Syntax]

```
int     SPI_SendMessage (
                unsigned char     *pucSendMessage,
                unsigned short    *pusReceivedMessageID)
```

[Function]

Sends a message to the S1V3034x.

[Input argument]

pucSendMessage	Specify the address of the memory area where the transmitted message is stored.
pusReceivedMessageID	Specify the pointer to the variable storing the message ID of the message received while data is transmitted.

[Output argument]

None

[Return value]

Returns a “0” for normal termination; otherwise returns the following error code.

Error code	Value	Description
SPIERR_SUCCESS	0	Returned for normal termination.
SPIERR_NULL_PTR	-1	Returned if the argument is a null pointer.

[Function description]

This API sends a REQ message to the S1V3034x.

This function sends data by referring to the value of the length field in the REQ message to be sent.

When the API receives a message from the S1V3034x while sending a REQ message, it stores the message ID in pusReceivedMessageID.

Note: This function is a message-sending function for full-version sample programs.

7.6.4 SPI_ReceiveMessage

[Syntax]

```
int     SPI_ReceiveMessage (
                unsigned short      *pucReceivedMessageID)
```

[Function]

Receives a message from the S1V3034x.

[Input argument]

None

[Output argument]

pucReceivedMessageID	Specify the pointer to the variable storing the message ID of the received message.

[Return value]

Returns a "0" for normal termination; otherwise returns one of the following error codes.

Error code	Value	Description
SPIERR_SUCCESS	0	Returned for normal termination.
SPIERR_NULL_PTR	-1	Returned if the argument is a null pointer.
SPIERR_GET_ERROR_CODE	-2	Returned if the message received from the S1V3034x contains an error code.
SPIERR_RESERVED_MESSAGE_ID	-3	Returned if the message received from the S1V3034x is a reserved ID.
SPIERR_ISC_VERSION_RESP	-4	Returned if the ISC_VERSION_RESP contents fail to match specifications.
SPIERR_TIMEOUT	1	Returned if the message sent from the S1V3034x is not received within the duration specified for the timeout.

[Function description]

This API receives a RESP message or IND message sent from the S1V3034x.

The API continues to receive data from the S1V3034x until the message start command (0x00 0xAA) is detected. After detecting the message header, it obtains the values set in the length field and id field contained in the received data and continues to receive data based on the value of the length field obtained. The obtained value of the id field is stored in the output argument, pucReceivedMessageID.

Note: This function is a message-receiving function for full-version sample programs. The function argument format is the same as for "7.6.6 SPI_ReceiveMessage_simple," except that this function performs error processing, including ID checks of received messages, as internal processing.

7.6.5 SPI_SendMessage_simple

[Syntax]

```
int     SPI_SendMessage_simple (
                unsigned char   *pucSendMessage,
                int             iSendMessageLength)
```

[Function]

Sends a message to the S1V3034x.

[Input argument]

pucSendMessage	Specify the address of the memory area where the transmitted message is stored.
iSendMessageLength	Specify the length in bytes of the message to be transmitted.

[Output argument]

None

[Return value]

Returns a “0” for normal termination; otherwise returns the following error code.

Error code	Value	Description
SPIERR_SUCCESS	0	Returned for normal termination.
SPIERR_NULL_PTR	-1	Returned if the argument is a null pointer.

[Function description]

This API sends a REQ message to the S1V3034x.

The API sends data with the number of bytes specified by iSendMessageLength.

Note: This function is a message transmission function for abridged-version sample programs. Abridged-version sample programs use this function to send REQ messages with padding data for RESP message-skipping. iSendMessageLength therefore specifies REQ message length, including padding data.

7.6.6 SPI_ReceiveMessage_simple

[Syntax]

```
int     SPI_ReceiveMessage_simple (
                unsigned short      *pucReceivedMessageID)
```

[Function]

Receives a message from the S1V3034x.

[Input argument]

iReceivedMessageLength	Specify the length in bytes of the received message.

[Output argument]

pucReceivedMessageID	Specify the pointer to the variable storing the message ID of the received message.

[Return value]

Returns a “0” for normal termination; otherwise returns the following error code.

Error code	Value	Description
SPIERR_SUCCESS	0	Returned for normal termination.
SPIERR_NULL_PTR	-1	Returned if the argument is a null pointer.

[Function description]

This API receives a RESP message or IND message sent from the S1V3034x.

The API continues to receive data from the S1V3034x until the message start command (0x00 0xAA) is detected. After detecting the message header, it obtains the values set in the length field and id field contained in the received data and continues to receive data based on the value of the length field obtained. The obtained value of the id field is stored in the output argument, pucReceivedMessageID.

Note: This function is a message-receiving function for abridged-version sample programs. Abridged-version sample programs use this function only to receive IND messages. The function argument format is the same as for “7.6.4 SPI_ReceiveMessage,” but with simplified internal processing.

7.6.7 GPIO_ControlChipSelect

[Syntax]

```
void    GPIO_ControlChipSelect (
                int             iValue)
```

[Function]

Controls the chip select signal (NSCSS) of the S1V3034x.

[Input argument]

iValue	Specify either 0: Low or 1: High.

[Output argument]

None

[Return value]

None

[Function description]

This API controls the chip select signal (NSCSS) of the S1V3034x. The chip select signal must be set to Low (0) for communications with the S1V3034x via SPI.

7.6.8 GPIO_ControlStandby

[Syntax]

```
void    GPIO_ControlStandby (
                int             iValue)
```

[Function]

Controls the standby signal (STBYEXIT).

[Input argument]

iValue	Specify either 0: Low or 1: High.

[Output argument]

None

[Return value]

None

[Function description]

This API controls the standby signal (STBYEXIT) of the S1V3034x. For control of the standby signal, this API is used in combination with the ISC_PMAN_STANDBY_REQ message. It handles the transition or cancellation of the S1V3034x standby mode.

When this function is used, the standby signal for the S1V3034x must be linked to the host processor GPIO.

7.6.9 GPIO_ControlMute

[Syntax]

```
void    GPIO_ControlMute (
                int             iValue)
```

[Function]

Controls the mute signal (MUTE).

[Input argument]

iValue	Specify either 0: Low or 1: High.

[Output argument]

None

[Return value]

None

[Function description]

This API controls the mute signal (MUTE) of an audio amplifier.

When this function is used, the mute signal for the audio amplifier must be linked to the host processor GPIO.

Note: In the sample programs, the mute signal is disabled by Hight (1) and enabled by Low (0) to suit the specifications of the audio amplifier mounted on the actual host system used by Seiko Epson to evaluate S1V3034x control performance. When incorporated into a client system, the mute signal polarity must be modified to suit the specifications of the audio amplifier mounted on the client's system.

8. Appendix

8.1 Confirming Communication Link

The status of communications between the host and S1V3034x can be checked using ISC_VERSION_REQ/RESP. Since ISC_VERSION_RESP always has fixed values, these values can be used as expected values for comparison to actual data sent to the host via the serial communication interface. If the actual data matches the expected values, communications have been established between the host and S1V3034x. The following table shows the expected values of ISC_VERSION_RESP sent by the S1V3034x.

Byte	Field	Value
0	length (lsb)	0x14
1	length (msb)	0x00
2	msg_id (lsb)	0x06
3	msg_id (msb)	0x00
4	hw_id_int	0x01
5	hw_id_frac	0x00
6	fw_version_int	0x01
7	fw_version_frac	0x00
8	fw_features	0x00
9		0x40
10		0x00
11		0x00
12-15	reserved	0x00
16-19	padding	0x00

8.2 Examples of SPI Register Specifications

The following table shows specifications for the SPI registers among the registers of the host processor used by the sample programs.

Table 8.1 List of SPI control registers

Address	Register name	Size	Function
0x00301700	SPI Receive Data Register (pSPI_RXD)	32	Received data
0x00301704	SPI Transmit Data Register (pSPI_TXD)	32	Transmission data
0x00301708	SPI Control Register 1 (pSPI_CTL1)	32	SPI transfer condition setting
0x0030170C	SPI Control Register 2 (pSPI_CTL2)	32	Slave mode control
0x00301710	SPI Wait Register (pSPI_WAIT)	32	Inter-character wait cycle setting
0x00301714	SPI Status Register (pSPI_STAT)	32	SPI transfer/error status
0x00301718	SPI Interrupt Control Register (pSPI_INT)	32	SPI interrupt control
0x0030171C	SPI Receive Data Mask Register (pSPI_RXMK)	32	Received data bit mask setting

Individual SPI control registers are described below.

The SPI control registers are allocated to the 32-bit device area ranging from 0x301700 to 0x30171C and are word-accessible.

Note:

- The SPI control registers are only word-accessible. Do not read or write using half words or in bytes.
- When setting an SPI control register, be sure to write 0 to the Reserved bit. Do not write 1 to the Reserved bit.

0x301700: SPI Receive Data Register (pSPI_RXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI receive data register (pSPI_RXD)	00301700 (W)	D31 | D0	SPIRXD31 | SPIRXD0	SPI receive data SPIRXD31 = MSB SPIRXD0 = LSB	0x0 to 0xFFFFFFFF	0x0	R	

D[31:0] SPIRXD[31:0]: SPI Receive Data Bits

Stores received data. (default: 0x0)

After data is received and data in the shift register is transferred to this register, the RDFF (D2/0x301714) is set to 1 (data full). At the same time, a received data full interrupt is generated. This allows data readout until the subsequent data is received. If the subsequent data is received before this register is read out, the existing data is overwritten by the newly received data, and the RDOF (D3/0x301714) is set to 1 (data overflow). At the same time, a received data overflow interrupt is generated.

Serial data input from the SDI terminal undergoes parallel conversion, with the MSB at the beginning, the High level bit set to 1, and the Low level bit set to 0, and the data loaded to this register.

It is also possible to mask (set to 0) a specified number of higher-order bits during loading from the shift register by setting the SPI Receive Data Mask Register (0x30171C).

This register is for readout only and cannot be written to.

0x301704: SPI Transmit Data Register (pSPI_TXD)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI transmit data register (pSPI_TXD)	00301704 (W)	D31 | D0	SPITXD31 | SPITXD0	SPI transmit data SPITXD31 = MSB SPITXD0 = LSB	0x0 to 0xFFFFFFFF	0x0	R/W	

D[31:0] SPITXD[31:0]: SPI Transmit Data Bits

Sets transmission data. (default: 0x0)

In master mode, writing data to this register initiates data transmission. In slave mode, the content of this register is sent to the shift register when a clock signal is input from the master, and then data transmission starts.

When data written to this register is transferred to the shift register, the TDEF (D4/0x301714) is set to 1 (empty). At the same time, an empty transmission data interrupt is generated. Subsequently, this allows transmission data to be written even during data transmission.

The SDO terminal outputs serial-converted data, with the MSB at the beginning, the bit set to 1 as High level, and the bit set to 0 as Low level.

When the number of transfer data bits is set to less than 32 in the BPT[4:0] (D[14:10]/0x301708), only the specified number of bits on the lower-order side of this register is transmitted.

0x301708: SPI Control Register 1 (pSPI_CTL1)

Register name	Address	Bit	Name	Function	Setting	Init	R/W	Remarks
SPI control register 1 (pSPI_CTL1)	0030170 8 (W)	D31 | D15	-	reserved	-	-	-	0 when being read.
		D14 D13 D12 D11 D10	BPT4 BPT3 BPT2 BPT1 BPT0	Number of data bits per transfer	Number of data bits per transfer = BPT + 1	0 0 0 0 0	R/W	
		D9	CPHA	SPI_CLK phase selection	1 Phase 1 0 Phase 0	0	R/W	
		D8	CPOL	SPI_CLK polarity selection	1 Active low 0 Active high	0	R/W	
		D7	MWEN	reserved	Fix at 0.	0	-	
		D6 D5 D4	MCBR2 MCBR1 MCBR0	Master clock bit rate (in master mode only)	Master clock divided value = 4×2^{MCBR}	0 0 0	R/W	
		D3	TXDE	Transmit DMA enable	1 Enabled 0 Disabled	0	R/W	
		D2	RXDE	Receive DMA enable	1 Enabled 0 Disabled	0	R/W	
		D1	MODE	SPI mode selection	1 Master 0 Slave	0	R/W	
		D0	ENA	SPI enable	1 Enabled 0 Disabled	0	R/W	

D[31:15] Reserved

D[14:10] BPT[4:0]: Number of Data Bits Per Transfer Setup Bits

Sets the number of bits for transfer data. (default: 0x0)

The value set in this register plus 1 (1 to 32) equals the number of bits transmitted or received in a single transfer.

D9 CPHA: SPI_CLK Phase Select Bit

Selects the phase of the SPI clock. (default: 0)

Sets the data transfer timing in combination with the CPOL (D8). (See Figure 8.1.)

D8 CPOL: SPI_CLK Polarity Select Bit

Selects the polarity of the SPI clock.

1 (R/W): Active Low

0 (R/W): Active High (default)

Sets the data transfer timing in combination with the CPHA (D9). (See Figure 8.1.)

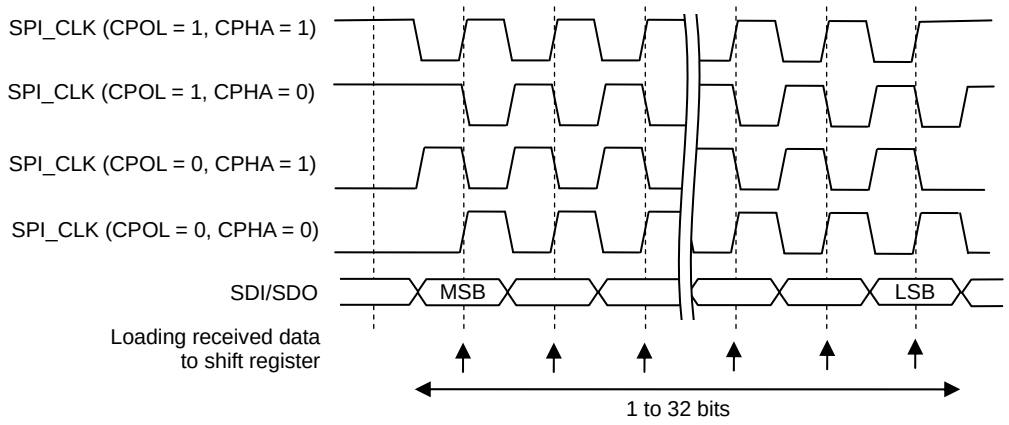


Figure 8.1 Clock and data transfer timing

D7 Reserved (Do not write 1.)

D[6:4] MCBR[2:0]: Master Clock Bit Rate Setup Bits

Sets the division rate for the source clock that generates an SPI clock signal. This setting determines the bit rate.

Table 8.2 Clock frequency setting

MCBR2	MCBR1	MCBR0	Clock frequency (Hz)
1	1	1	MCLK/512
1	1	0	MCLK/256
1	0	1	MCLK/128
1	0	0	MCLK/64
0	1	1	MCLK/32
0	1	0	MCLK/16
0	0	1	MCLK/8
0	0	0	MCLK/4

(default: 0x000)

In slave mode, operations are based on the clock input from the master. This means there is no need to set the bit rate.

D3 TXDE: Transmit DMA Enable Bit

Permits or prohibits transmission DMA interrupt.
1 (R/W): Permit
0 (R/W): Prohibit (default)

When the TXDE is set to 1, output of a transmission DMA interrupt request to the ITC is permitted. A transmission DMA interrupt request is generated by the transfer (the start of transmission) of data written in the SPI Transmit Data Register (0x301704) to the shift register. If the TXDE is set to 1 (permit), the FSPITX (D5/0x300289), the interrupt flag for the ITC, is set to 1 at that time. This interrupt request can also start the HSDMA.
When the TXDE is set to 0, no transmission DAM interrupt is generated.

D2 RXDE: Receive DMA Enable Bit

Permits/prohibits reception DMA interrupt.
1 (R/W): Permit
0 (R/W): Prohibit (default)

When the RXDE is set to 1, output of a reception DMA interrupt request to the ITC is permitted. A reception DMA interrupt request is generated by the transfer (end of reception) of data received in the shift register to the SPI Receive Data Register (0x301700). If the RXDE is set to 1 (permit), the FSPIRX (D4/0x300289), the ITC interrupt flag, is set to 1 at that time. This interrupt request can also start the HSDMA.
When the RXDE is set to 0, no reception DAM interrupt is generated.

D1 MODE: SPI Mode Select Bit

Sets the SPI to master or slave mode.
1 (R/W): Master mode
0 (R/W): Slave mode (default)

Setting MODE to 1 selects master mode; setting MODE to 0 selects slave mode. In master mode, data is transferred based on the clock signal generated in the module. In slave mode, data is transferred based on the clock signal input from the master.

D0 ENA: SPI Enable Bit

Permits/prohibits SPI module operations.
1 (R/W): Permit (On)
0 (R/W): Prohibit (Off) (default)

When the ENA is set to 1, the SPI module begins operating and stands by for a data transfer.
When the ENA is set to 0, the SPI module stops operating.
When setting data transfer conditions, be sure to set the ENA to 0.

0x30170C: SPI Control Register 2 (pSPI_CTL2)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI control register 2 (pSPI_CTL2)	0030170C (W)	D31-12	-	reserved	-	-	-	0 when being read.
		D11	SSA	reserved	Fix at 0.	0	-	
		D10	SS	Slave select control	Fix at 0.	0	R/W	Master mode
					1 SPI select 0 SPI deselect			Slave mode
		D9	SSP	reserved	Fix at 0.	0	-	
		D8	SSC	reserved	Fix at 0.	0	-	
		D7-3	-	reserved	-	-	-	0 when being read.
		D2	RDYP	reserved	Fix at 0.	0	-	
		D1	RDYS	reserved	Fix at 0.	0	-	
		D0	RDYE	reserved	Fix at 0.	0	-	

D[31:11] Reserved (Do not write 1.)

D10 SS: Slave Select Control Bit
Set the slave to a selectable condition.
1 (R/W): Select
0 (R/W): Unselect (default)

Set the SS to 1 before transmitting or receiving data in slave mode. When ENA = 1 and SS = 1, clock input from the master is enabled, allowing transmission and reception in slave mode.
For master mode, set a fixed value of 0 in the SS.

D[9:0] Reserved (Do not write 1.)

0x301710: SPI Wait Register (pSPI_WAIT)

Register name	Address	Bit	Name	Function	Setting	Init.	R/W	Remarks
SPI wait register (pSPI_WAIT)	00301710 (W)	D31 - D0	SPIW31 | SPIW0	Wait cycle control SPIW31 = MSB SPIW0 = LSB	Number of wait cycles = SPIW[31:0] + 1 (1 to 65536)	0x0	R/W	

D[31:0] SPIW[31:0]: Wait Cycle Control Bits

Sets the wait cycle count to be inserted between data transfers. The value set in this register plus 1 becomes the wait cycle count. The range of settings is from 1 to 65536 cycles of the SPI_CLK clock.

0x301714: SPI Status Register (pSPI_STAT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI status register (pSPI_STAT)	00301714 (W)	D31–7	-	reserved	-				-	-	0 when being read.
		D6	BSYF	Transfer busy flag	1	Busy	0	Idle	0	R	Master mode
		D5	MFEF	reserved	-				-	-	0 when being read.
		D4	TDEF	Transmit data empty flag	1	Empty	0	Not empty	1	R	
		D3	RDOF	Receive data overflow flag	1	Occurred	0	Not occurred	0	R	
		D2	RDFF	Receive data full flag	1	Full	0	Not full	0	R	
		D1–0	-	reserved	-				-	-	0 when being read.

D[31:7] Reserved

D6 BSYF: Transfer Busy Flag

Indicates that the SPI is transmitting or receiving. (in master mode)
1 (R): Transmitting or receiving
0 (R): In standby (default)

When the SPI begins transmission/reception in master mode, the BSYE is set to and remains 1 during transmission/reception, including during the wait cycle. On completion of transmission/reception, the BSYE is cleared to 0.
The BSYF is invalid (always 0) in slave mode.

D5 Reserved

D4 TDEF: Transfer Data Empty Flag

Indicates the status of the SPI Transmit Data Register (0x301704).
1 (R): Empty (default)
0 (R): Contains data

The TDEF becomes 0 when transmission data is written to the SPI Transmit Data Register (0x301704), changing to 1 when the data is transferred (the start of transmission) to the shift register.
Transmission data is written when this bit is 1.

D3 RDOF: Receive Data Overflow Flag

Indicates a received data overflow.
1 (R): Overflow generated
0 (R): No overflow (default)

If the subsequent reception process is completed before the data received in the SPI Receive Data Register (0x301700) is read out, the RDOF is set to 1, indicating that the data in the register has been overwritten.
When data in the SPI Receive Data Register (0x301700) is read out, the RDOF reverts to 0.

D2 RDFF: Receive Data Full Flag

Indicates the status of the SPI Receive Data Register (0x301700).
1 (R): Data full
0 (R): No data (default)

When the data received in the shift register is transferred (end of reception) to the SPI Receive Data Register (0x301700), the RDFF becomes 1, indicating that data is ready for readout. It reverts to 0 when the data is read out.

D[1:0] Reserved

0x301718: SPI Interrupt Control Register (pSPI_INT)

Register name	Address	Bit	Name	Function	Setting				Init.	R/W	Remarks
SPI interrupt control register (pSPI_INT)	003017180 (W)	D31–6	-	reserved	-				-	-	0 when being read.
		D5	MFIE	reserved	Fix at 0.				0	-	
		D4	TEIE	Transmit data empty int. enable	1	Enabled	0	Disabled	0	R/W	
		D3	ROIE	Receive overflow interrupt enable	1	Enabled	0	Disabled	0	R/W	
		D2	RFIE	Receive data full interrupt enable	1	Enabled	0	Disabled	0	R/W	
		D1	MIRQ	Manual IRQ set/clear	1	Set	0	Clear	0	R/W	
		D0	IRQE	Interrupt request enable	1	Enabled	0	Disabled	0	R/W	

D[31:5] Reserved (Do not write 1.)

D4 TEIE: Transmit Data Empty Interrupt Enable Bit

Permits/prohibits SPI interrupt due to empty transmission data.
1 (R/W): Permit
0 (R/W): Prohibit (default)

When the TEIE is set to 1, the output of an SPI interrupt request to the ITC due to empty transmission data is permitted. This interrupt request is generated by the transfer (the start of transmission) of data written in the SPI Transmit Data Register (0x301704) to the shift register. If the TEIE and IRQE (D0) have been set to 1 (permit), the FP8 (D0/0x3002A9), the ITC interrupt flag, is set to 1 at that time. When the TEIF is set to 0, no SPI interrupt due to empty transmission data is generated.

D3 ROIE: Receive Data Overflow Interrupt Enable Bit

Permits/prohibits SPI interrupt due to received data overflow.
1 (R/W): Permit
0 (R/W): Prohibit (default)

When the ROIE is set to 1, output of an SPI interrupt request to the ITC due to received data overflow is permitted. This interrupt request is generated when the subsequent received data is loaded before data received in the SPI Receive Data Register (0x301700) is read out. If the ROIE and IRQE (D0) have been set to 1 (permit), the FP8 (D0/0x3002A9), the ITC interrupt flag, is set to 1 at that time. When the ROIE is set to 0, no SPI interrupt due to received data overflow is generated.

D2 RFIE: Receive Data Full Interrupt Enable Bit

Permits/prohibits SPI interrupt due to received data full.
1 (R/W): Permit
0 (R/W): Prohibit (default)

When the RFIE is set to 1, output of an SPI interrupt request to the ITC due to received data full is permitted. This interrupt request is generated by the transfer (end of reception) of data received in the shift register to the SPI Receive Data Register (0x301700). If the RFIE and IRQE (D0) have been set to 1 (permit), the FP8 (D0/0x3002A9), the ITC interrupt flag, is set to 1 at that time.
When the RFIE is set to 0, no SPI interrupt due to received data full is generated.

D1 MIRQ: Manual IRQ Set/Clear Bit

Manually generates an SPI interrupt request to the ITC.
1 (R/W): Set interrupt request
0 (R/W): Clear interrupt request (default)

When the MIRQ is set to 1 while the IRQE (D0) is 1, the SPI interrupt request signal to the ITC becomes active and the FP8 (D0/0x3002A9), the ITC interrupt flag, is set to 1.
When 0 is written to the MIRQ, the interrupt request is cleared. However, writing 0 will not clear the interrupt flag, FP8 (D0/0x3002A9).

D0 IRQE: Interrupt Request Enable Bit

Permits/prohibits the output of an SPI interrupt request to the ITC.
1 (R/W): Permit
0 (R/W): Prohibit (default)

When the IRQE is set to 1, an interrupt request is output to the ITC when a permitted SPI interrupt is generated or when 1 is written to the MIRQ (D1). This sets the interrupt flag, FP8 (D0/0x3002A9), to 1.
When the IRQE is set to 0, no interrupt request to the ITC is generated even if individual SPI interrupts are set to Permit. This also prohibits a manual interrupt request by the MIRQ (D1).

0x30171C: SPI Receive Data Mask Register (pSPI_RXMK)

Register name	Address	Bit	Name	Function	Setting	Init	R/W	Remarks
SPI receive data mask register (pSPI_RXMK)	0030171C (W)	D31-15	-	reserved	-	-	-	0 when being read.
		D14 D13 D12 D11 D10	RXMASK4 RXMASK3 RXMASK2 RXMASK1 RXMASK0	Bit mask for reading received data	0x0 to 0x1F	0 0 0 0 0	R/W	
		D9–2	-	reserved	-	-	-	0 when being read.
		D1	RXME	Receive data mask enable	1 Enabled 0 Disabled	0	R/W	
		D0	-	reserved	-	-	-	Do not write 1.

D[31:15] Reserved

D[14:10] RXMASK[4:0]: Receive Data Mask Setup Bits

Specify the lower-order bit to be enabled when all data must be masked except for necessary lower-order bits in the readout of received data. (default: 0x0)
The value to be set is the MSB of the valid bit (examples: 31 = no mask, 15 = masking of D[31:16]).
To enable this bit mask, the RXME (D1) must be set to 1. If the bit mask is enabled, the received data is read out from the SPI Receive Data Register (0x301700), with the masked bit as 0.

D[9:2] Reserved

D1 RXME: Receive Data Mask Enable Bit

Enables setting of the RXMASK[4:0] (D[14:10]).
1(R/W): Enable
0 (R/W): Disable (default)

When the RXME is set to 1, higher-order bits are masked (0) based on the setting for the RXMASK[4:0] as received data is loaded from the received data buffer to the SPI Receive Data Register (0x301700). When the RXME is set to 0, data other than the data bits (lower-order bits) specified by the BPT[4:0] (D[14:10]/0x301708) are masked (0) during the loading of data received in the shift registered to the SPI Receive Data Register (0x301700).

Figure 8.2 illustrates the relationship between the mask control bit setting and the received data loaded to the SPI Receive Data Register (0x301700).

D0 Reserved

Do not set to 1.

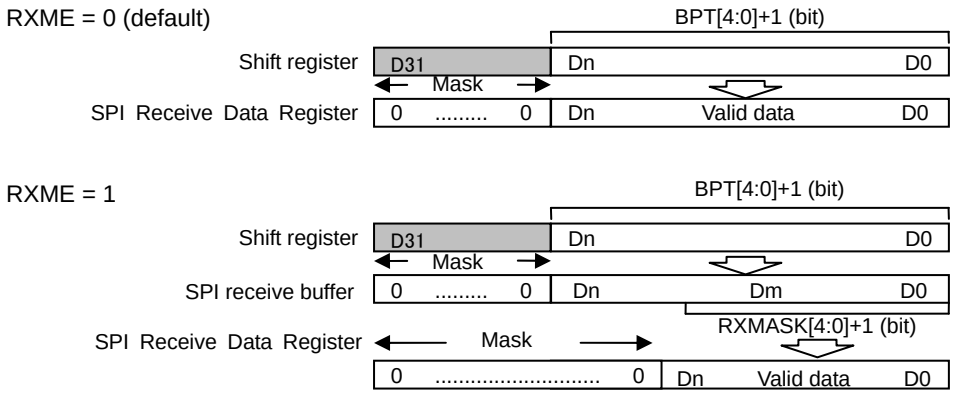


Figure 8.2 Masking the received data

Important Notice

- Always use 32-bit access commands to read/write SPI control registers (0x301700 to 0x30171C). Never use 16-bit or 8-bit access commands.
- Do not access the SPI Control Register 1 (0x301708), SPI Control Register 2 (0x30170C) or SPI Wait Register (0x301710) while the BSYF (D6/0x301714) is 1 (during data transfer operations).

  BSYF: Transfer Busy Flag in the SPI Status Register (D6/0x301714)
- To prevent malfunctions, write 0x0 to the SPI Interrupt Control Register (0x301718) and prohibit all SPI interrupt requests before stopping the SPI circuit (setting the ENA (D0/0x301708) to 0).

  ENA: SPI Enable Bit in the SPI Control Register 1 (D0/0x301708)

9. Quick Start

This section describes the minimum required message flowchart for audio playback using the S1V3034x.

The bold arrows in the diagram indicate the message direction. Right-facing arrows indicate transmission by the host and reception by the S1V3034x. Left-facing arrows indicate transmission by the S1V3034x and reception by the host.

The numbers shown above the bold arrows are the data values sent or received via the serial communication interface. Bytes should be sent or received 1 byte at a time in sequence from the left.

9.1 Streaming Playback

The flowchart in this section shows the minimum required message flowchart for streaming playback. For more information, refer to “4.8 Streamed Playback Messages.”

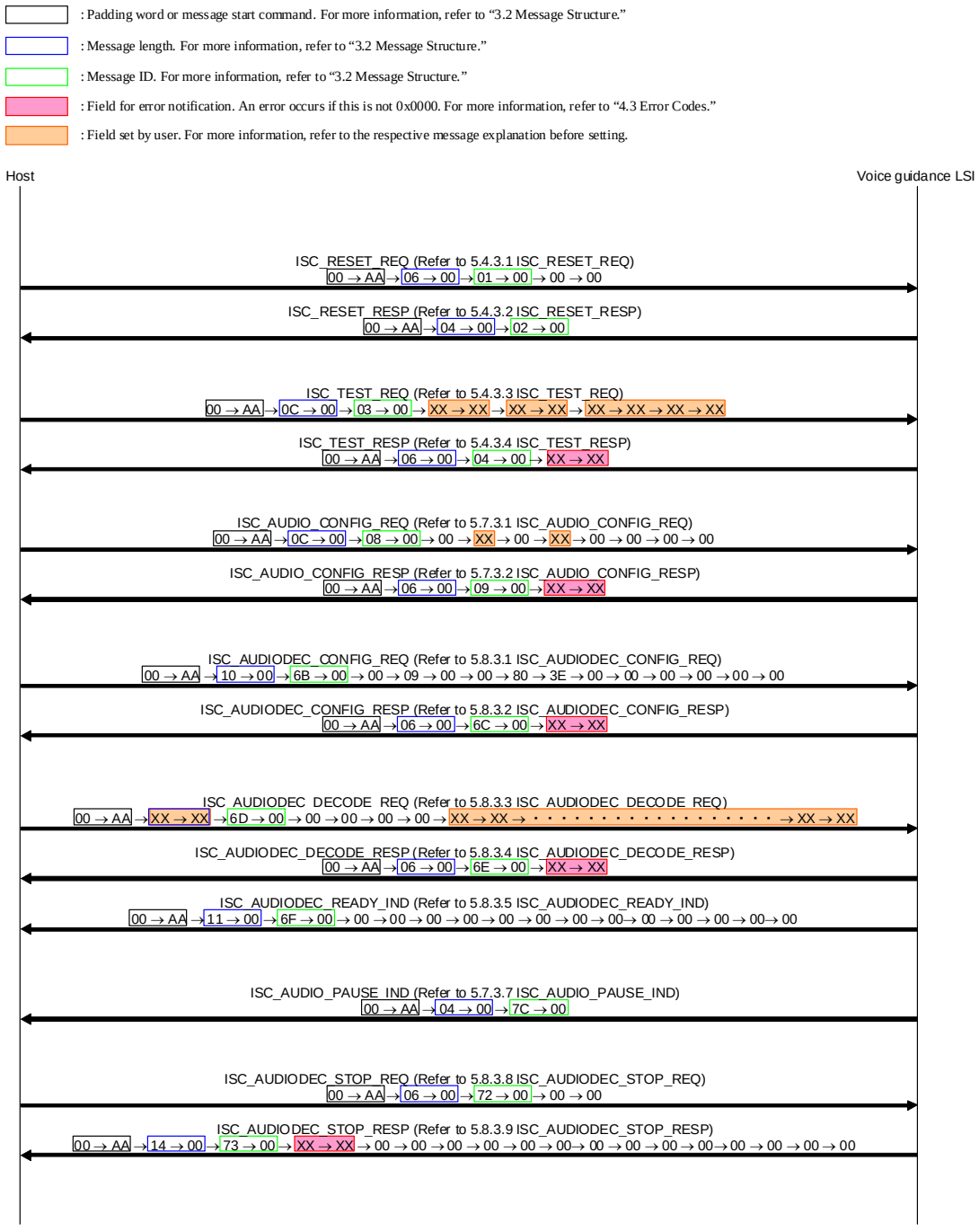


Figure 9.1 Message flowchart for streaming playback

9.2 Sequence Playback

The flowchart in this section shows the minimum required message flowchart for sequence playback. For more information, refer to “4.9 Sequenced Playback Messages.”

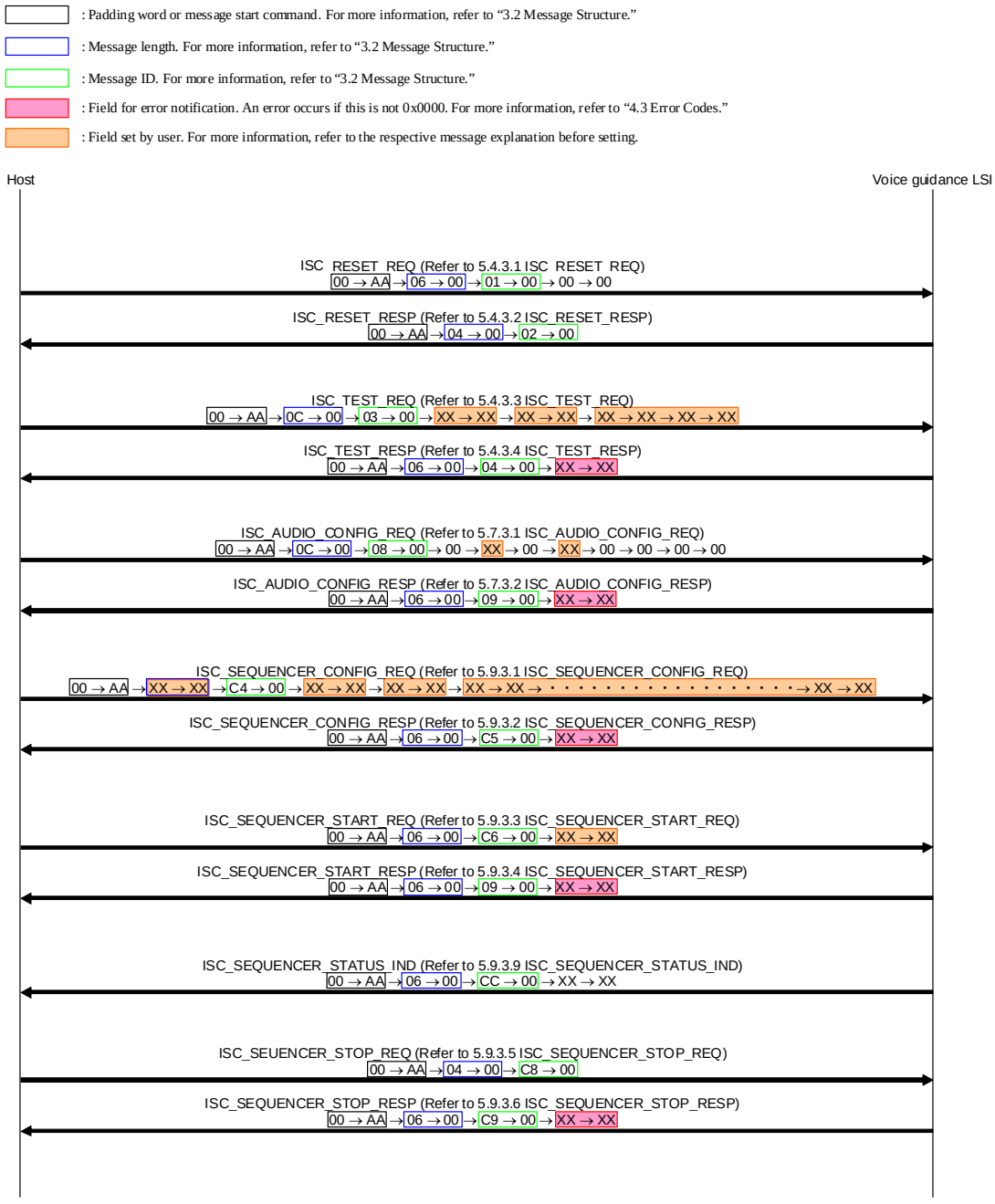


Figure 9.2 Message flowchart for sequence playback

9.3 Immediate Ending of Streaming Playback

The flowchart in this section shows the minimum required message flowchart for immediate ending of streaming playback. ISC_AUDIODEC_STOP_REQ/RESP must be sent and received twice in order to immediately end streaming playback. For more information, refer to “4.8 Streamed Playback Messages.”

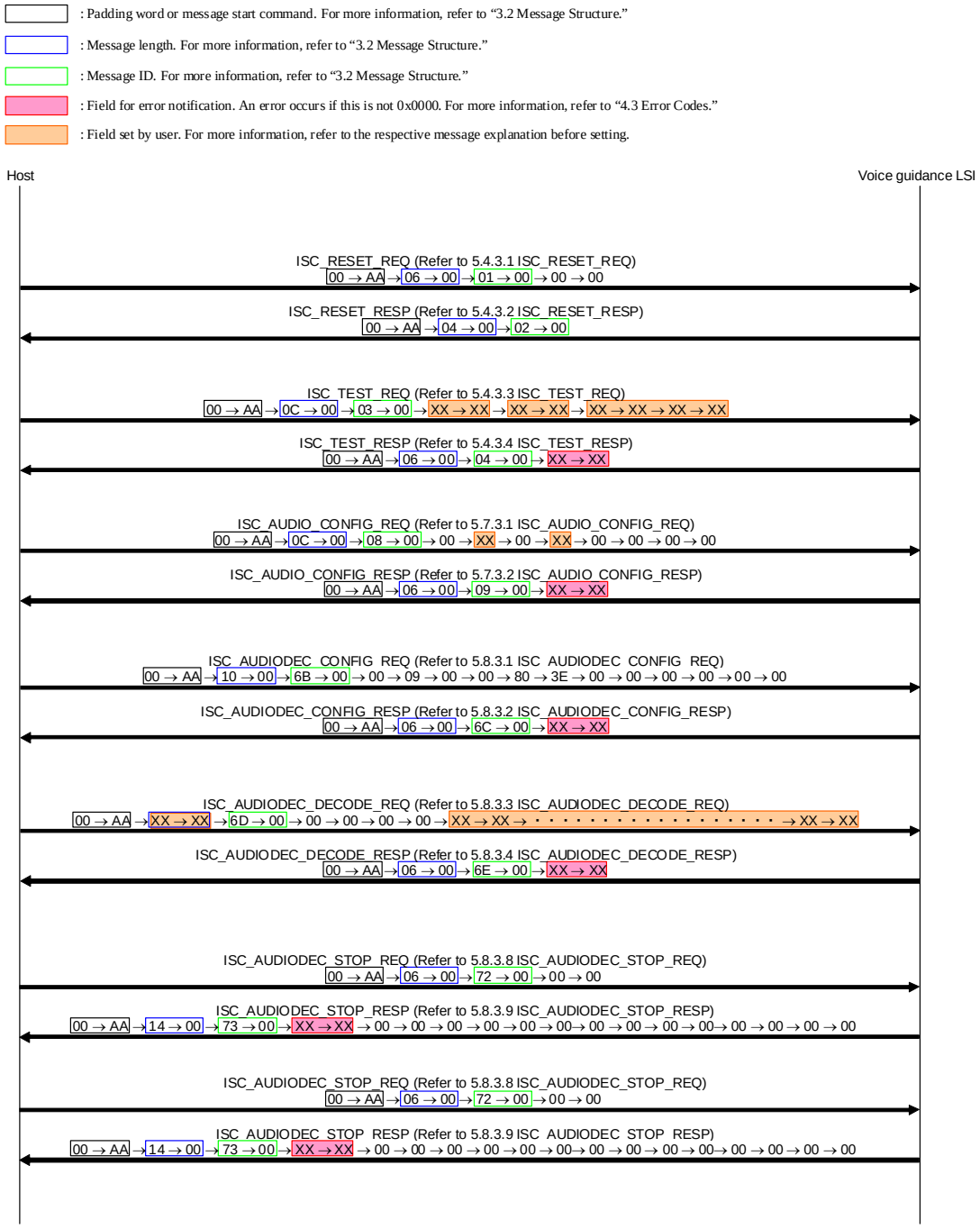


Figure 9.3 Message flowchart for immediate ending of streaming playback

9.4 Immediate Ending of Sequence Playback

The flowchart in this section shows the minimum required message flowchart for immediate ending of sequence playback. ISC_SEQUENCER_STOP_REQ/RESP must be sent and received twice in order to immediately end sequence playback. For more information, refer to “4.9 Sequenced Playback Messages.”

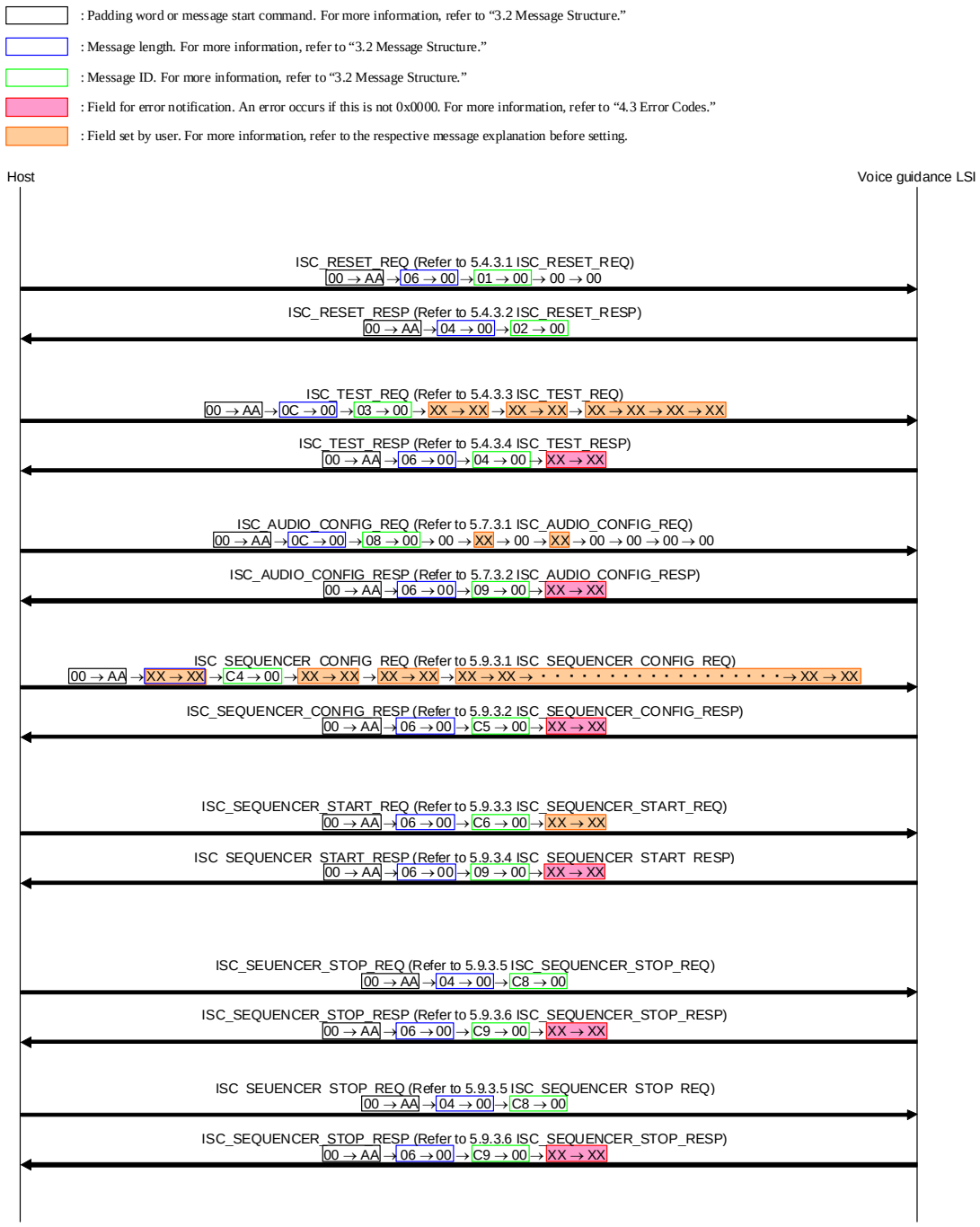


Figure 9.4 Message flowchart for immediate ending of sequence playback

Revision History

Date	Revision details			
	Rev.	Page	Type	Details
07/22/2008	1.00	All	New	Newly established
02/13/2009	1.10	Page 86 Page 88	Addition	Added general-purpose output control message Added flash access message
07/24/2009	1.20	Page 45 Page 55	Correction Correction	4.7.2.3: Corrected Figure 4.11 4.8.2.3: “Streamed Playback End Immediately” Corrected number of transmissions of ISC_AUDIODEC_STOP_REQ to twice. Corrected Figure 4.15
		Page 56	Correction	4.8.2.4: “Streamed playback End” Altered explanation for ISC_AUDIODEC_STOP_REQ. Corrected Figure 4.16
		Page 74	Correction	4.9.2.5: “Sequencer FINALISE IMMEDIATELY” Corrected number of transmissions of ISC_SEQUENCER_STOP_REQ to twice. Corrected Figure 4.25
		Page 75	Correction	4.9.2.6: “Sequencer FINALISE” Altered explanation for ISC_SEQUENCER_STOP_REQ. Corrected Figure 4.26
		Page 76	Addition	4.9.2.8: “Sequenced Playback Message Flow” Added note about non support for repeated playback of multiple phrases
		Page 78	Addition	4.9.3.1: “ISC_SEQUENCER_CONFIG_REQ” Added note about non support for repeated playback of multiple phrases in play_count field
		Page 94	Correction	5.1.2: “Return from Non-fatal Error during Streamed Playback” Corrected number of transmissions of ISC_AUDIODEC_STOP_REQ to twice
		Page 94	Correction	5.1.3: “Return from Non-Fatal Error during Sequenced Playback” Corrected number of transmissions of ISC_SEQUENCER_STOP_REQ to twice
		Page 105	Correction	6: “Use Case of Message Protocol” Corrected No. 7 from repeated playback of multiple phrases to repeated playback of 1 phrase
		Page 106	Correction	Corrected Figure 6.1 heading
		Page 107	Correction	Corrected Figure 6.2. Corrected Figure 6.2 heading
		Page 108	Correction	Corrected Figure 6.3 heading
		Page 109	Correction	Corrected Figure 6.4 heading
		Page 110	Correction	Corrected Figure 6.5 heading
		Page 111	Correction	Corrected Figure 6.6 heading
		Page 112	Correction	Corrected Figure 6.7. Corrected Figure 6.7 heading
		Page 113	Correction	Corrected Figure 6.8. Corrected Figure 6.8 heading
		Page 114	Correction	Corrected Figure 6.9 heading
		Page 115	Correction	Corrected Figure 6.10 heading
		Page 122	Correction	Corrected completion message in (13) to ISC_SEQUENCER_STATUS_IND
		Page 125	Correction	Corrected completion message in (11) to ISC_SEQUENCER_STATUS_IND
		Page 150	Addition	Added 9: “Quick Start”
		Page 151	Addition	Added 9.1
		Page 152	Addition	Added 9.2
		Page 153	Addition	Added 9.3
		Page 154	Addition	Added 9.4

01/26/2010	1.21	Page 39	Correction	t2: Altered explanation for t2. Add t2(CLOCK: 32.768kHz) Max to “800μs”
02/19/2010	1.22	Page 11 Page 26 Page 38 Page 40 Page 47 Page 60 Page 123 Page 151 Page 152 Page 153 Page 154	Correction	Corrected 3.3.3 I2C Communication(IND message sending) Corrected Table 4.6 (Checksum function) as follows 0x0000: corrected “enable” to “disable” 0x0001: corrected “disable” to “enable” Typo corrected. Typo corrected. Typo corrected. Typo corrected. Corrected the MUTE control flow before and after standby mode. Corrected Figure 9.1 Corrected Figure 9.2 Corrected Figure 9.3 Corrected Figure 9.4
10/28/2010	1.23	Page 77 Page 80	Addition Addition	4.9.2.8 “Sequenced Playback Message Flow”: Added Note2 Table 4.37: Added Note 2

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
2580 Orchard Parkway,
San Jose, CA 95131, USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

EUROPE

EPSON EUROPE ELECTRONICS GmbH
Riesstrasse 15, 80992 Munich,
GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.
7F, Jinbao Bldg., No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, CHINA
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH
7F, Block B, Hi-Tech Bldg., 900 Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

SHENZHEN BRANCH
12F, Dawning Mansion, Keji South 12th Road,
Hi-Tech Park, Shenzhen 518057, CHINA
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON HONG KONG LTD.
20/F, Harbour Centre, 25 Harbour Road,
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110, TAIWAN
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.
1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORP.
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong,
Youngdeungpo-Ku, Seoul 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

SEIKO EPSON CORP.
MICRODEVICES OPERATIONS DIVISION

Device Sales & Marketing Dept.
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117

Document Code: 411517305
First Issue September 2008
Revised October 2010 in JAPAN Ⓓ