



LIBRE SYNC

Light Weight Universal Control Interface Technical Note Module : LSx

Revision: 9.9

Libre Wireless Technologies Private Limited

librewireless.com

Copyright © 2017 Libre Wireless Technologies. All rights reserved.

Circuit diagrams and other information relating to Libre Wireless Technologies products are included as a means of illustrating typical applications. Consequently, complete information sufficient for construction purposes is not necessarily given. Although the information has been checked and is believed to be accurate, no responsibility is assumed for inaccuracies. Libre Wireless Technologies reserves the right to make changes to specifications and product descriptions at any time without notice. Contact your local Libre Wireless Technologies sales office to obtain the latest specifications before placing your product order. The provision of this information does not convey to the purchaser of the described semiconductor devices any licenses under any patent rights or other intellectual property rights of Libre Wireless Technologies or others. All sales are expressly conditional on your agreement to the terms and conditions of the most recently dated version of Libre Wireless Technologies standard Terms of Sale Agreement dated before the date of your order (the "Terms of Sale Agreement"). The product may contain design defects or errors known as anomalies which may cause the product's functions to deviate from published specifications. Anomaly sheets are available upon request. Libre Wireless Technologies products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of Libre Wireless Technologies and further testing and/or modification will be fully at the risk of the customer. Copies of this document or other Libre Wireless Technologies literature, as well as the Terms of Sale Agreement, may be obtained by visiting Libre Wireless Technologies website.

LIBRE WIRELESS TECHNOLOGIES DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT AND THE LIKE, AND ANY AND ALL WARRANTIES ARISING FROM ANY COURSE OF DEALING OR USAGE OF TRADE. IN NO EVENT SHALL LIBRE WIRELESS TECHNOLOGIES BE LIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES; OR FOR LOST DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND; REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF LIBRE WIRELESS TECHNOLOGIES OR OTHERS; STRICT LIABILITY; BREACH OF WARRANTY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER OR NOT LIBRE WIRELESS TECHNOLOGIES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES

Table of Contents

1. Document Information.....	10
1.1. Abstract.....	10
1.2. Document Convention.....	10
1.3. Revision History	10
2. Introduction	11
2.1. MCU as A Controller	11
2.2. IR Devices (TBD)	11
2.3. Smart Phone App	12
2.4. Tact Keys Interface.....	12
2.5. Bluetooth.....	12
2.6. Cyclic Redundancy Check	13
3. LUCI Architecture.....	14
3.1. LUCI Framework.....	15
4. LUCI Command and Response Packet.....	17
5. LUCI Interface and Connectivity Details.....	19
5.1. UART Connectivity	19
5.1.1. UART Boot-Up Sequence	20
5.2. TCP Connectivity.....	21
5.2.1. Speaker Discovery	22
5.2.2. Pairing and Authentication Socket (TBD)	27
5.2.3. TCP Command Socket.....	30
5.2.4. TCP Notification Socket	30
6. LUCI Message-Box Commands	31
6.1. Message-Box 1 (Authentication).....	37
6.2. Message-Box 2 (Authentication).....	37
6.3. Message-Box 3 (Register Async).....	37
6.4. Message-Box 4 (Deregister Async)	38

6.5.	Message-Box 5 (Firmware Version Information)	40
6.6.	Message-Box 6 (Host Version Information)	41
6.7.	Message-Box 9 (Is Host Present)	41
6.8.	Message-Box 10 (New / Update Playback Source)	43
6.9.	Message-Box 11 (IsAllowed)	46
6.10.	Message-Box 12 (AirPlay Source Switch Command)	48
6.11.	Message-Box 14 (ACPshareCommand)	51
6.12.	Message-Box 15 (ACPshareResponse)	52
6.13.	Message-Box 20 (Power save: Deep Sleep Start)	54
6.14.	Message-Box 21 (Power save: Deep Sleep End)	54
6.15.	Message-Box 22 (Power save: Standby Start)	55
6.16.	Message-Box 23 (Power save: Standby End)	55
6.17.	Message-Box 24 (Power save: Status)	56
6.18.	Message-Box 36 (Device Detachment Status)	57
6.19.	Message-Box 37 (Graceful Shutdown)	57
6.20.	Message-Box 38 (Device Attachment Status)	60
6.21.	Message-Box 40 (Play Control Commands)	61
6.22.	Message-Box 41 (Browse Control Commands)	63
6.23.	Message-Box 42 (Get UI)	65
6.24.	Message-Box 49 (Current Time)	67
6.25.	Message-Box 50 (Current Source)	68
6.26.	Message-Box 51 (Current Play Status)	70
6.27.	Message-Box 52 (Play List Control Commands)	71
6.28.	Message-Box 53 (Play List view Commands)	74
6.29.	Message-Box 54 (IS-PLAY STATUS)	75
6.30.	Message-Box 55 (Reboot from APP)	76
6.31.	Message-Box 64 (Volume Control)	76
6.32.	Message-Box 65 (Firmware Upgrade Request)	79
6.33.	Message-Box 66 (Firmware Upgrade Progress)	82

6.34.	Message-Box 68 (Host-Image-Present).....	82
6.35.	Message-Box 69 (Request Firmware Upgrade)	83
6.36.	Message-Box 70 (APP CONTROL)	84
6.37.	Message-Box 71 (SD-Card Status).....	88
6.38.	Message-Box 75 (Spotify Preset Actions).....	89
6.39.	Message-Box 80 (Play Audio Index).....	91
6.40.	Message-Box 81 (I2C Client Access).....	92
6.41.	Message-Box 90 (Device Name).....	93
6.42.	Message-Box 91 (Device MACID)	95
6.43.	Message-Box 95 (Aux-Input Start).....	95
6.44.	Message-Box 96 (Aux-Input Stop)	96
6.45.	Message-Box 97 (External PlayBack)	97
6.46.	Message-Box 100 (DDMS Trigger)	98
6.47.	Message-Box 101 (Stand Alone Mode).....	99
6.48.	Message-Box 102 (Search LS-Module)	101
6.49.	Message-Box 103 (Query DDMS).....	103
6.50.	Message-Box 104 (Set Zone-ID)	104
6.51.	Message-Box 105 (DDMS SSID).....	105
6.52.	Message-Box 106 (Speaker Type).....	106
6.53.	Message-Box 107 (Scene Name).....	107
6.54.	Message-Box 108 (Setup Stereo Pair) – LS9 Module Only.....	107
6.55.	Message-Box 111 (TCP/IP Tunneling Start)	109
6.56.	Message-Box 112 (Tunnel Data).....	110
6.57.	Message-Box 113 (Miracast Control).....	112
6.58.	Message-Box 114 (Re-boot Request)	113
6.59.	Message-Box 115 (Reboot Command)	114
6.60.	Message-Box 124 (Network Connection Status)	115
6.61.	Message-Box 125 (Configure Network).....	117
6.62.	Message-Box 126 (i-Devices Wi-Fi Settings Sharing).....	118

6.63.	Message-Box 134 (LINKSTATUS) – LS9 Module Only.....	120
6.64.	Message-Box 140 (WPS Configuration status).....	121
6.65.	Message-Box 141 (WPS Configuration)	121
6.66.	Message-Box 142 (Network Configuration).....	122
6.67.	Message-Box 143 (Network Configuration Status)	126
6.68.	Message-Box 144 (Stop WAC)	127
6.69.	Message-Box 150 (Factory Reset).....	128
6.70.	Message-Box 151 (RSSI Indicator).....	129
6.71.	Message-Box 207 (IOT Control Commands)	129
6.72.	Message-Box 208 (NV Read/Write)	131
6.73.	Message-Box 209 (Bluetooth Control Command)	133
6.74.	Message-Box 210 (DMR restart).....	134
6.75.	Message-Box 211 (Start Firmware Upgrade)	135
6.76.	Message-Box 213 (Name Password Notifier)	141
6.77.	Message-Box 214 (Enable-Share-Mode).....	142
6.78.	Message-Box 215 (Enable-Pair- Mode).....	143
6.79.	Message-Box 216 (SLAVEINFO).....	144
6.80.	Message-Box 219 (Zone Volume Control).....	145
6.81.	Message-Box 221 (Pair Status).....	146
6.82.	Message-Box 222 (Cast OTA Update) - LS9 Module only.....	147
6.83.	Message-Box 223 (Firmware Upgrade INTERNET) – LS9 Module only.....	148
6.84.	Message-Box 224 (Cast is Enabled) - LS9 Module only	149
6.85.	Message-Box 226 (Google Cast Settings Info) – LS9 Modules Only	150
6.86.	Message-Box 227 (DMR STOP).....	152
6.87.	Message-Box 228 (DMR START).....	152
6.88.	Message-Box 229 (GET NTP Time).....	154
6.89.	Message-Box 230 (Audio Output FS) – LS9 Module Only	156
6.90.	Message-Box 231 (gCast Serial Number) – LS9 Module Only	157
6.91.	Message-Box 232 (AC_POWERED) - LS9 Module only.....	158

6.92.	Message-Box 233 (MIC Control)	159
6.93.	Message-Box 234 (AVS_APP_SERVICE).....	160
6.94.	Message-Box 236 (Cloud Tunnelling Request) – LS6 / LS5B Only.....	160
6.95.	Message-Box 237 (Cloud Tunnelling Response) – LS6 / LS5B Only.....	163
6.96.	Message-Box 494 (CAST SETUP STARTED) - LS9 Module only.....	164
6.97.	Message-Box 43690 (0xAAAA) (UART Interface Ready)	164
6.98.	Message-Box 43981 (0xABCD) (LS-IN-BSL Indication).....	165
7.	Appendix	166
7.1.	Acronyms and Abbreviations.....	166

Table of Figures

Figure 3-1: LUCI Customization Sample	14
Figure 3-2: LUCI Frame Work	15
Figure 4-1: LUCI Command and Response Packet	17
Figure 5-2: LS-Module Boot-Up Sequence	21
Figure 5-4 -A : Discovery Sequence using LSSDP	25
Figure 5-4 -B : Discovery Sequence using LSSDP	26
Figure 5-5: Authentication Socket during Authentication	29
Figure 5-6: Authentication Socket Upon Reception of Frame	29
Figure 6-1: Allow – Disallow Sequence Diagram.....	47
Figure 6-2: Source Switch from Airplay to Other Source	49
Figure 6-3: Source Switch from Other Source to Airplay	50
Figure 6-4: LUCI Graceful Shutdown	58
Figure 6-5: Two Way Volume Control Sequence	79
Figure 6-6: LUCI TCP Tunneling Scenario.....	109
Figure 6-7: TCP / IP Tunneling Sequence	111
Figure 6-8 a: SD-Card / USB / Network Method of Firmware upgrade from Web Server	137
Figure 6-8 b: Internet Method of Firmware upgrade from Web Server	138
Figure 6-8 c: SD-Card / USB / Network Method of Firmware upgrade from HOST-MCU	139
Figure 6-8 d: Internet Method of Firmware upgrade from HOST-MCU.....	140

List of Tables



Table 5-1: UART Communication Pins.....	19
Table 5-1: M-Search Packets	23
Table 5-2: LSSDP Packet	23

1. Document Information

1.1. Abstract

Light-weight Universal Control Interface (LUCI) is a protocol, defined to control any LibreSync (LS) enabled device. This document provides LUCI specifications for LibreSync.

1.2. Document Convention

Icon	Meaning	Description
	Note	Provides information good to know
	Caution	Indicates situation that might result in loss of data or hardware damage

1.3. Revision History

Revision	Date	Description of change	Author
9.9	January 11, 2017	Added Message-Box 494	Prem
9.8	December 26, 2016	Added Message-Box 236 and 237	Bharath
9.7	December 22, 2016	Added Message-Box 134 and 108	Lokesh Manu
9.6	December 16, 2016	Updated Message-Box 209	
9.4	December 12, 2016	Updated Message-Box 65 and 69	
9.3	December 9, 2016	Added Message-Box 231	Setu Sharma
9.2	December 7, 2016	Updated Message-Box 41	Sachin

2. Introduction

LibreSync enabled devices provides a set of Control and Status APIs to, remote control itself. LUCI Architecture is designed to enable developers to remote control LibreSync products using a common light weight protocol.

LUCI supports the following Interfaces for Remote Control

- MCU as a Controller
- IR Devices (TBD)
- Smart Phone APP
- Tact Keys Interface
- Bluetooth

2.1. MCU as A Controller

Some products in addition to network apps, may require to connect a HOST-MCU over a hardware interface to LS.

LS provides multiple Interfaces to connect to the MCU

- UART
- TCP

2.2. IR Devices (TBD)

LS extends LUCI framework to support any external Infrared (IR) devices through IR receiver. IR codes are transformed into LUCI architecture by the framework.

2.3. Smart Phone App

Developers can develop custom Applications on Android / IOS or any other smartphone platform to remote control LS speakers.

LUCI provides TCP based interfaces for App developers to send commands and receive status over the network.

LUCI UI for Phone App/ HOST-MCU Display

LUCI provides complete UI Information of the LS speaker in JSON format. This enables App developer to build custom remote UI for the speaker.

Why JSON and not XML?

- JSON is processed more easily, because its structure is simpler.
- JSON is a simpler notation, needs much less specialized software. The very light open code is available for every language.

For more details on JSON see <http://www.json.org/xml.html>

2.4. Tact Keys Interface

Tact keys can be configured over the free GPIOs available with LS-Module. The behavior on a Key press can be defined and mapped to any of the functionalities like Play / Pause, Volume +/-, Next / Prev. LUCI handles the commands received from the keys.



Note:

Tact Key interface behaviour completely depend on the final product spec and availability of free GPIOs on the LS-Module

2.5. Bluetooth

TBD

2.6. Cyclic Redundancy Check

LibreSync supports **CRC-16-CCITT** in the LUCI protocol. Libre releases 'C' source code for calculation of CRC, along with Host simulator reference.

The calculate range of CRC is from Remote-ID till end of the message-data (but excluding CRC [16-bit] field itself), see [Figure 4-1](#).

On receipt of CRC value from HOST-MCU, LibreSync calculates CRC and allows the data to pass only if the HOST-MCU CRC matches with the LibreSync CRC.

Customers who do not prefer to calculate CRC should send CRC value as zero '0'.



Note:

CRC follows 'Big Endian' format in LUCI UART HOST-MCU interface.

3. LUCI Architecture

LUCI makes interacting with LS speakers much easier than before. [Figure 3-1](#) shows how to customize and use LUCI in different scenarios and different product designs.

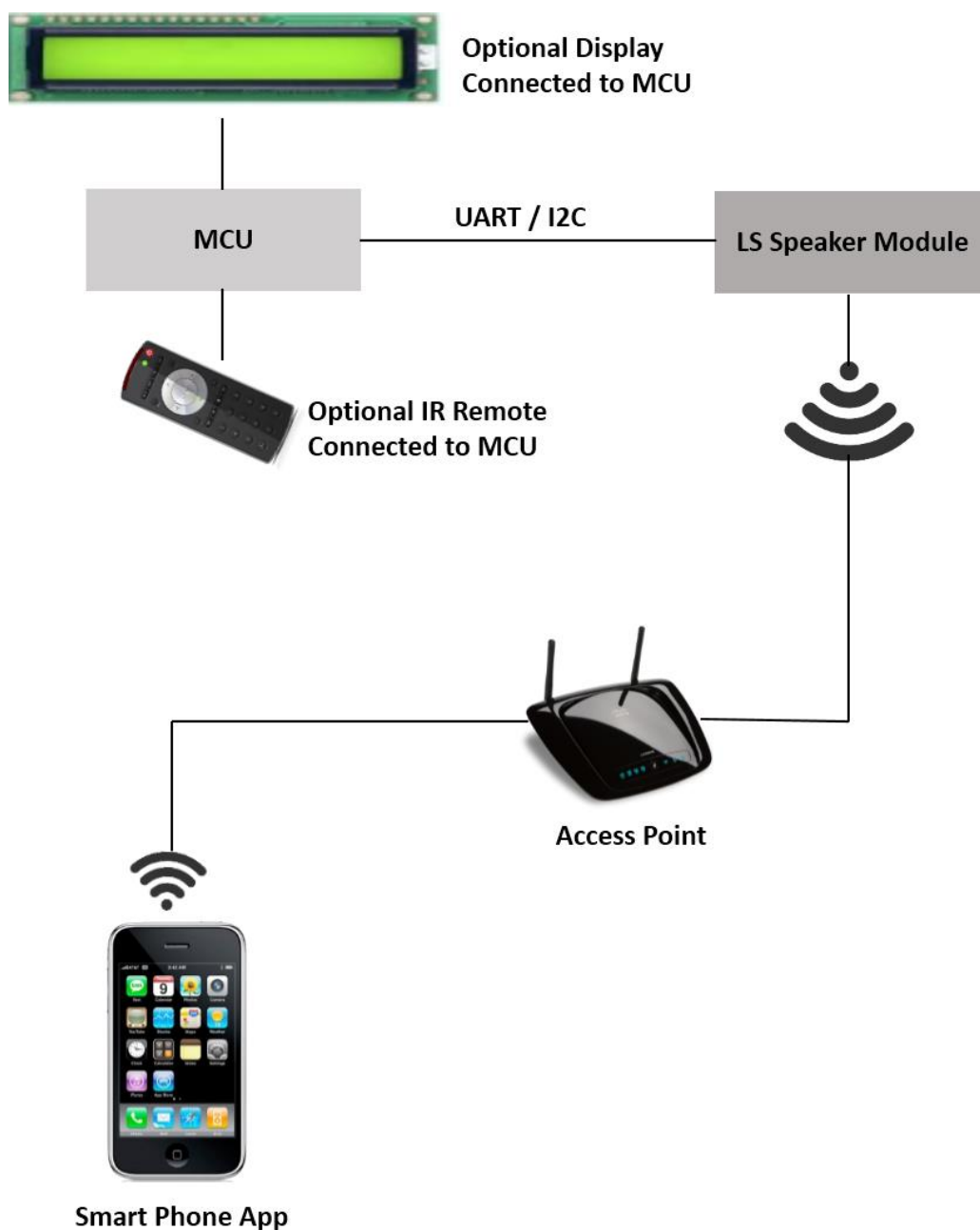


Figure 3-1: LUCI Customization Sample

3.1. LUCI Framework

LUCI framework runs on LibreSync Software stack, based on Linux architecture. [Figure 3-2](#) provides insight to LibreSync Frame work and LUCI.

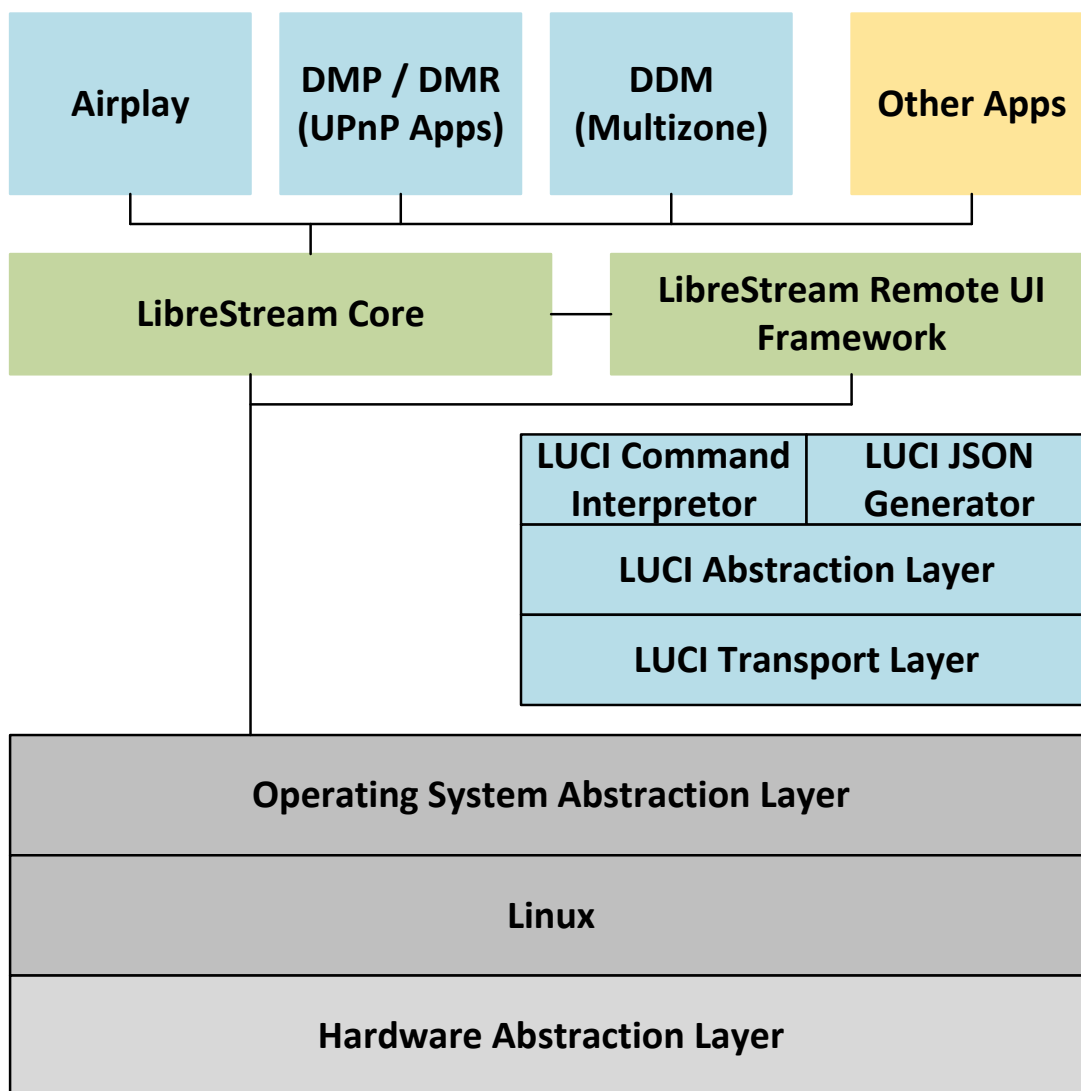


Figure 3-2: LUCI Frame Work

LUCI framework consist of following components

- **LUCI Transport Layer** - Transport Layer is a media dependent layer. It transports the packets to the actual Hardware Interface. Typically LUCI will send the packets generated by the upper layer to all the interfaces available .If an Interface is not available LUCI will skip the Interface.

Similarly, if LUCI receives command packets from any interface it will pass the packets to Hardware abstraction Layer for processing.

- **LUCI Abstraction Layer** - Abstraction Layer provides a media independent interfaces to the upper layers. All events or command-packet received from the interfaces are converted to a common format before processing.
- **LUCI Command Interpreter** – Command Interpreter, interprets the commands received and passes on to the applications as needed. This layer is also used to register specific events by the broadcast receiver.
- **LUCI-JSON Generator** – JSON Generator block hooks into LS Remote UI framework and extracts basic UI information to create LUCI UI representation.

4. LUCI Command and Response Packet

Figure 4-1 provides the LUCI standard command and response packet format.

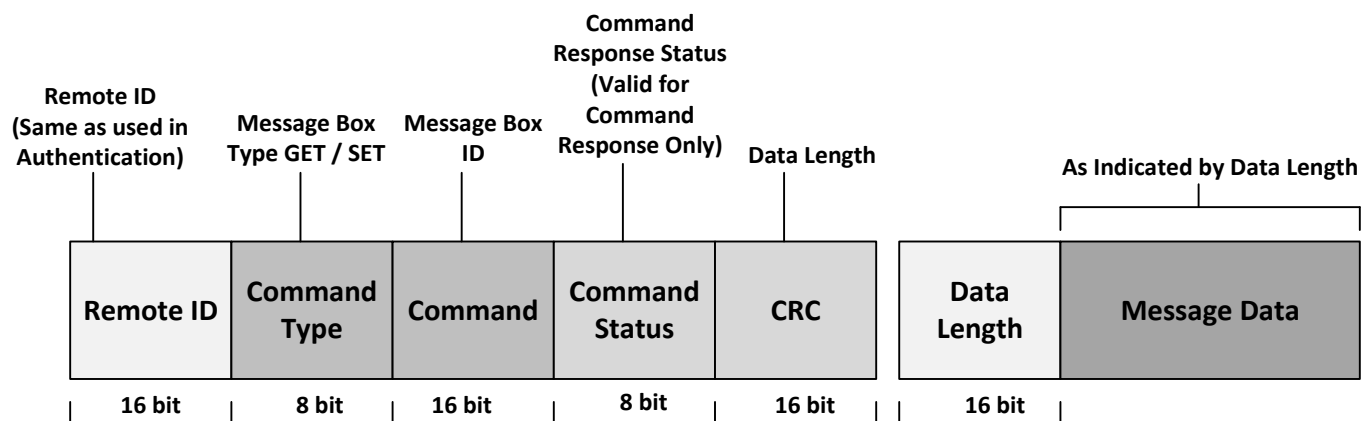


Figure 4-1: LUCI Command and Response Packet

Standard Command	Description
Remote ID	A unique ID used to register the Remote App with LS so that all the remote controller Apps can communicate to LS independently.
Command Type	LUCI provides two command type as GET or SET for any command 00000001: GET(decimal 1) 00000010: SET(decimal 2)
Command	16 bit representation of a command.
Command Status	Valid and non-Zero only for Command response. Only to be sent by LS after receiving a command over the Remote Controller interface. 0: Invalid and can be ignored 1: SUCCESS 2: Generic Error 3: Device not ready for specific command 4: LUCI_CRCERROR

Standard Command	Description
CRC	LibreSync supports CRC-16-CCITT in the LUCI protocol.
Data Length	Length in Bytes of the Data Payload. Can be Zero if no Data is associated
Data	Actual data as specified in the Message-Box. Data field is always a NULL terminated string Type and where ever necessary has to be converted to Integer wherever applicable. If the Data value is mentioned as '1' for any Message-Box, the value to be interpreted in ASCII of it, that is "0x31".

Note:

- Remote ID for UART interface is fixed to **0xAAAA**
- Command Status is only applicable in case Remote Controller Interface initiates a communication over a MessageBox and LS responds with the Command Status back, on the same MessageBox. It is a confirmation that LS has successfully received the Command.
When ever LS initiates a communication (Example – MB #124 [Networ Status]) then Remote Controller Interface can act over it, but sending the response back to LS is not required

5. LUCI Interface and Connectivity Details

This section details the interface and connectivity details.

5.1. UART Connectivity

UART interface is one of the possible interface used by the HOST-MCU to communicate with LibreSync.

Communication lines RX and TX are used to communicate for full duplex communication. The HOST-MCU sends a logic low start bit, a configurable number of data bits (1 Byte in our case), an optional parity bit, and one or more logic high stop bits.



Note: At present, use of CTS / RTS for flow control is optional as we don't see it proving to be much useful for HOST-MCU communication. The maximum achievable speed in this case 345 Kbps.

Table 5-1 details the pins dedicated for UART communication with the HOST-MCU on LS5 and LS6.

Description	Function	Pin
UART_TXD/GPIO	UART_TXD1/GPIO12	7
UART_RXD/GPIO	UART_RXD1/GPIO14	8

Table 5-1: UART Communication Pins

HOST-MCU UART Communication Settings

Serial	Settings
Baud Rate	LS9: 115200 LS6: 57600
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

5.1.1. UART Boot-Up Sequence

For MCU communicating with LS-Module using UART, it's important to understand the boot-up sequence of LS-Module and the readiness of LUCI communication.

The boot-up sequence of LS-Module is as below

Step 1. The HOST-MCU waits for LS-Module to complete the boot-up and initialize all necessary modules and UART connection.

Step 2. Once LS-Module is ready, it writes to Message-Box 43690 with 5 Byte of AA in the data Field.

Example:

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	43690 (0xAAAA)	NA	Variable	Length of Data Field	0xAAAAAAAAAA

Step 3. Once the HOST-MCU receives this message, it should respond back on Message-Box 9.

Figure 5-2 depicts the complete boot-up sequence.

Details of Message-Box 43690 (0xAAAA) and Message-Box 9 can be found later in the document in their respective sections.

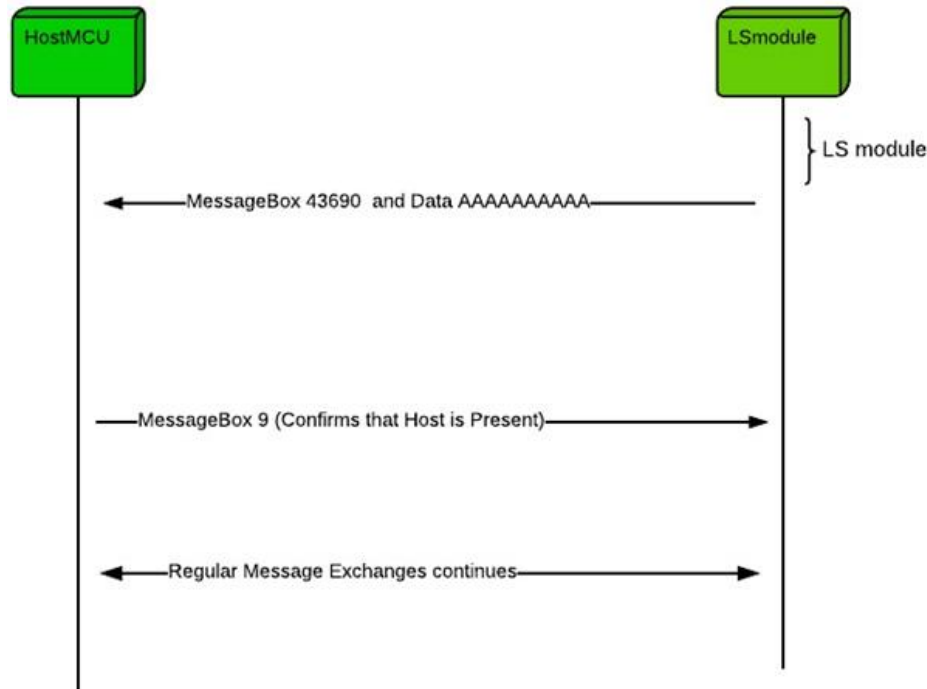


Figure 5-2: LS-Module Boot-Up Sequence

After HOST-MCU confirms on Message-Box#9(Is Host Present) that it is ready to communicate with LS, HOST-MCU has to send its firmware version information over Message-Box#6. Please refer to.

5.2. TCP Connectivity

Transport Control Protocol (TCP) is used in Smart Phone and PC applications. LUCI extends and opens its interfaces to all network apps. Developers can use this to interact with the LS speaker.

The App developers can also use LUCI JSON UI elements, to build their own customized UI for the speaker on the Remote App.

Following interfaces are provided by LUCI for TCPconnectivity

- Speaker Discovery
- Pairing and Authentication Socket

- TCP Command Socket
- TCP Notification Socket

5.2.1. Speaker Discovery

A Smart Phone/ PC App needs to discover a LUCI enabled speaker before it tries to communicate with it.

LUCI provides following discovery protocol for supporting Device discovery

Device Discovery using Lightweight SSDP

As UPNP Simple Service discovery is heavy and involves parsing huge device description files, LUCI supports a light weight SSDP which eliminates the need for parsing XML files.

LSSDP works on the same principle of SSDP.

- Respond to M-Search packets over TCP
- Beacon out NOTIFY as a heartbeat
- Send Bye-Bye before terminating discovery

LSSDP packet are simpler and involves simpler headers.

Table 5-1 and 5-2 are some example packets

M-Search	M-Search Response
M-SEARCH * HTTP/1.1\r\n HOST: 239.255.255.250:1800\r\n\r\n PROTOCOL:Version 1.0	HTTP/1.1 200 OK\r\n HOST: 239.255.255.250:1800\r\n PROTOCOL: Version 1.0 DeviceName: Libre Node XXXXX\r\n DeviceState: M\r\n PORT: 3333\r\n

	ZoneID: XXXX-XXXX-XXXX-XXXX StreamURL: 239:255:255:251:3000\r\n
--	--

Table 5-1: M-Search Packets

LSSDP NOTIFY	LSSDP BYE-BYE
NOTIFY * HTTP/1.1 \r\n	NOTIFY * HTTP/1.1 \r\n
HOST: 239.255.255.250:1800\r\n	HOST: 239.255.255.250:1800\r\n
PROTOCOL: Version 1.0	PROTOCOL: Version 1.0
NTS: ssdp-alive\r\n	NTS: ssdp-byebye\r\n
DeviceName: Libre Node XXXXX\r\n	DeviceName: Libre Node XXXXX\r\n
DeviceState: M\r\n	DeviceState: M\r\n
PORT: 3333\r\n	PORT: 3333\r\n
StreamURL: 239:255:255:251:3000\r\n	StreamURL: 239:255:255:251:3000\r\n

Table 5-2: LSSDP Packet

LSSDP New Headers

The new LSSDP Headers included are as below

- **DeviceName:** This header includes name of the device with the MAC address appended.
- **NTS:** Type of notify message

The following headers are applicable to DDMS functionality only

- **DeviceState:** DeviceState header includes the current state of the device
 - M - The device is the Audio-Master
 - F - Device is in free state and not part of any Zone
 - S - It is an Audio-Client to the Zone mentioned in ZoneID header
- **ZoneID:** ZoneID header sends the current Zone ID. This header is only applicable to state M and S

- **StreamURL** : This contains the IP and port where the Audio-Master will multicast the Audio packets
- **PORT**: The Port which can be used to send unicast control packets to the node.

Figure 5-4 depicts the Discovery sequence between a Smartphone and the 2 nodes.

Device is Up and Smart Phone (APP) is trying to connect to device

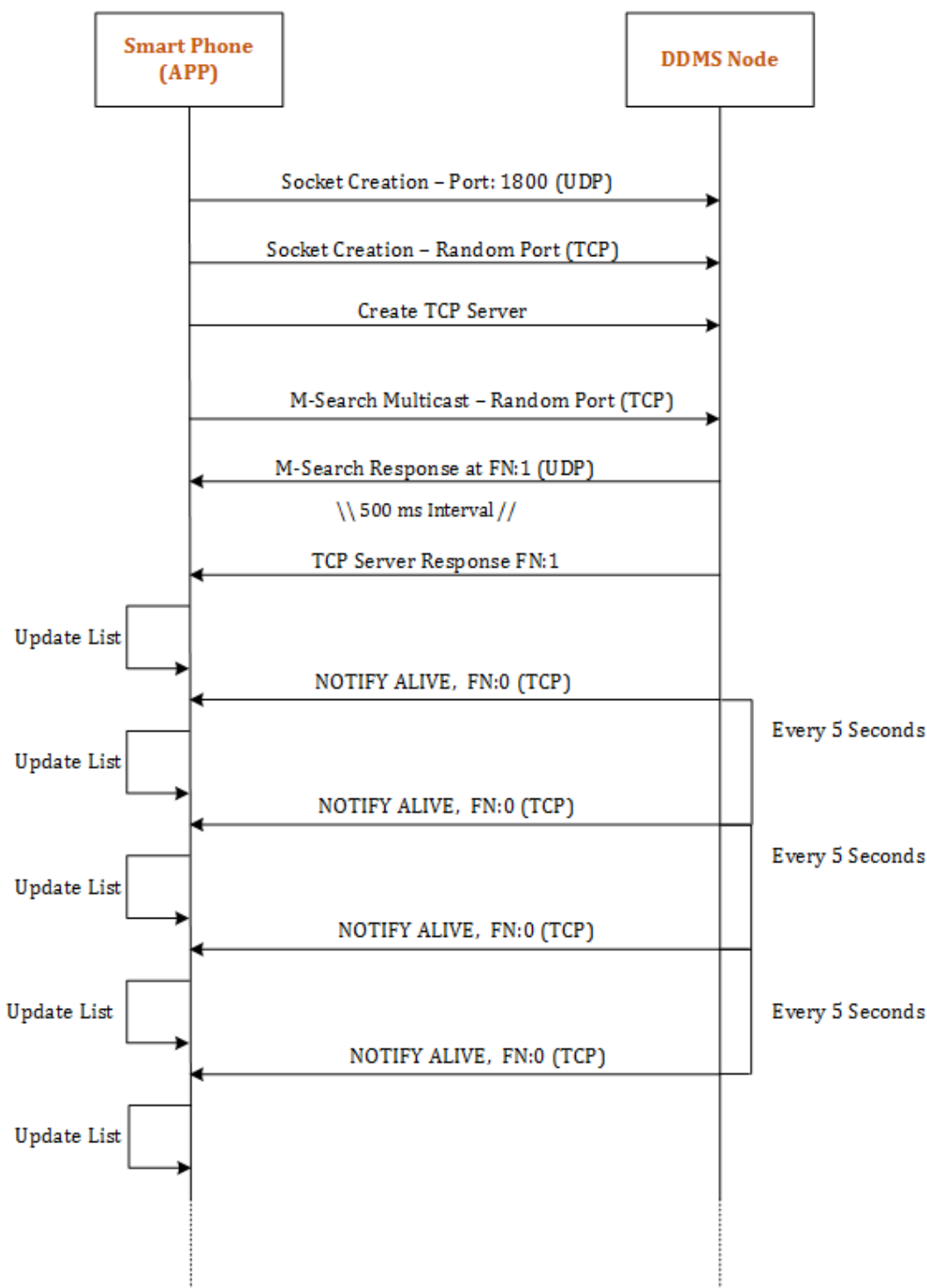


Figure 5-4 -A : Discovery Sequence using LSSDP

Smart Phone (APP) is Up and Device is trying to connect to Device

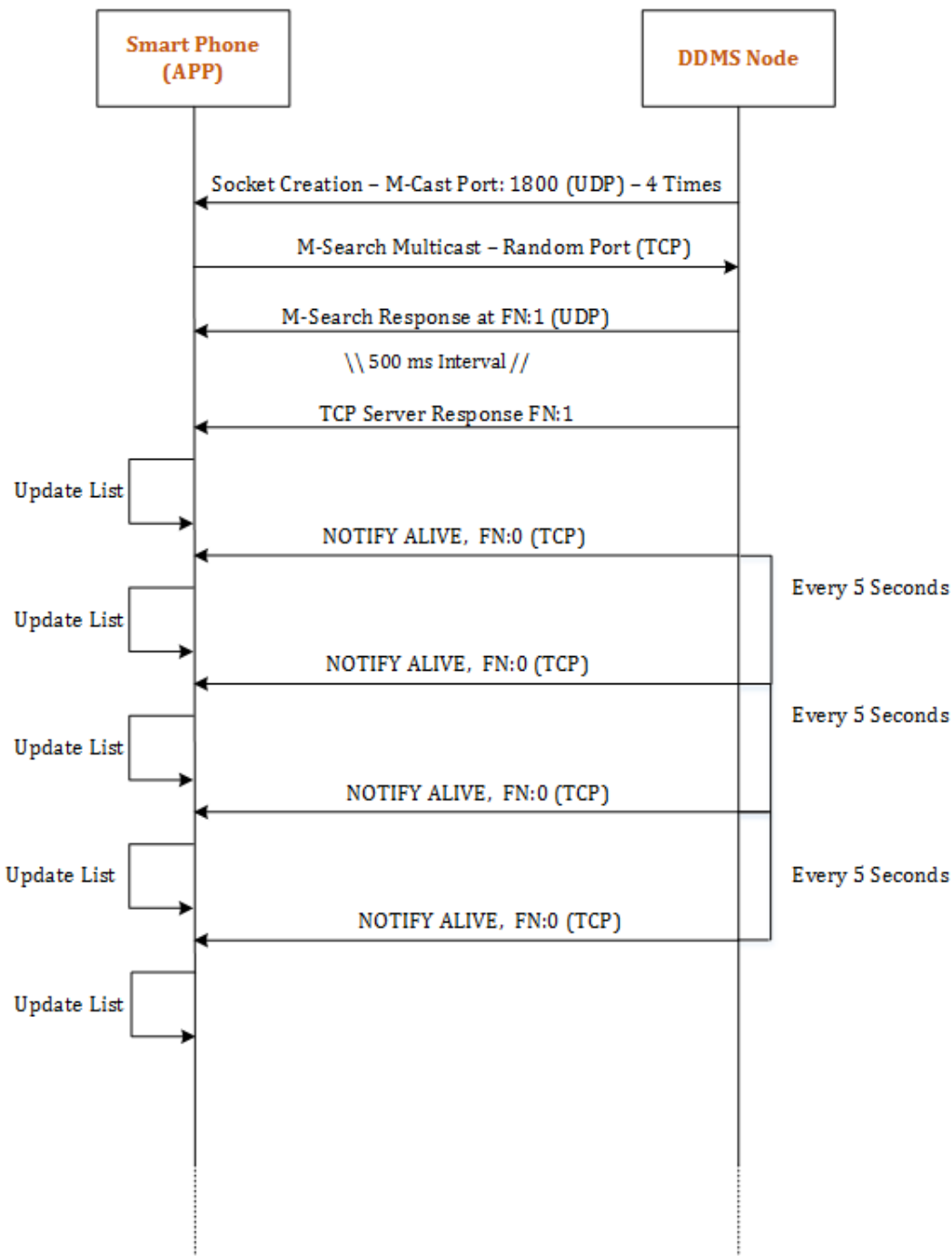


Figure 5-4 -B : Discovery Sequence using LSSDP

If the App developer chooses to discover our devices using Legacy discovery protocol like SSDP or Bonjour, then an additional step of sending a LSSP unicast to the LS device will be necessary.

Steps to discover LUCI enabled devices using Legacy discovery is as below.

- Step 1.** Discover the devices using SSDP or Bonjour
- Step 2.** On discovery send a Unicast LSSDP M-Search to the discovered device on port 1800
- Step 3.** Confirm the response from the device on the same port with LSSDP header

This allows the App to even use Legacy discovery protocol along with LSSDP.

5.2.2. Pairing and Authentication Socket (TBD)



Note: TBD and current release won't require this step

To prevent unauthorized apps to communicate with the speakers and for security reasons, LUCI provides an optional, and simple authentication procedure. Authentication procedure includes a Crypto IC capable of keeping the pre-configured keys in a safe location, and uses the Secure Hash Algorithm (SHA) to scramble the pre-configured keys.

LUCI authentication scheme reduces authentication time and minimize user intervention. Under the LUCI speaker authentication scheme, the paired devices involved in the authentication are pre-configured with the same set of authentication keys used to implement the Cryptographic Hash Algorithm.

During authentication, the device to be authenticated such as a Smart Phone App performs the steps listed below.

- Step 1.** The authenticating device sends a request for authentication.
- Step 2.** Libre Speaker sends Nonce (a pseudo-random number) back to the device.
- Step 3.** The device randomly selects a key from its pre-configured authentication keys.

Step 4. The device now scrambles the selected pre-configured authentication key based on the mutually known scramble algorithm SHA, and generates a signature.

Step 5. Device transmits this signature to the authenticator, such as a speaker which is discovered over a TCPsocket in an appropriate frame.

Upon reception of the frame, the authenticator in turn performs the following steps

Step 1. Descrambles the received authentication key based on the mutually known scrambling algorithm.

Step 2. Checks whether the received authentication key matches one of the pre-configured one.

Step 3. If the authentication is successful, checks if the received authentication key is among the pre-configured authentication keys or unsuccessful otherwise.

Step 4. Communicates authentication status (Success or Failure), in an appropriate frame.

Authentication frame can be sent only on TCP command socket. All other commands will be accessible to the Remote App only after successful completion of the Authentication command.

In the authentication frame sent by the app its mandatory to embed the Remote ID. On successful authentication of the LUCI framework on the LS speaker, Remote ID of the app is registered in its Database.

Only Apps which have successfully performed Authentication are allowed to communicate on the command socket and can register for Asynchronous events.

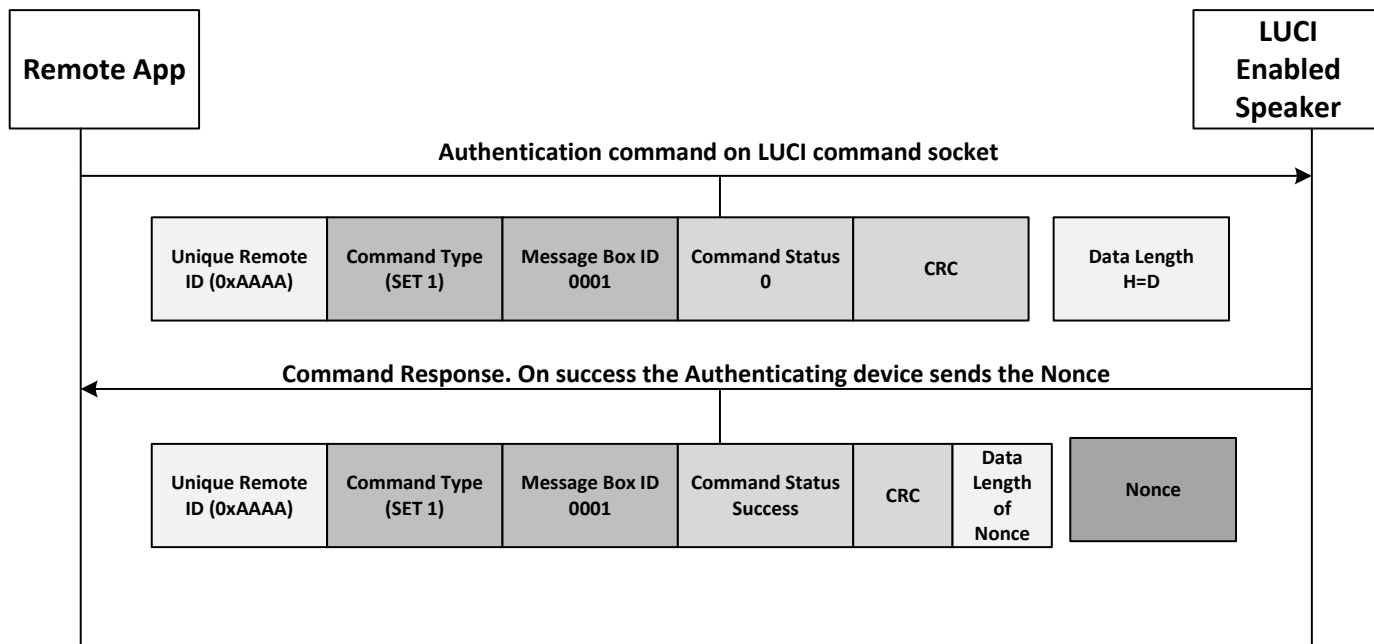


Figure 5-5: Authentication Socket during Authentication

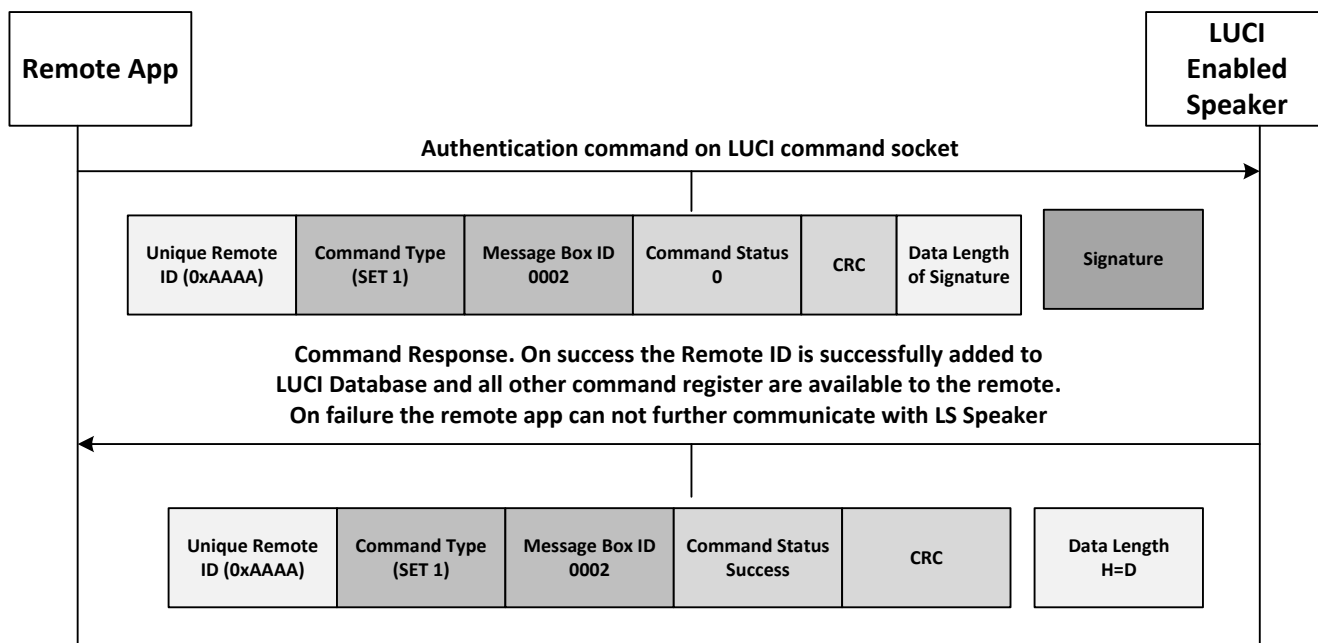


Figure 5-6: Authentication Socket Upon Reception of Frame

Note: Currently LS firmware will skip authentication step. But the App which requires to communicate will still need to register its Remote ID and its IP address with LS-Module using Message-Box 3.

5.2.3. TCP Command Socket

LUCI command socket is a listening socket at port 7777. The LS speakers listens to any valid TCP frames on this socket.

All commands sent to this socket should have a Remote ID, registered during the authentication phase. If Remote ID is not registered, LUCI will ignore the request. Only Authentication command is available to remotes before completion of authentication procedure.

5.2.4. TCP Notification Socket

LUCI provides App Developer to register for Asynchronous events. To use the notification mechanism, the app should register its port number to LUCI on command socket.

On successful registration, LUCI will connect to the socket registered and provide Asynchronous events to the variables registered for.

6. LUCI Message-Box Commands

Message-Box Number (Decimal & Hex)	Direction	Name	Status (Implemented / Reserved/ Not Implemented)	Response required? (Yes / No)
1 (0x1)		Authentication Message-Box	Reserved	No
2 (0x2)		Authentication Message-Box	Reserved	No
3 (0x3)	LSx -->Network Client	Register for Changed Message-Boxes(Async)	Implemented	Yes (From MB 1)
4 (0x4)	LSx -->Network Client	Deregister for Registered Message-Boxes (Async)	Implemented	Yes (From MB 1)
5 (0x5)	HOST-MCU -->LSx	Firmware Version Information	Implemented	No
6 (0x6)	HOST-MCU -->LSx	Host Version Information	Implemented	NO
9 (0x9)	HOST-MCU --> LSx	Is Host Present	Implemented	Yes (From MB 43690)
10 (0xA)	LSx --> HOST-MCU	New/Updated Source	Implemented	Yes (From MB 11)
11 (0xB)	HOST-MCU --> LSx	IsAllowed	Implemented	No
12 (0xC)	HOST-MCU --> LSx	AirPlay	Implemented	
14 (0xE)	Host-MCU --> LSx	ACPshareCommand	Implemented	Yes (From MB 15)
15 (0xF)	LSx --> Host-MCU	ACPshareResponse	Implemented	No
20 (0x14)	HOST-MCU --> LSx	Power Save: Deep Sleep Start	Implemented	No

Message-Box Number (Decimal & Hex)	Direction	Name	Status (Implemented / Reserved/ Not Implemented)	Response required? (Yes / No)
21 (0x15)	HOST-MCU --> LSx	Power Save: Deep Sleep End	Implemented	No
22 (0x16)	HOST-MCU --> LSx	Power Save: Standby Start	Implemented	No
23 (0x17)	HOST-MCU --> LSx	Power Save: Standby End	Implemented	No
24 (0x18)	HOST-MCU --> LSx	Power Save: Status	Implemented	No
36 (0x24)	LSx --> HOST-MCU HOST-MCU --> LSx	Device Disconnected	Implemented	No
37 (0x25)	HOST-MCU --> LSx	LS Shutdown	Implemented	No
38 (0x26)	LSx --> HOST-MCU HOST-MCU --> LSx	Device Attachment Status	Implemented	No
40 (0x28)	HOST-MCU --> LSx Network Client --> LSx	Play Control Commands	Implemented	Yes (From MB 42 - Might / Might Not - Multiple MB)
41 (0x29)	HOST-MCU --> LSx Network Client --> LSx	Browse Control commands	Implemented	Yes (From MB 42 - Might / Might Not - Multiple MB)
42 (0x2A)	LSx --> Network Client LSx --> HOST-MCU	Get UI	Implemented	No
49 (0x31)	LSx --> HOST-MCU	Current Time	Implemented	No
50 (0x32)	LSx --> HOST-MCU HOST-MCU --> LSx	Current Source	Implemented	No
51 (0x33)	LSx --> HOST-MCU	Current Play State	Implemented	No

Message-Box Number (Decimal & Hex)	Direction	Name	Status (Implemented / Reserved/ Not Implemented)	Response required? (Yes / No)
52 (0x34)	Network Client → LSx	Play List Control Commands	Implemented	No
53 (0x35)	LSx --> Network Client	Play List View Commands	Implemented	Yes (From MB 52)
54 (0x36)	LSx --> HOST-MCU LSx --> Network Client	IS-PLAY STATUS	Implemented	No
64 (0x40)	LSx --> HOST-MCU HOST-MCU --> LSx	Volume Control	Implemented	No
65 (0x41)	HOST-MCU --> LS	Firmware Upgrade Request	Implemented	Yes (From MB 43981)
66 (0x42)	LSx --> HOST-MCU	Firmware Upgrade Progress	Implemented	No
68 (0x 44)	LSx --> HOST-MCU	Host-Image-Present	Implemented	No
69 (0x 45)	LSx --> HOST-MCU	Request Firmware Upgrade	Implemented	No
70 (0x46)	Network Client → LSx HOST-MCU → LSx	SD-Card Control Commands	Implemented	For SCANSd, response is sent on MB 42
71 (0x47)	Network Client → LSx HOST-MCU → LSx	SD-Card Status	Implemented	No
75 (0x4B)	LSx --> HOST-MCU HOST-MCU --> LSx	Spotify Preset Actions	Implemented	No
80(0x50)	HOST-MCU --> LSx	Play Audio Index	Implemented	No
81 (0x51)	LSx → HOST-MCU	I2C Client Access	Implemented	NO
90 (0x5A)	LSx --> HOST-MCU HOST-MCU --> LSx	Device Name	Implemented	No

Message-Box Number (Decimal & Hex)	Direction	Name	Status (Implemented / Reserved/ Not Implemented)	Response required? (Yes / No)
91 (0x5B)	HOST-MCU --> LSx	Device MAC ID	Implemented	No
95 (0x5F)	HOST-MCU --> LSx	Aux Start	Implemented	Yes (From MB 11)
96 (0x60)	HOST-MCU --> LSx	Aux Stop	Implemented	No
97 (0x61)	HOST-MCU --> LSx	External Playback	Implemented	No
100 (0x64)	Network Client / HOST-MCU --> LSx	DDMS Trigger	Implemented	No
101 (0x65)	Network Client / HOST-MCU --> LSx	Direct Out of Home Mode(Master mode)	Implemented	No
102 (0x66)	Network Client / HOST-MCU --> LSx	Client mode	Implemented	No
103 (0x67)	LSx --> HOST-MCU LSx --> Network Client	DDMS status	Implemented	No
104 (0x68)	Network Client → LSx	Set Zone ID	Implemented	No
105 (0x69)	LSx --> HOST-MCU LSx --> Network Client	DDMS SSID	Implemented	No
106 (0x6A)	Network Client / HOST-MCU --> LSx	Speaker Type	Implemented	No
108 (0x6C)	HOST-MCU--> LSx LSx --> HOST-MCU	Setup Stereo Pair	Implemented	Yes (MB 108)
111 (0x6F)	HOST-MCU--> LSx	TCP / IP Tunneling Start	Implemented	No
112 (0x70)	HOST-MCU--> LSx	Tunnel Data	Implemented	Yes (From MB111)
113 (0x71)	Network Client / HOST-MCU --> LSx	Miracast Control	Implemented	Yes (From MB 11)

Message-Box Number (Decimal & Hex)	Direction	Name	Status (Implemented / Reserved/ Not Implemented)	Response required? (Yes / No)
114 (0x72)	LSx --> HOST-MCU	Reboot Request	Implemented	No
115 (0x73)	HOST-MCU --> LSx	Reboot Command	Implemented	No
124 (0x7C)	LSx --> HOST-MCU	Connection Status	Implemented	No
125 (0x7D)	HOST-MCU--> LSx	Configure Network	Implemented	No
126 (0x7E)	Network Client / HOST-MCU --> LSx	i-Devices Wi-Fi Settings sharing	Implemented	No
134 (0x86) LS9 Only	HOST-MCU --> LSx	LINKSTATUS	Implemented	No
140 (0x8C)	LSx --> HOST-MCU	WPS Configuration Status	Implemented	No
141 (0x8D)	HOST-MCU--> LSx	WPS Configuration	Implemented	No
142 (0x8E)	HOST-MCU-->LSx	Network Configuration	Implemented	No
143 (0x8F)	LSx --> HOST-MCU	Network Configuration Status	Implemented	Yes (From MB 142)
144 (0x90)	Network Client / HOST-MCU --> LSx	Stop WAC	Implemented	No
150 (0x96)	Network Client / HOST-MCU --> LSx	Factory Reset	Implemented	No
151 (0x97)	Network Client → LSx	RSSI Indicator	Implemented	No
207 (0xCF)	Network Client → LSx	IOT Control Command	Implemented	No
208 (0xD0)	Network Client / HOST-MCU --> LSx	NV Read / Write Command	Implemented	No
209 (0xD1)	Network Client → LSx	Bluetooth Control Command	Implemented	No

Message-Box Number (Decimal & Hex)	Direction	Name	Status (Implemented / Reserved/ Not Implemented)	Response required? (Yes / No)
210 (0xD2)	HOST-MCU → LSx	DMR Restart	Implemented	No
211 (0xD3)	HOST-MCU-->LSx	Start Firmware Upgrade	Implemented	Yes (From MB 69)
213 (0xD5)	Network Client --> LSx	Name Password Notifier	Implemented	No
214 (0xD6)	HOST-MCU --> LSx	Enable-Share-Mode	Implemented	No
215 (0xD7)	HOST-MCU --> LSx	Enable-Pair-Mode	Implemented	No
216 (0xD7)	Network Client --> LSx	SLAVEINFO	Implemented	No
219 (0xDB)	Network Client --> LSx HOST-MCU --> LSx	Zone Volume Control	Implemented	Yes (From MB 64)
221 (0x DD)	LSx --> HOST-MCU	Pair Status	Implemented	No
222 (0xDE)	LSx --> HOST-MCU LSx --> Network Client	Cast OTA Upgrade	Implemented	No
223 (0xDF)	LSx --> Network Client LSx --> HOST-MCU	Firmware Upgrade Internet	Implemented	No
224 (0xE0)	HOST-MCU --> LSx	Cast is Enabled	Implemented	No
226 (0xE2)	LSx --> HOST-MCU LSx --> Network Client	Google Cast Setting Info	Implemented	No
227 (0xE3)	HOST-MCU --> LSx	DMR STOP	Implemented	No
228 (0xE4)	HOST-MCU --> LSx	DMR START	Implemented	No
229 (0xE5)	HOST-MCU --> LSx LSx --> HOST-MCU	GET NTP Time	Implemented	No
230 (0xE6)	LSx --> HOST-MCU	Audio Output FS	Implemented	No

Message-Box Number (Decimal & Hex)	Direction	Name	Status (Implemented / Reserved/ Not Implemented)	Response required? (Yes / No)
(LS9 Only)				
231 (0xE7) (LS9 Only)	Network Client --> LS9 HOST-MCU --> LS9	gCast Serial Number	Implemented	No
232 (0x E8)	LSx --> HOST-MCU	AC_POWERED	Implemented	No
233 (0xE9)	LSx --> HOST-MCU HOST-MCU --> LSx	MIC_CONTROL	Implemented	No
234 (0xEA)	Reserved	AVS_APP_Service	Reserved	No
236 (0xEC)	HOST-MCU --> LSx	Cloud Tunneling Request	Implemented	No
237 (0xED)	LSx --> HOST-MCU	Cloud Tunneling Response	Implemented	No
494 (0x4)	LSx --> HOST-MCU	CAST SETUP STARTED	Implemented	No
43690 (0xAAAA)	LSx --> HOST-MCU	UART Interface Ready	Implemented	Yes (From MB 9)
43981 (0xABCD)	LSx --> HOST-MCU	LS-IN-BSL Indication	Implemented	No

Message-Box 1 and 2 are used for Authentication

6.1. Message-Box 1 (Authentication)

Reserved for Authentication Message-Box

6.2. Message-Box 2 (Authentication)

Reserved for Authentication Message-Box

6.3. Message-Box 3 (Register Async)

Message-Box 3 is the **Register Async** Message-Box. A Remote Client can register or subscribe for any asynchronous Message-Box changes in LibreSync based systems.

TCP based interface

After the authentication is complete on receipt of a successful response for Message-Box 1, it registers for Asynchronous messages providing its port number. Any changes to the Message-Boxes will be sent directly to the network client on the specified port number.

A response to the command can be failure if any of the Message-Box ID requested to be evented asynchronously is not available or cannot be asynchronously evented.

UART

Any changes to the Message-Boxes on LS will be immediately sent over UART interface. So Message-Box#3 is not applicable for UART interface.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	3	NA	Variable	Length of Message	For TCP Interface <IPAddress , port>

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	3	SUCCESS / FAILURE	Variable	0	-

6.4. Message-Box 4 (Deregister Async)

Message-Box 4 is **De-Register N/W Client** Message-Box. This Message-Box is used in case the Remote Client wants to de-register or un-subscribe from LibreSync based systems. Pre-condition is that the N/W client (say a Phone App) should already be registered using

MB#3(Register Async) and establish a session first. MB#4 is then used to disconnect this existing session so that there will no more be any further communication possible with LS. MB#4 has to be used typically when the N/W client is going to close the existing N/W session say when closing the Phone App.

De-registration of individual N/W client:

The N/W client has to send the IP Address of itself as part of LUCI data to de-register that particular N/W client with LS.

A response to this command will be sent to that particular client in case of SUCCESS.

Command from Remote Client (Phone APP)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	4	NA	Variable	Length of Message	<IPAddress>

Response from LS

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	4	SUCCESS / FAILURE	Variable	0	-

De-registration of all the N/W clients connected to LS:

In case there is a requirement to disconnect all the current active sessions of different N/W clients with LS then any of the N/W client can send MB#4 with data set to ALL to de-register/discontinue all the current active sessions.

The response for this command will be sent to all the connected clients in case of SUCCESS.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	4	NA	Variable	Length of Message	ALL

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	4	SUCCESS / FAILURE	Variable	0	-

6.5. Message-Box 5 (Firmware Version Information)

Message-Box 5 is the **Firmware Version Information** Message-Box. This Message-Box can be used to query the version information of LS speaker.

Example of response: <language>.<modulename>.<>.<Release type>.<data>.<Release number>

For example,

REL.LS6.615. 6023.K279.20141216.234713

Command

Unique Remote ID	Command Type	Message -Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	5	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	5	SUCCESS / FAILURE	Variable	Length of Data	Data

6.6. Message-Box 6 (Host Version Information)

Message-Box 6 is the **Host Version Information** Message-Box. This Message-Box is used to send the Host version information of the HOST-MCU to the LS speaker.

HOST-MCU has to send the Host firmware version information in the Data field of Message-Box#6 as a string.

For example,

Cust_name_1.0.1

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	6	NA	Variable	Length of Data	Data

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	6	SUCCESS / FAILURE	Variable	0	-

6.7. Message-Box 9 (Is Host Present)

Message-Box 9 is the **Is Host Present** Message-Box. This Message-Box is used by the HOST-MCU to signal its presence.

After having received the Message over Message-Box#43690 from LS about readiness of LUCI communication over UART, the HOST-MCU signals its presence to LUCI on Message-Box#9

This Message-Box is only applicable for a valid UART interface where LS-HOST-MCU communication is possible.


Command Type value is always SET. Data is sent as String type.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	9	NA	Variable	Length of the data field	Host Available

'HostAvailable' values

value	1
Description	Host is Available

 Note:	If the Data value is mentioned as '1' for any Message-Box, the value to be interpreted in ASCII of it, that is "0x31".
--	---

Response from LS

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	9	SUCCESS / FAILURE	Variable	0	-

6.8. Message-Box 10 (New / Update Playback Source)

Message-Box 10 is the **New / Updated Playback Source** Message-Box.

LS informs HOST-MCU by sending a command with SET Command Type to the HOST-MCU whenever there is a new/changed playback request (Airplay, DMR, USB, SD-Card) received by LS from any of the Media Controller.

HOST-MCU may send a command to LS (Command Type: GET) over Message-Box#50 (Current Source) to get to know the current playback source, if any. HOST-MCU then takes a decision if the new Playback source is allowed or not.

This Message-Box is then used by the HOST-MCU (Command Type: SET) for confirmation of the new source to take over the existing one and is acknowledgement of the same in Message-Box 11.

This Message-Box is applicable only for a valid UART interface where LS-HOST-MCU communication is possible.

Command Type value is always SET. Data is sent as String type.

For Example, from **LS to HOST-MCU**

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	10	NA	Variable	Length of the data field	New / Updated Data

New/Updated Data Values: Data Format

Value	Description	Support in LSx
1	Airplay	Supported
2	DMR	Supported
3	DMP	Supported
4	Spotify	Supported
5	USB	Supported
6	SD-Card	Supported
7	Melon	Supported
8	vTuner	Supported
9	TuneIn	Supported
10	Miracast	Supported
12	DDMS-Slave	Supported
14	Line-IN (Aux-In)	Supported
15	-	-
16	Apple Device (iOS Device over USB)	Supported
17	Direct URL	Supported
18	-	-
19	Bluetooth	Supported
21	Deezer	Supported
22	Tidal	Supported
23	Favorites	Supported
24	Google Cast For Audio	Supported in LS9 Only
25	External Source	Supported

27	Roon Labs	Supported in LS9
----	-----------	------------------

6.9. Message-Box 11 (IsAllowed)

Message-Box 11 is the **IsAllowed** Message-Box. Through this Message-Box the Host controls and confirms whether the new Audio source request received over Message-Box 10, which HOST-MCU has got over Message-Box#10 has to be allowed or not.

Steps involved in confirmation of the new audio source request received over Message-Box # 10 is as below

- Step 1.** LS sends the new source playback request to the HOST-MCU over Message-Box# 10
- Step 2.** HOST-MCU may send a command to LS (Command Type: GET) over Message-Box#50 (Current Source) to know the current playback source, if any.
- Step 3.** HOST-MCU takes a decision whether it has to allow or not the new Audio Source.
- Step 4.** HOST-MCU sends a confirmation on Message-Box#11 with data field set to 1/0 for Allowed/Not-allowed respectively.

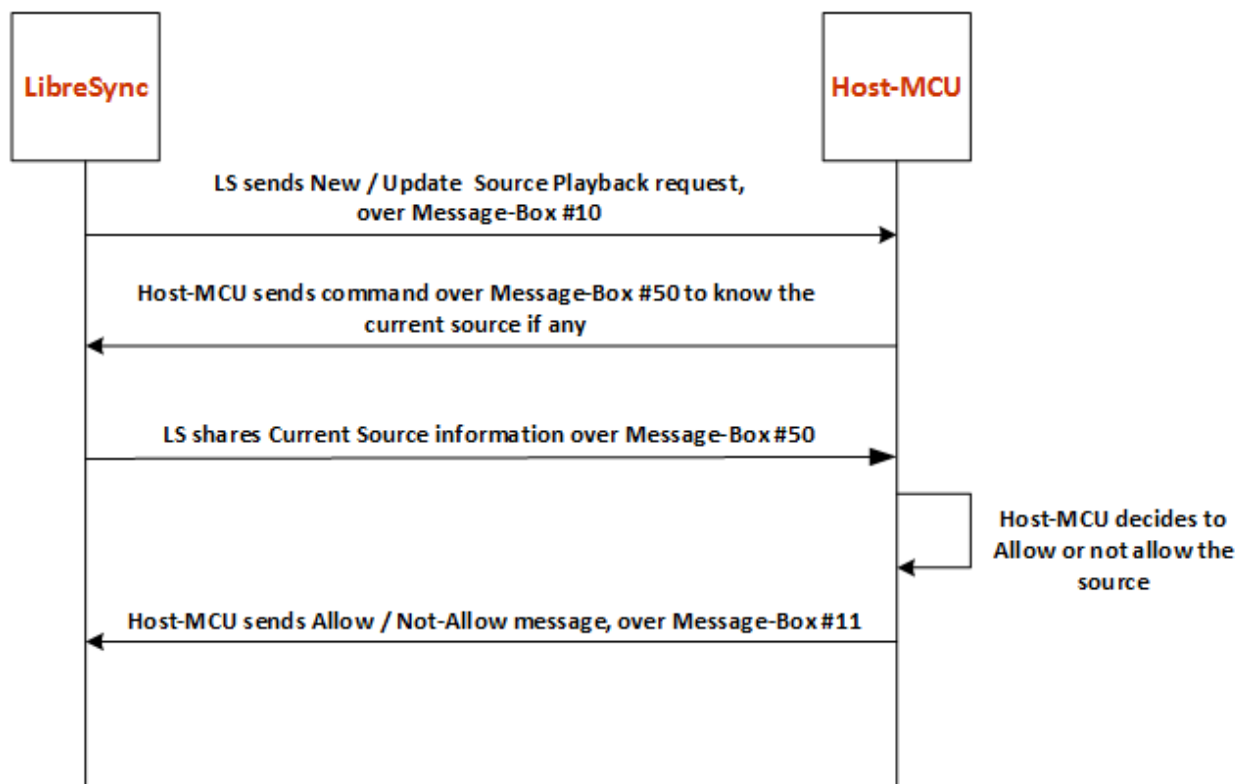


Figure 6-1: Allow – Disallow Sequence Diagram

For Example from **HOST-MCU** to **LS**:

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	11	NA	Variable	Length of the data field	IsAllowed

IsAllowed values:

Data Format

value	0	1
description	New audio source not allowed	New audio source allowed

Response from LS:

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	11	SUCCESS / FAILURE	Variable	0	-

6.10.Message-Box 12 (AirPlay Source Switch Command)

Message-Box 12 is the **AirPlay** Message-Box related to source switch command when Airplay playback is involved.

LS needs to send the status of Airplay to iTunes in case there is a source switching from Airplay to other source or vice versa.

- Source Switch from Airplay to other source:
 - LS sends a command over Message-Box#10 for new/updated source
 - In response, HOST-MCU may send as **0 (New audio source not allowed) or 1 (New audio source allowed)** over Message-Box#11 depending upon whether the playback for the new source has to be allowed or not.
 - In case the data sent is **1**, HOST-MCU sends PREVENT, BUSY, and ALLOW commands respectively to LS in the data field of Message-Box#12.
 - LS events out the respective commands to iTunes.
 - The response for Message-Box#12 will be immediately sent to the HOST-MCU as SUCCESS/FAILURE.

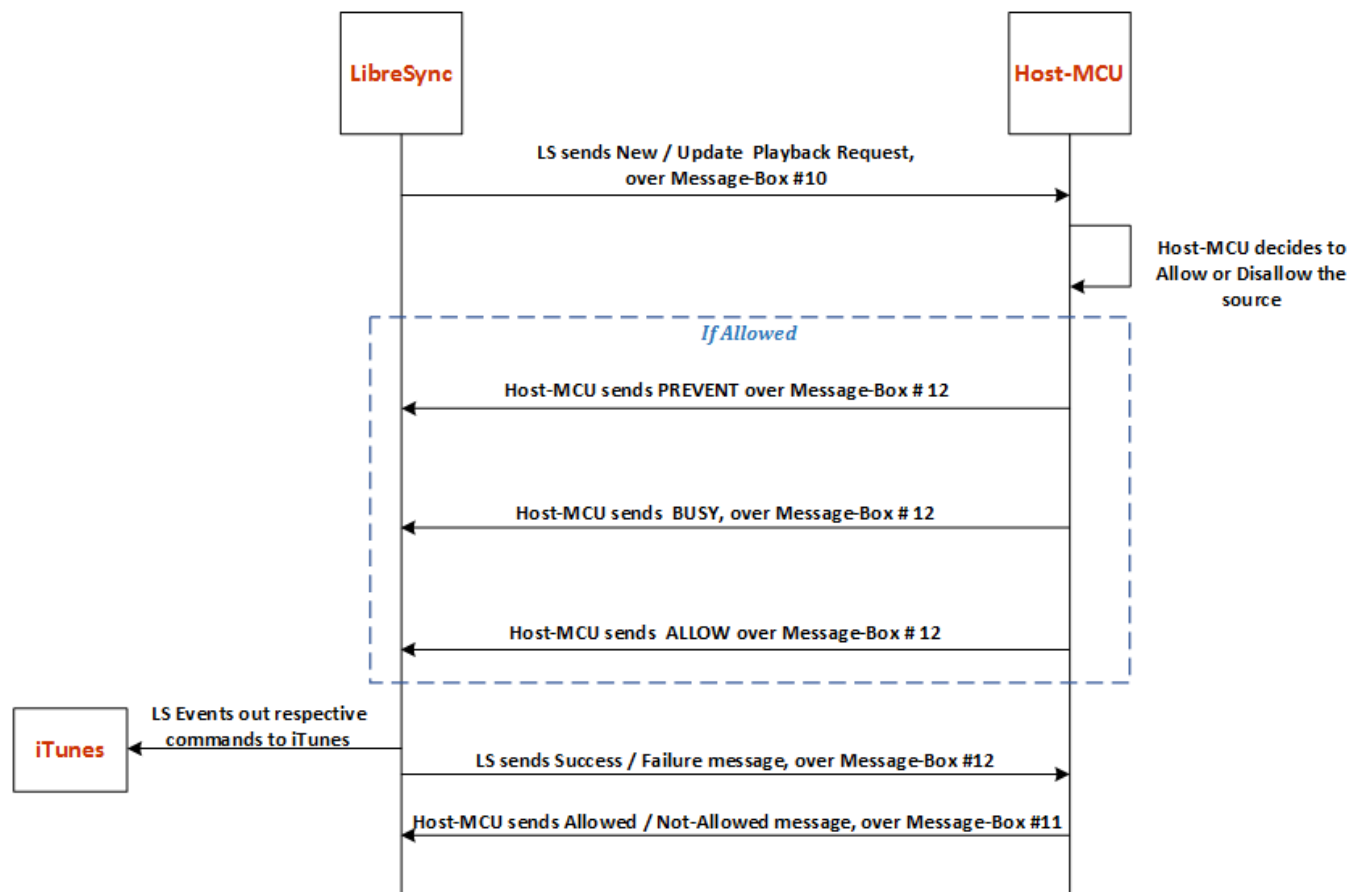


Figure 6-2: Source Switch from Airplay to Other Source

- Source Switch from Other Source to Airplay:
 - LS sends a command over Message-Box#10 for new/updated source
 - In response, HOST-MCU may send ‘is allowed’ as 0 or 1 over Message-Box#11 depending upon whether the playback for the new source has to be allowed or not.
 - HOST-MCU will send AVAILABLE in the data field of Message-Box#12 in case ‘Is Allowed’ is 1 and then LS events out this to iTunes in the similar manner.

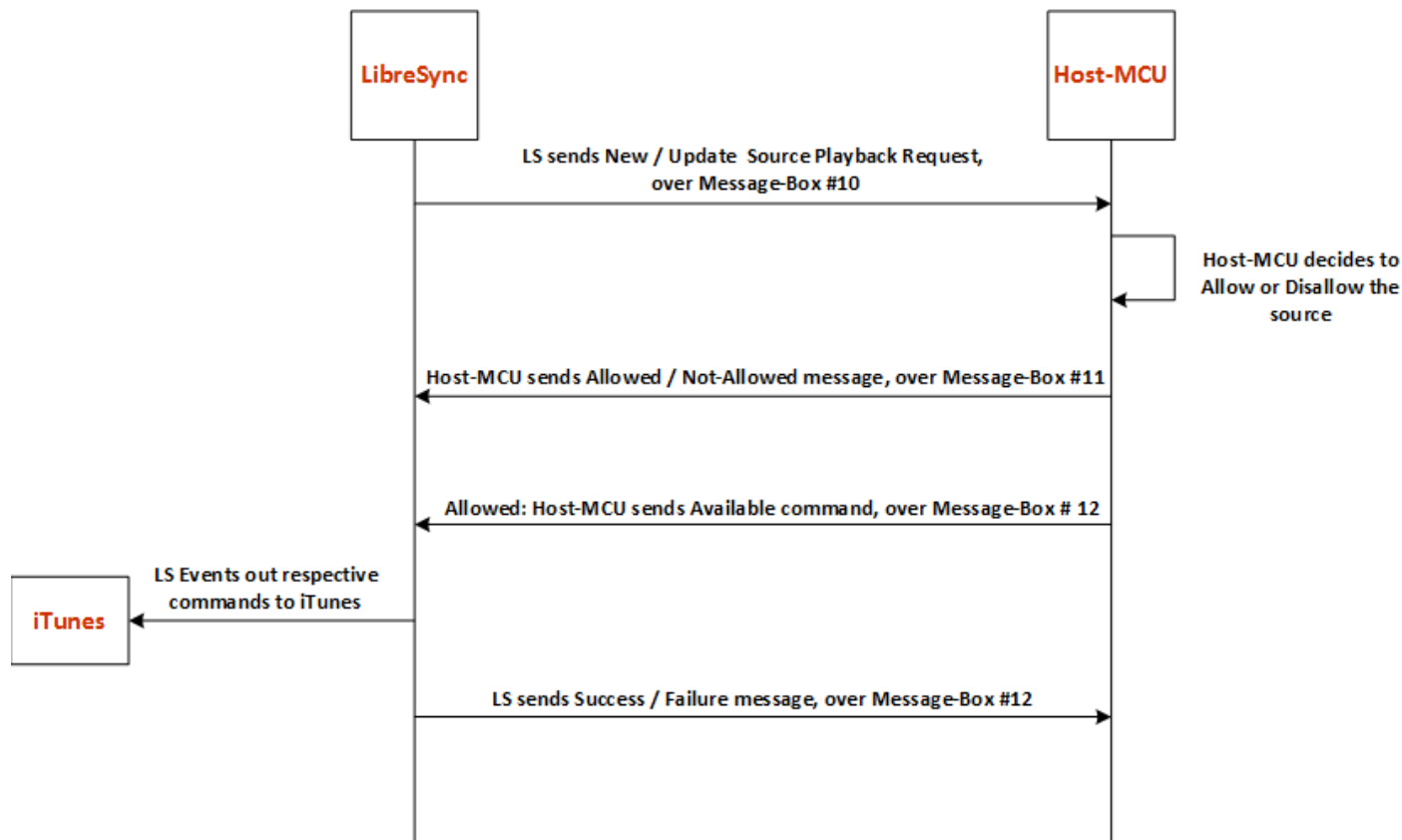


Figure 6-3: Source Switch from Other Source to Airplay

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	12	NA	Variable	Length of the data field	AirPlay Command

Data Format: to be sent in the following sequence in case source switch happens from Airplay to other source and **is allowed** in Message-Box#11 is sent as 1.

Value	PREVENT	BUSY	ALLOW	AVAILABLE
Description	LS events out PREVENT to iTunes	LS events out BUSY to iTunes	LS events out ALLOW to iTunes	LS events out AVAILABLE to iTunes

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	Set	12	Success / Failure	Variable	0	-

6.11. Message-Box 14 (ACPshareCommand)

Message-Box #14 is **ACPshareCommand** Message-Box. Using this Message-box, Host-MCU sends command to LS, requesting 'ACP related data' like ACP certificate, Authentication, Version information and so on.

The LS-Module responds back to Host-MCU in same message box just acknowledging the command. Refer Message-Box #15 for format of actual ACP Data sent by LS-Module.

Command (Host-MCU to LS-Module)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	14	Success / Failure	Variable	Length of Data	Data

Data in String format:

- CERTI
- CHALL, <<ACP-Challenge Data>>
- DEVVE
- FIRVE
- MAJOV
- MINOV
- DEVID

Response (LS-Module to Host-MCU)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	14	Success / Failure	Variable	0	-

Data:

LS-Module gives acknowledgement to Host-MCU on Message-box #14 about reception of command. The data responded in the Message-Box #14 would be NIL.

The actual data of ACP, is responded by LS-Module to Host-MCU is loaded in Message-Box #15.

6.12. Message-Box 15 (ACPshareResponse)

Message-Box #15 is **ACPshareResponse** Message-Box. Based on the command asked by Host-MCU over Message-box #14, the LS-Module responds the required data using Message-Box #15.

Example:

1. Host-MCU passes the 'Challenge' information over Message-Box #14 to LS-Module and requests 'Signature response'. The LS-Module responds back to Host-MCU in Message-Box #15, with the data of 'Signature Response'.
2. Host-MCU asks for 'ACP Device Version' using MB #14 and LS-Module responds with 'ACP Device Version' data on MB #15.

Command (LS-Module to Host-MCU)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	15	Success / Failure	Variable	Variable	ACP-Data

‘ACP-Data’ in String format contains the below info:

- CERTI, << Actual ACP Certificate>>
- CHALL, <<ACP Signature-Response>>
- DEVEE, <<ACP Device Version>>
- FIRVE, <<ACP Firmware Version>>
- MAJOV, <<ACP Authentication Protocol Major Version>>
- MINOV, <<ACP Authentication Protocol Minor Version>>
- DEVID, <<ACP Device ID>>
- ACPNP, <<ACP Not Present>>

Example: CERTI

- If Host-MCU asked (over #14) for certification data, the LS-Module responds (over #15) data starting with a string CERTI and then appends the actual ACP certificate to the data.
- Once the Host-MCU receives the MB #15, Host-MCU can look for starting STRING to identify what data has been sent by LS-Module.



The comma separates the initial string and the actual data.

Example: After the string CERTI, there is a “,” before the actual certification-data passed by LS-Module to Host-MCU on MB #15.

6.13. Message-Box 20 (Power save: Deep Sleep Start)

Message-Box 20 is the **Power save Message-Box** to trigger LS into Deep Sleep state. Most of the peripherals are in standby mode and in this state LS consumes less power so as to keep operating with minimal functionality running. Wake on LAN/WLAN is not supported in this mode.

The status of Power save is sent by LS over Message-Box#24.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	20	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	20	SUCCESS / FAILURE	Variable	0	-

6.14. Message-Box 21 (Power save: Deep Sleep End)

Message-Box 21 is the **Power save Message-Box** to wake up LS out of Deep Sleep state.

The status of Power save is sent by LS over Message-Box#24.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	21	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	21	SUCCESS / FAILURE	Variable	0	-

6.15.Message-Box 22 (Power save: Standby Start)

Message-Box 22 is the **Power save Message-Box**, to trigger LS into standby state. Wake on LAN/WLAN is supported in this mode.

The status of Power save mode is sent by LS over Message-Box#24.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	22	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	22	SUCCESS / FAILURE	Variable	0	-

6.16. Message-Box 23 (Power save: Standby End)

Message-Box 23 is the **Power save Message-Box** to wake up LS out of standby state. The status of Power save mode is sent by LS over Message-Box#24.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	23	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	23	SUCCESS / FAILURE	Variable	0	-

6.17.Message-Box 24 (Power save: Status)

Message-Box 24 is the **Power Save Status** Message-Box. This Message-Box is used by the LS to send the status for the Power Save state to the HOST-MCU.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	24	NA	Variable	Length of data field	power_save_status

power_save_status **values:**

Status	Value
PSM Deep SleepEnd or PSM Network Standby End	0
PSM Deep Sleep Start	1

Status	Value
PSM Network Standby Start	2

6.18. Message-Box 36 (Device Detachment Status)

Message-Box 36 is the **Device Detachment Status** Message-Box. This Message-Box is used by LS to inform HOST-MCU in case any device is disconnected on runtime.

This Message-Box is only applicable only for a valid UART interface where HOST-MCU communication is possible.

Command Type value is always SET when command is sent by LS to HOST-MCU.

Command Type value is always GET when command is sent by HOST-MCU to LS.

Command sent by LS to HOST-MCU with Command Type: SET

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	36	SUCCESS / FAILURE	Variable	Length of Data	Disconnected _Device

Connected _device Values:

Value	1	2	3	4	5	6	7	8
Description	USB	i-Pod	SD-Card	HDMI (TBD)	AUX (TBD)	Reserved	Reserved	Reserved

6.19. Message-Box 37 (Graceful Shutdown)

Message-Box 37 is the **LS Graceful Shutdown** Message-Box. This Message-Box is used by the HOST-MCU to gracefully shutdown LS.

This Message-Box can be SET by HOST-MCU to gracefully shutdown LS module.

Once this command is received LS module will de-register itself from Bonjour, SSDP and LUCI LSSDP.

LS will also complete critical sections of work like saving a file etc .

Once completed LS will respond back on the Same Message-Box on the final Status.

HOST-MCU has to discontinue the supply (like 3v3 supply in case of LS-6) to LS module after receiving this command. It returns Fail in case LS can-not be shut down at this moment.

Below is the sequence of Steps involved.

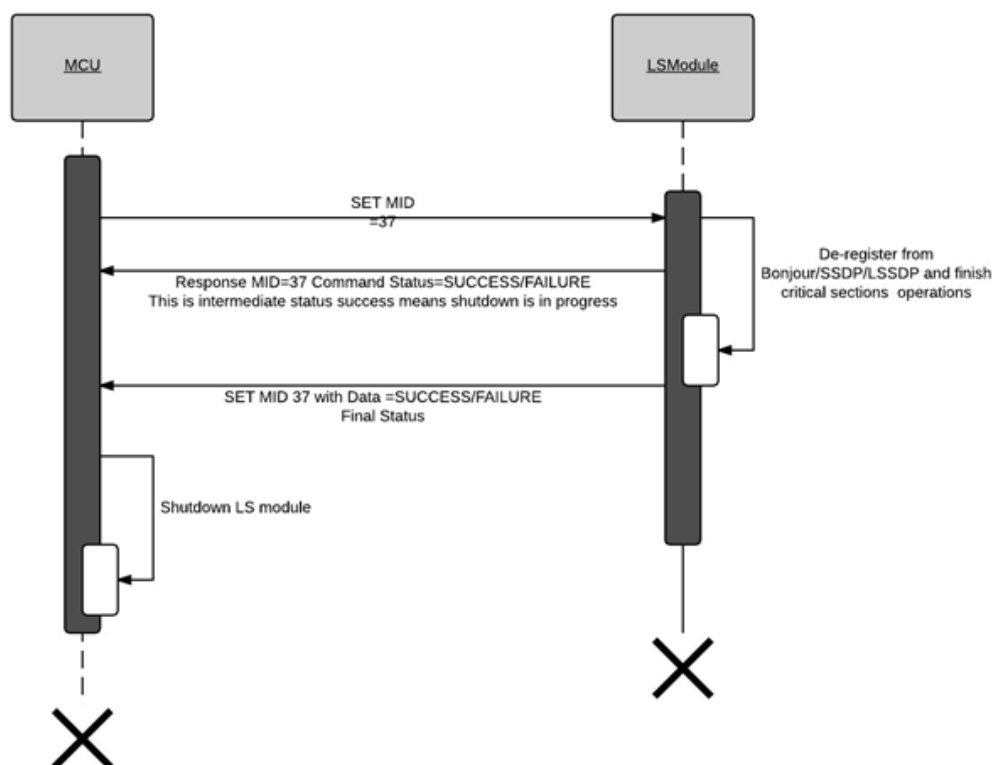



Figure 6-4: LUCI Graceful Shutdown

Command (HOST->LS)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	37	NA	Variable	0	-

Response (LS->HOST)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	37	SUCCESS/FAILURE	Variable	0	-



Note: This is intermediate status and success means shutdown is in progress

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	37	NA	Variable	Length of Data	-

Where Data can be

- 0: Invalid and can be ignored
- 1: SUCCESS
- 2: Generic Error
- 3: Device not ready for specific command

6.20.Message-Box 38 (Device Attachment Status)



LS first sends response to Message-Box#37 with SUCCESS to acknowledge that it has received a shutdown command. HOST-MCU has to wait for receiving another response with data as (SUCCESS/FAILURE) on Message-Box#37 indicating that LS has actually deregistered from Bonjour/LSSDP and done with all the critical operations.

Message-Box 38 is the **Device Attachment Status** Message-Box. This Message-Box is used by LS to inform HOST-MCU in case any device is connected on runtime.

This Message-Box is only applicable only for a valid UART interface where HOST-MCU communication is possible.

Command Type value is always SET when command sent by LS to HOST-MCU.

Command Type value is always GET when command sent by HOST-MCU to LS.

Command sent by LS to HOST-MCU with Command Type: SET

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	38	SUCCESS / FAILURE	Variable	Length of Data	Connected _device

Connected _device Values:

value	1	2	3	4	5	6	7	8
Description	USB	iPod	SD-Card	HDMI (TBD)	AUX (TBD)	Reserved	Reserved	Reserved

6.21.Message-Box 40 (Play Control Commands)

Message-Box 40 is used for **Play Control Commands**. The Play control commands can be used by any of the available interface controller (Example HOST-MCU over UART or Phone App over Network) to send Play, Pause, Stop, Next, Previous, Repeat and Shuffle commands to the current selected source.

Command Type value is always **SET** when command sent by the available interface Controller to LS


For Example,

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	40	NA	Variable	Length of data field	PlayControlKeyCode

HOST-MCU or Remote App should send the below commands as a string for the particular selected functionality.

PlayControlKeyCode	Description	Use case
PLAY	Start Playback with URL	PLAY
PAUSE	Pause Playback	PAUSE
STOP	Stop Playback	STOP
RESUME	Resume playback after PAUSE	RESUME
NEXT	Play Next song	NEXT
PREV	Play Previous song	PREV
MUTE	Mute playback volume	MUTE
UNMUTE	Unmute playback volume	UNMUTE

PlayControlKeyCode	Description	Use case
SEEK	Seek to a location while song is playing	SEEK: time in milliseconds
REPEAT	Loop play back over a song Turn off repeat Loop playback over playlist	REPEAT:ONE REPEAT:OFF REPEAT:ALL
SHUFFLE	Turn ON shuffle mode Turn OFF shuffle mode	SHUFFLE:ON SHUFFLE:OFF
TOGGLEFAV	Toggles favorite state of current song. That is, it adds or removes the song from favorite playlist.	TOGGLEFAV
SHUFFLETOGGLE	Turn ON shuffle mode in Airplay	SHUFFLETOGGLE
REPEATTOGGLE	<p>Repeats One Song / Playlist in Airplay. It can also be used to turn Repeat OFF in Airplay.</p> <div>  <p>Note: The repeat toggle behavior is different in iTunes and in other iOS Devices. Repeat Toggle functionality for different iOS sources should be taken care by HOST-MCU. Behavior: iTunes - Repeat OFF state > ALL > ONE > OFF.</p> </div>	REPEATTOGGLE

PlayControlKeyCode	Description	Use case
	iOS Device - Repeat OFF state > songs/one > ALL/Playlist/Album/Artist > OFF	
FFSTART	Begins Fast Forward of the playback in BT Source	FFSTART
FFRELEASE	Stops the Fast Forward of the playback in BT Source	FFRELEASE
FRSTART	Begins Fast Rewind of the playback in BT Source	FRSTART
FRRELEASE	Stops the Fast Rewind of the playback in BT Source	FRRELEASE

Response Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	40	SUCCESS / FAILURE	Variable	0	-

6.22.Message-Box 41 (Browse Control Commands)

Message-Box 41 is used for **Browse Control commands**. The Browse control commands can be used by any of the available interface controller (e.g. HOST-MCU over UART or Phone App over Network) to control the selection of Enter, Back, Up, Down and Select Item Commands.

Command Type value is always SET when command sent by the available interface Controller to LS

Command

Unique Remote ID	Command Type	Message -Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	41	NA	Variable	Length of data field	BrowseControlKeyCode

The HOST-MCU or Remote APP should send the below commands as a string for the particular selected functionality.

Valid **BrowseControlKeyCode** values for a LibreSync device is as below.

- ENTER
- BACK
- UP
- DOWN
- ONETOUCH
- SCROLLUP
- SCROLLEDOWN
- SELECTITEM:index
select a particular item, for example SELECTITEM:10
- PLAYITEM: HTTP URL of The Song
 - For Melon Playback - Append **MELON:** in the beginning of the URL of the Song
 - Example, PLAYITEM:MELON:<<HTTP URL of The Song>>

- For Direct URL Playback - Append **DIRECT:** in the beginning of the URL of the Song
 - Example, PLAYITEM:DIRECT:<<HTTP URL of The Song>>
- GETUI - To Get Current View from the Host-MCU / APP
 - For Play view Host-MCU / APP should send command GETUI: PLAY.
 - For Home view Host-MCU / APP should send command GETUI: HOME.

Response:

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	41	SUCCESS / FAILURE	Variable	0	-

6.23.Message-Box 42 (Get UI)

Message-Box 42 is the **Get UI** Message-Box. This Message-Box is updated with the current active View and all its elements in LS speaker.

For Mobile Apps the Message-Box will be directly sent to the App specified port. The UI information is embedded in JSON format.

At present LS supports only two views that is Browse View and Play view. The concept of GETUI Message-Box can be extended for different views using the same principle.



To use this Message-Box make sure you have set the NV-Item HostUiEnabled as 1.

Command


Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	42	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	42	SUCCESS / FAILURE	Variable	Length of JSON Data	Data JSON UI Information

For Example,

For Repeat and Shuffle Commands from Message-Box #40, response is sent with Play JSON. Repeat and Shuffle fields are modified in the response.



Note: You will get Play JSON only, if the current view is play view.

Example of Browse View JSON

```
{ "GetuiView": {
  "browseview": {
    "line0": "MediaServer",
    "line1": "RadioStations",
    "line2": "Airplay",
    "line3": "Multiroom menu"
  }
}}
```

Example PlayView JSON

```
{
  "CMD ID":3,
  "Title":"PlayView",
  "Window CONTENTS":{
    "Album":"Freedom",
    "Artist":"Akon",
    "CoverArtUrl":"coverart.jpg",
    "Current Source":6,
    "Current_time":-1,
    "Genre":"(14)",
    "PlayState":0,
    "Repeat":0,
    "Shuffle":0,
    "TotalTime":240797,
    "TrackName":"Right Now (Na Na Na)"
  }
}
```

6.24.Message-Box 49 (Current Time)

Message-Box 49 is the **Current Time** Message-Box. This message-box is used to send the current playback time from LS to Host-MCU, when the song is played from DMP/ SD-Card/ vTuner / TuneIn, Airplay, C4A, and all other audio sources.

Current Time is updated at intervals of 1 second.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	49	NA	Variable	Length of Data Field	Current Playtime in Seconds

Data Format

Data is the integer value of time in seconds.

Length of the data is 1 byte for playtime 0-9 seconds, 2 bytes for playtime 10-99, and so on.

6.25.Message-Box 50 (Current Source)

Message-Box 50 is the **Current Source** Message-Box. This Message-Box is used by LS to inform the HOST-MCU about the Current Audio Source selected (Command Type: SET).

This happens when the user changes to a new Audio Source in LS and after getting a confirmation from HOST-MCU over Message-Box#10 (Is Playback Allowed), LS updates this Message-Box with the current Audio source selected and sends to HOST-MCU.

This Message-Box can also be used by the HOST-MCU to know the Current Audio Source selected at LS. (Command type: GET in that case)

This Message-Box is only applicable only for a valid UART interface where HOST-MCU communication is possible.

- In case of boot-up and no playback source currently active, LS6 will return success with source name as NO_SOURCE.
- In case there is any source already playing and source switching takes place then LS6 returns success with the current playback source.

Command Type value is always SET when command sent by LS to HOST-MCU.

Command Type value is always GET when command sent by HOST-MCU to LS.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	50	NA	Variable	Length of data field	Current_Source

'Current_Source' values:

Value	Description
1	Airplay
2	DMR
3	DMP
4	Spotify
5	USB
6	SD-Card
7	Melon
8	vTuner
9	TuneIn
10	Miracast
12	DDMS-Slave
14	Line-IN (Aux-In)
15	-
16	Apple Device (iOS Device over USB)
17	Direct URL
18	-
19	Bluetooth
21	Deezer
22	Tidal
23	Favorites
24	GoogleCast

Value	Description
25	External source controlled by HOST-MCU



Note:

When a device is Zone-Station in DDMS and any other source, say Airplay or DMR or SD-Card and so on, tries to hijack, the device source switches and plays the new source audio. Also the device frees itself from the DDMS Mode and enters Station Mode.

6.26.Message-Box 51 (Current Play Status)

Message-Box 51 is the **Current Play State** Message-Box. This Message-Box is used by LS to inform the Remote Controller Interface (HOST-MCU or LUCI APP) about the Current play state of LS (Command Type: SET). Whenever there is a change in in current play state, LS informs the Remote Controller Interface over Message-Box#51.

Command Type value is always **SET** when command sent by LS to HOST-MCU.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	51	NA	Variable	Length of data field	Current_play_State

'Current_play_State' values:

value	0	1	2	3	4	5
Description	Playing	Stopped	Paused	Connecting	Receiving	Buffering



Aux-In

In case Aux-In is controlled by HOST-MCU, and user wants to select Aux-In as a source, then HOST-MCU has to send Stop command (data field set to 1) with Command Type SET over Message-Box#40. LS stops the current Audio playback source if any.

In case, if there was no playback of any other source before switching to AUX source, then there is no necessity to send STOP commands to LS module. The HOST-MCU should then update Message-Box#50 (Current Source) with data field set to 14 (i.e Line-IN (Aux-In/Bluetooth)).

Airplay

There are only two play-states in Airplay that is PLAY and STOP. Airplay does not have a PAUSE state communicated to LS. Pause for 10 seconds is STOP in Airplay.

6.27.Message-Box 52 (Play List Control Commands)

Message-Box 52 is used for **Play List Control Commands**. The Play List control commands will typically be used by the Phone App over Network or HOST-MCU with GUI to send commands to Create, Delete, Clear, Add to, Remove track from Playlist, Get Available Playlists and Get contents of the Playlist.

Command Type value is always **SET** when command sent by the available interface Controller to LS

For Example,

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	52	NA	Variable	Length of data field	PlaylistctrCmd

HOST-MCU or Remote App should send the below commands as a string for the particular selected functionality.

Valid PlaylistctrCmd Values for LibreSync device are as below

Play List Control Command	Description
CREATE:<PlayListName>	User can create custom playlist by sending the above command. Playlist will be generated on USB.
DELETE:<playlist_name>	User can delete the existing playlist by sending the above command.
CLEAR:<playlist_name>	Host can clear the contents existing playlist by sending the above command.
ADDTRACK:<playlist_name>#<url>	<p>User can add contents to the playlist by using the command.</p> <ul style="list-style-type: none"> • <playlist_name> = Name of the playlist already created using create command. • <url > = Playable URL of the content.
RMVTRACK:<playlist_name>#<url>	<p>User can remove contents from the playlist by using the command.</p> <ul style="list-style-type: none"> • <playlist_name> = Name of the playlist already created using create command. • <url > = Playable URL of the content. The Url mentioned will be removed from the playlist.
LISTALL	Lists all the existing playlists.
GETCONTENT:<playlist_name>	This command is used to get contents from the playlist mentioned in <playlist_name>. Contents are sent in the form of JSON.
LOADCURRENT:<playlist_name>	This command is used to load current playlist with contents of playlist mentioned in <playlist_name>.
UPDATECURRENT:<content_jSon>	This command is used to load current playlist with contents of JSONmentioned in <content_jSon>.

Example

```
{
  "PlayIndex": 0
  "UrlList" : [
    {
      "Url" : "/media/sdb/a1.mp3"
    },
    {
      "Url" : "/media/sdb/a2.mp3"
    },
    {
      "Url" : "/media/sdb/a3.mp3"
    }
  ]
}
```

Where playindex is the index from which to start the playback.

Response Command:

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	52	SUCCESS / FAILURE	Variable	0	-

Example,

- CREATE: Libre
- DELETE: Libre
- CLEAR: Libre
- ADDTRACK: Libre#/media/sdb/adc.mp3
- RMVTRACK: Libre#/media/sdb/adc.mp3
- LISTALL
- GETCONTENT: Libre

- CLEAR: Clear Current Playlist
- UPDATECURRENT: "Url":"/media/sdb/ringtone3/song_1.mp3"

The response to LISTALL and GETCONTENT request is sent to host on Message-Box#53 in form of JSON file. Please go through **Message-Box#53** for details.

6.28. Message-Box 53 (Play List view Commands)

Message-Box 53 is used by LS platform to send the JSON info tags to the Network Controller App in response to the **Play List Control Commands (Message-Box#52)** LISTALL and GETCONTENT. Please see the example below:

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	53	SUCCESS / FAILURE	Variable	Length of JSON Data	Data JSON UI Information

This Message-Box is only sent by LS to the Network controller (Phone App).

Example,

<p>LISTALL Response</p> <pre>{ "CMD ID" : 6, "Items" : [{ "PLAYLIST" : "LMRecentlyPlayed" }, { "PLAYLIST" : "LMFavurites" }, { "PLAYLIST" : "Libre" }] }</pre> <p>GETCONTENT Response</p>

```
{
  "CMD ID" : 7,
  "Frame" : 0,
  "Items" : [
    {
      "Album" : "null",
      "Artist" : "null",
      "Genre" : "null",
      "Title" : "null",
      "Url" : "/media/sdb/example1.mp3"
    },
    {
      "Album" : "null",
      "Artist" : "null",
      "Genre" : "null",
      "Title" : "null",
      "Url" : "/media/sdb/example1.mp3"
    }
  ]
}
```

6.29.Message-Box 54 (IS-PLAY STATUS)

Message-Box 54 is used to determine whether the playback is success or failure.

LS sends the Success or Error notification on playback event.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	54	NA	Variable	Length of data field	Success / Error Notification

Data

- success -When playback succeeds.
- error_playfail-When the song has bad header and this song is at the last position in the playlist.
- error_nonextsong – Indicates no next song or end of playlist.
- error_noprevsong – Indicates no previous song.

- error_playfailtrynext -When the song has bad header.
- NO_USB –USB drive not attached.
- NO_SDCARD – SD-Card not attached
- NO_SONGS – USB drive or SD-Card attached but no play list found.

6.30. Message-Box 55 (Reboot from APP)

Message-Box 55 is used to send the reboot command from APP to LS.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	55	SUCCESS	Variable	Length of Data	Reboot

6.31. Message-Box 64 (Volume Control)

Message-Box 64 is the **Volume Control** Message-Box.

LS maintains the volume level from 0 to 100, where 0 is no volume and 100 is max volume.

Volume Change at HOST-MCU side

Whenever there is any volume change done at the HOST-MCU, it is important to inform the Media Controllers about the corresponding volume changes. HOST-MCU has to make sure that the volume level sent to LS is always in the range of 0-100.

Steps involved during the Volume change at HOST-MCU side is as below

- Step 1.** HOST-MCU sends the volume control command with Command Type=SET and value between 0-100 to LS.

Step 2. LS events out the changed volume level to the currently active Media



LS does not send Success / Failure response for the volume change event request from the HOST-MCU side, from release version LS6 v6090 and LS5B5026 and onwards.

Controller.

Step 3. LS sends the volume control command back to HOST-MCU with Command Type= SET and value between 0-100 as a confirmation to HOST-MCU that the changed volume has taken affect and it has been evented out.

Step 4. The HOST-MCU has to change the volume by configuring the Audio Amp.



In case volume level is **zero '0'**, HOST-MCU should inform the volume changes to LS-Module ONLY once.

It is recommended that HOST-MCU should not send Volume Level Zero '0' consecutively multiple times to LS-Module.

Volume Change at Media Controller side

Whenever there is any volume change over any of the Media Controllers such as DMC, Airplay and so on, or LUCI phone based App, it is important to inform the HOST-MCU about the corresponding volume changes.

Any volume control command sent to HOST-MCU is always sent in the range of 0-100. The HOST-MCU has to internally map their volume range to LS (which is always 0-100). In this case only step 5 of the above scenario is applicable.

Command Type value is always SET when volume command sent by HOST-MCU to LS.

Command Type value is always SET when volume command sent by LS to HOST-MCU.

Volume set Command from HOST-MCU to LS

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	64	NA	Variable	Length of data field	Volume_level

'Volume_Level' values: Volume level values are always between 0-100

Volume set Command confirmation from LS to HOST-MCU

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	64	NA	Variable	Length of data field	Volume_level

Figure 6-2 illustrates the Volume Control Message-Box and its message exchanges in more detail.

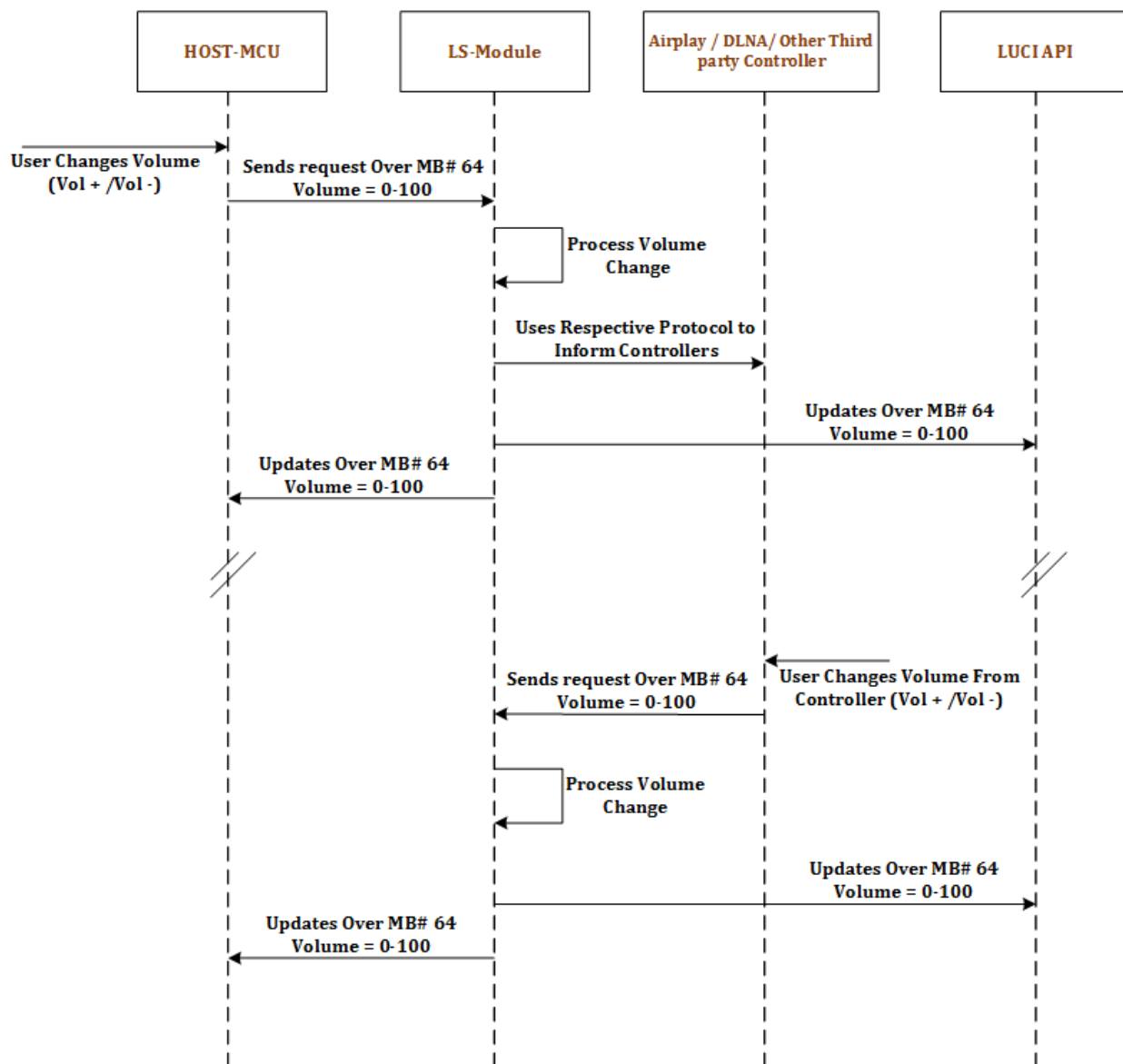


Figure 6-5: Two Way Volume Control Sequence

6.32. Message-Box 65 (Firmware Upgrade Request)

Message-Box 65 is the **Firmware Upgrade Request** Message-Box. HOST-MCU triggers to LS module to initiate firmware upgrade process.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	65	0	Variable	Length of the data field	Data

Data Format

Data is sent in the format **Firmware Upgrade Method, IP, Port, Factory Reset Condition**

See details for each of these parameters below.

Firmware Upgrade Method Format

value	sdcard	usb	Network	Internet
Description	FW upgrade by SD-Card	FW upgrade by USB	FW upgrade by Network	FW upgrade by Internet.

IP

IP address of the URL of *firmware_download.xml* file.

Example: *192.168.1.120*

Port

Port number of the server.

Example: *8080*

Factory Reset Condition format

value	1	0
Description	Reset NV-Item Values	Retain NV-Item Values

Data Format Example,

- Update via SD-Card:
 - Reset NV-Item Values: ***sdcard,0,0,1***
 - Retain NV-Item Values: ***sdcard,0,0,0***
- Update via USB:
 - Reset NV-Item Values: ***usb,0,0,1***
 - Retain NV-Item Values: ***usb,0,0,0***
- Update via Network:
 - Reset NV-Item Values: ***Network,0,0,1***
 - Retain NV-Item Values: ***Network,0,0,0***
- Update via Internet:
 - If IP address to notify progress and Port number is available
 - Reset NV-Item Values: ***Internet,IP,Port,1***
 - Retain NV-Item Values: ***Internet,IP,Port,0***
 - If IP address to notify progress and Port number is not available
 - Reset NV-Item Values: ***Internet,0,0,1***
 - Retain NV-Item Values: ***Internet,0,0,0***

Device boots in BSL and send response to HOST-MCU over Message-Box **0xABCD**.

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	0xABCD	SUCCESS	Variable	Length of the Data	0xAAAAAAAAAA

6.33.Message-Box 66 (Firmware Upgrade Progress)

Message-Box 66 is the **Firmware Upgrade Progress** Message-Box. It indicates the upgrade progress of the LS module **application-image** to HOST-MCU

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	66	SUCCESS	Variable	1	Data

Data Format

value	0 to 100	Complete
Description	Percentage progress varying from 0 to 100. For example, 0%...2%...4%...6%...8%...10%...12%.....100%	LS sends the data as Complete on successful upgrade.

6.34. Message-Box 68 (Host-Image-Present)

Message-Box 68 is the **Host-Image-present** Message-Box. This Message-Box is used by LS to inform the HOST-MCU the availability of Host-Image for Update.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	68	SUCCESS / FAILURE	Variable	10 bytes	HostImageSize

Data

HostImageSize: Indicates the size of the Host-Image available for update.

6.35. Message-Box 69 (Request Firmware Upgrade)

Message-Box 69 is used to Request Firmware Upgrade. On Firmware Upgrade trigger LS sends firmware upgrade request to HOST-MCU and waits for a response from Message-Box 211.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	69	NA	Variable	25 bytes	Data

Data Format

- SD-Card, USB, Network Method of Firmware Update

When the firmware upgrade method is SD-Card, USB or Network, LS sends a text string "**RequestForFirmwareUpgrade**" to HOST-MCU.

- Internet Method of Firmware Update

When the firmware upgrade method is internet, LS follows the steps as listed below and sends the appropriate notifications to HOST-MCU.

Step 1. Download XML file: LS downloads the *fwdownload.xml* from the URL provided.

- If the XML download fails, LS sends error message as "2" to HOST-MCU.

Step 2. Parse XML file: On successful download LS parses the *fwdownload.xml* file.

- If the XML parsing fails, LS sends error message as "1" to HOST-MCU.

Step 3. Send Firmware Version: After successful parsing, LS sends message as "3,<NEW/OLD/SAME>,<New FW version>, <New Host version>".

Where,

- NEW - ***fwdownload.xml*** file has a NEW firmware, than the one present in the module.
- OLD - ***fwdownload.xml*** file has an OLD firmware, than the one present in the module.
- SAME - ***fwdownload.xml*** file has the same firmware, that is present in the module.



LS sends the notifications to HOST-MCU listed in the above steps, only when the **fwinternet_host** NV-Item is set as **1** in the FENV.xml file.

End --

6.36.Message-Box 70 (APP CONTROL)

Message-Box 70 is used for **APP CONTROL**. This command is used to trigger data set as follows:

- **PLAYSD:** When LS receives Message-Box#70 with cmd_type =2 and data set to PLAYSD, LS module starts the SD-Card playback from the first file of the first folder of the SD-Card.
- **PLAYUSB:** When LS receives Message-Box#70 with cmd_type =2 and data set to PLAYUSB, LS module starts the USB playback from the first file of the first folder of the USB.
- **STATUSSD:** When LS receives Message-Box#70 with cmd_type =2 and data set to STATUSSD, LS module writes the status of SD-Card , for example, total size of SD-Card, free space available etc. , in Message-Box #71 (SD-Card status).
- **SCANS:** This option is used typically by the Phone App to help browsing the contents of the SD-Card based in ID3 Tags. When LS receives Message-Box#70 with cmd_type =2 and data set to SCANS, LS responds back on Message-Box#42 with the JSON file containing the ID3 Tags(Album, Artist, Genre and Title) along with the url of all audio files present on SD-Card in the following format:

```
{
  "CMD ID":5,
  "Frame":1,
  "Items":[
    {
      "Album":"3",
      "Artist":"Mandolin Seenu",
      "Genre":"(24)",
      "Title":"Theme of 3",
      "Url":"/media/sdb/songs/3/TamilBeat.Com - 3 Movie Theme.mp3"
    },
    {
      "Album":"3",
      "Artist":"Theme Music",
      "Genre":"(24)",
      "Title":"A life Full of love",
      "Url":"/media/sdb/songs/3/TamilBeat.Com - A Life Full of
Love.mp3"
    },
    {
      "Album":"3",
      "Artist":"Aniruth, Nadisha Thomas, Malavika Manoj",
      "Genre":"(24)",
      "Title":"Come on Girls",
      "Url":"/media/sdb/songs/3/TamilBeat.Com - The Celebration of
Love.mp3"
    }
  ]
}
```



Typically there can be thousands of audio files present on SD-Card so the above information based on ID3 tags is sent per frame bases where every frame contains information of 10 Audio Files. So if SD-Card has 100 audio files stored then this information is sent over 10 frames.

Also this option is used by the phone Application only when the phone App intends to display the SD-Card contents on basis of ID3 tags.

The SD-Card control command **PLAYSD** can be used by any of the available interface controller (Example HOST-MCU over UART or Phone App over Network). **SCANS** commands is typically used by the Phone App or any HOST-MCU with GUI.

- **FAV_SAVE:** HOST-MCU sends command **FAV_SAVE** to save the current playing song
- **FAV_PLAY:** HOST-MCU sends command **FAV_PLAY** to play the latest favourite song saved.
- **FAV_SAVE: <<Index>>:** HOST-MCU send command **FAV_SAVE: <<Index>>**, to save the current playing song in the specified location. Index range is from 0-9.

For Example, To save the current playing song in location 5 of the Favourite list HOST-MCU sends the command as **FAV_SAVE:5**.

- LS sends the following response to HOST-MCU on receipt of command **FAV_SAVE**.
 - **GENERIC_FAV_SAVE_SUCCESS:** LS sends command **GENERIC_FAV_SAVE_SUCCESS**, on successful save of the favorite list.
 - **GENERIC_FAV_SAVE_FAIL:** LS sends command **GENERIC_FAV_SAVE_FAIL**, when the save action is not successful.
 - **GENERIC_FAV_EXISTS:** LS sends command **GENERIC_FAV_EXISTS**, if the list is already save as favorite.
- **FAV_PLAY: <<Index>>:** HOST_MCU sends command **FAV_PLAY: << Index>>**, to play the favourite song saved in the specified location.

For Example, to play the song in location 6 of the Favourite list HOST-MCU sends the command as **FAV_SAVE:6**.

- LS sends the following response to HOST-MCU on receipt of command **FAV_SAVE**.

- **GENERIC_FAV_PLAY_SUCCESS**: LS sends command **GENERIC_FAV_PLAY_SUCCESS**, on successful playback of the favorite list.
- **GENERIC_FAV_PLAY_FAIL**: LS sends command **GENERIC_FAV_PLAY_FAIL**, when the playback is not successful.
- **GENERIC_FAV_PLAY_NOURL**: LS sends command **GENERIC_FAV_PLAY_NOURL**, when the URL for the playback list is not found.

Command Type value is always **SET** when command sent by the available interface Controller to LS

For Example

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	70	NA	Variable	Length of data field	data

HOST-MCU or Remote App should send the below commands as a string for the particular selected functionality.

Valid data field Values for LibreSync device are as below

- PLAYS
- PLAYUSB
- STATUS
- SCANS
- FAV_SAVE
- FAV_PLAY
- FAV_DELETE

- FAV_SAVE:<<Index>>
- FAV_PLAY:<<Index>>

Response Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	70	SUCCESS / FAILURE	Variable	Length of data	Data

LS sends the below commands as a string to HOST-MCU or Remote APP to report the status of the Favorite actions.

Valid data field Values for LibreSync device are as below

- FAV_SAVE_SUCCESS / FAV_SAVE_FAILURE / FAV_EXIST
- FAV_DELETE_SUCCESS / FAV_DELETE_FAILURE

6.37.Message-Box 71 (SD-Card Status)

Message-Box 71 is **SD-Card Status** Message-Box. This Message-Box is used by Libre APP or the Host-MCU controller to retrieve the status of SD-Card such as, Total Size of SD-Card, Free space available, and so on.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	71	SUCCESS / FAILURE	Variable	Length of Data	xxx:yyy:zzz:aaa

Data Format

- xxx: Total Size of SD-Card in bytes
- yyy: Used Space of SD-Card in bytes
- zzz: Free Space of SD-Card in bytes
- aaa: Total number playable file on SD-Card

6.38.Message-Box 75 (Spotify Preset Actions)

Message-Box 75 is **Spotify Preset Actions** Message-Box. This Message-Box is used by HOST-MCU to command LS-Module to Save, Play, or Reset the presets in Spotify.

Command from HOST-MCU to LS

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	75	SUCCESS/FAILURE	Variable	Variable	Preset Actions

Data

- PLAY:(1,2,3)
- SAVE:(1,2,3)
- DELETE:(1,2,3)
- RESET

At present there are 3 slots for Spotify Preset and the preset actions can be used to Save, Play, Reset or Delete the preset slots in Spotify.

For all the above issued actions HOST-MCU will receive SUCCESS / FAILURE response from the APP as per LUCI protocol.

For Example,

- To save the preset in slot 1, HOST-MCU issues the command **SAVE:1**

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	75	SUCCESS/FAILURE	Variable	Length of data	SAVE:1

- To Play the preset in slot 1, HOST-MCU issues the command **PLAY:1**

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	75	SUCCESS/FAILURE	Variable	Length of data	PLAY:1

- To Delete the preset in slot 1, HOST-MCU issues the command **DELETE:1**

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	75	SUCCESS/FAILURE	Variable	Length of dat	DELETE:1

- To Reset all the 3 slots at once, HOST-MCU issues the command **RESET**

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	75	SUCCESS/FAILURE	Variable	Length of Data	RESET

Response from LS to HOST-MCU

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	75	SUCCESS/FAILURE	Variable	0	-

6.39. Message-Box 80 (Play Audio Index)

When HOST-MCU sends a command (play 1/play 2/play 3/play 4/play 5) on MB#80 the songs (song1.mp3/song2.mp3/song3.mp3/song4.mp3/song5.mp3) present in root will be played and LS will respond with status "SUCCESS" on the same Message-Box and if HOST-MCU sends some other command other than those which are mentioned LS will responds back with status "NI"(No Index) on the same Message-Box indicating failure.

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	80	NA	variable	length of data field	data

Data Format:

- play 1
- play 2
- play 3

- play 4
- play 5

Response

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	80	success/ failure	variable	length of data field	SUCCESS/NI

Response in data field will be "SUCCESS" if a valid data as mentioned in data format is sent by the HOST-MCU else "NI" (No Index) will be sent back to host indicating failure.

6.40.Message-Box 81 (I2C Client Access)

Message-Box 81, is I2C Client Access Message-Box. This Message-Box provides the flexibility to configure the I2C Clients. Currently this functionality is limited to configure CODEC (only write mode) and will be expanded later.

To configure, the register should be written with the string in the below format.

I2CclientType:RegAddr:nRegToWrite:RegVal1,RegVal2....

Where,

- I2CclientType: The identifier that provides info on the I2C client to write. Should be terminated by a delimiter ":" (colon). Supported types are, CODEC: To configure I2C Client, codec. (WM8904)
- RegAddr: The register address of the I2C Client to be written. Should be terminated by a delimiter ":" (colon).
- nRegToWrite: The number of sequential registers to be written starting from "RegAddr". This mode is useful when we need to configure a series of registers(say from 0x8C to 0x9D) in a single command rather than configuring the registers

individually. Make sure that all "nRegToWrite" is provided with a proper "RegValX". Should be terminated by a delimiter ":" (colon).

- RegValX: The register value to be written.. The value should be in Hexadecimal. For codec it can be up to 2 bytes long. The values should be separated by delimiter "," (comma).

Example: "CODEC:0x86:0x4:0x1,0x2,0x3,0x4"

This will write the codec register starting with 0x86 to 0x89(4 registers) with values 0x1, 0x2, 0x3 and 0x4 resp. Again for example register#87 have a range of [0:4] bits and hence the value in the above string can be provided as 0x0002, or 0x02 or 0x2. We recommend to provide only valid bits 0x2 to reduce the number of bytes to be transferred.

For example to configure two registers starting from 0x8B,(range-> 0x8B[0:4], 0x8c[0:15]), provide a string as

"CODEC:0x8B:0x2:0xc,0xABCD"

- To-Do: The response of this register is not implemented with proper error values, will be implemented later. Improper parameters (number and value) are not handled.

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	81	NA	Variable	Length of Data	I2CclientType:RegAddr:nRegToWrite:RegVal1,RegVal2....

6.41.Message-Box 90 (Device Name)

Message-Box 90 is **Device Name** Message-Box. This Message-Box can be used by LUCI App or MCU controller to read the Device Name and also change the device name.

To read Device name from MCU or LUCI App:

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	90	NA	Variable	0	-

Response form LS module

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	90	SUCCESS/FAILURE	Variable	Length of Data	Device Name

To set the Device name from HOST-MCU or LUCI App

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	90	NA	Variable	Length of Data	New Device name

Response from LS module

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	90	SUCCESS/FAILURE	Variable	0	-

This Message-Box will also be asynchronously sent to all controller whenever Device name gets changed by other Applications (Like Webserver or WAC).

6.42.Message-Box 91 (Device MACID)

Message-Box 91 is Device MACID message-box. : This message box will send LS MAC ID to Host-MCU over LUCI.

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	91	NA	NA	Length of MAC id (12 bytes)	MAC id

6.43.Message-Box 95 (Aux-Input Start)

Message-Box 95 is the **Aux-Input Start** Message-Box. This Message-Box is used by the HOST-MCU to switch to audio source to Aux-Input.

Steps involved in Aux-Input Start is as below.

- Step 1.** HOST-MCU may send a command to LS (Command Type: SET) over Message-Box#95 without data
- Step 2.** LS sends the new source to the HOST-MCU over Message-Box# 10 as "15"
- Step 3.** LS sends the success/Failure status to the HOST-MCU over Message-Box# 95
- Step 4.** HOST-MCU may send a command to LS (Command Type: GET) over Message-Box#50 (Current Source) to get to know the current playback source, if any
- Step 5.** HOST-MCU takes a decision whether it has to allow/disallow the new Audio Source.
- Step 6.** Host-MCU sends confirmation on Message-Box#11 with data field set to 1/0 for Allowed/Not-allowed respectively.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	95	NA	Variable	0	None

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	95	SUCCESS / FAILURE	Variable	0	None

6.44.Message-Box 96 (Aux-Input Stop)

Message-Box 96 is the **Aux-Input Stop** Message-Box. This Message-Box is used by the HOST-MCU to stop the audio source from Aux-Input.

Steps involved in Aux-Input Stop is as below.

Step 1. HOST-MCU may send a command to LS (Command Type: SET) over Message-Box#96 without data

Step 2. LS sends the success/Failure status to the HOST-MCU over Message-Box# 96

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	96	NA	Variable	0	None

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	96	SUCCESS / FAILURE	Variable	0	None

6.45.Message-Box 97 (External PlayBack)

Message-Box 97 is External Playback message-box. This message-box is used to enable or disable the audio playback from external source.

- HOST-MCU sends **START** to enable playback from external source.
- HOST-MCU sends **STOP** to stop playback from external source.

External source is the source that is not supported in LS-Platform. LS-Platform supports the following source.

- Airplay
- DMR
- DMP
- Spotify
- USB
- SD-Card
- Melon
- vTuner
- TuneIn
- Miracast
- DDMS-Slave
- DDMS Line-IN (Aux-In)
- Line-IN (Aux-In)
- Apple Device
- Direct URL
- Bluetooth
- Deezer
- Tidal
- Favorites
- Google Cast For Audio
- External Source

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	97	NA	Variable	Length of Data	START / STOP

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	97	SUCCESS / FAILURE	Variable	Length of Data	START / STOP

6.46.Message-Box 100 (DDMS Trigger)

This Message-Box is used by LS-Module to receive trigger for DDMS.

This Message-Box is applicable for DDMS triggers from HOST-MCU and also from the LUCI Smartphone App.

The Response with a command Status is sent back immediately to the Host and status of the command execution can be seen by querying using a GET command to the DDMS status Message-Box

Following are the Data field, which can be sent by HOST-MCU or by LUCI Smartphone App in this MID.

- **SETMASTER:** Changes the mode of the speaker to DDMS Audio-Master mode. The device becomes the prime node to receive original content from the Content Source.
- **SETSLAVE:** Changes the mode of the speaker to DDMS Audio-Client. The speaker now can receive DDMS streams from DDMS Audio-Master and playback synchronously.
- **SETFREE:** Frees a speaker from DDMS Group and acts like a standalone speaker

- **JOINALL:** When this command is triggered on the existing Audio-Master, all speakers becomes Audio-Client to this Audio-Master.
- **JOINNEXT:** When this command is triggered on Audio-Client, this speaker-device joins to the next Audio-Master in the network.
- **DROPALL:** Frees all the speakers in all DDMS Groups and speakers will act like a standalone speaker.
- **DROPME:** Frees a speaker from DDMS Group and acts like a standalone speaker

For Example,

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	100	NA	Variable	Length of Data	DataField (as explained above)

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	100	SUCCESS / FAILURE	Variable	0	-

6.47.Message-Box 101 (Stand Alone Mode)

This Message-Box can be set to make LS-Module go into SA (Stand Alone) Mode.

In this mode LS-Device disconnects from the Home-AP and forms a P2PGo (Network Master).

A Smart phone can be used to connect to the Speaker directly for DDMS configuration or DirectPlayback.

On receiving the command over this message-box, LS-Enabled device enters into SA-Mode and becomes Audio-Master.

For Example

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	101	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	101	SUCCESS / FAILURE	Variable	0	-

LS-Device reboots, every time when it is switched from HN-Mode to SA-Mode and vice-versa. The sub-scenario's involved in this mode switching and functional behavior is described below:

Current State of the product and the User action	Functional Behavior
HN (already configured to AP) to SA	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
HN (WAC Mode) to SA	No reboot

Current State of the product and the User action	Functional Behavior
HN (Not in WAC mode & factory reset mode) to SA	No reboot
SA (Network Master) to HN (already configured to AP)	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
SA (Network Master) to HN (Not in WAC mode & Factory reset mode)	No reboot
SA (Network client) to HN (already configured to AP)	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
SA (Network client) to HN (Not in WAC mode & Factory reset mode)	No reboot. It should enter into WAC mode

6.48.Message-Box 102 (Search LS-Module)

This Message-Box can be used to make LS-Module to search for another LS-module in DirectMode and connect to it.

Once connected it will be also discoverable to the Smartphone App connected to LS-Module in Direct mode.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	102	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	102	SUCCESS / FAILURE	Variable	0	-

LS-Device reboots, every time when it is switched from HN-Mode to SA-Mode and vice-versa. The sub-scenario's involved in this mode switching and functional behavior is described below:

Current State of the product and the User action	Functional Behavior
HN (already configured to AP) to SA	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
HN (WAC Mode) to SA	No reboot
HN (Not in WAC mode & factory reset mode) to SA	No reboot
SA (Network Master) to HN (already configured to AP)	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
SA (Network Master) to HN (Not in WAC mode & Factory reset mode)	No reboot
SA (Network client) to HN (already configured to AP)	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.

Current State of the product and the User action	Functional Behavior
SA (Network client) to HN (Not in WAC mode & Factory reset mode)	No reboot. It should enter into WAC mode

6.49.Message-Box 103 (Query DDMS)

This Message-Box can be used to query DDMS status of LS-Module.

This Message-Box will also be evented out to both UART and TCP interface as soon as the status changes.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	103	NA	Variable	Length of data	Length of Data

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	103	SUCCESS / FAILURE	Variable	Length of data	Length of Data

Data Field contains the status of DDMS in the below format

<DeviceType>, <SpeakerType> <ID>

Where,

DeviceType can be

- MASTER
- SLAVE

- FREE

SpeakerType can be

- STEREO
- LEFT
- RIGHT

ID can be

- In HN-Mode: Concurrent SSID
- In SA-Mode: Zone ID

6.50. Message-Box 104 (Set Zone-ID)

Message-Box 104 is the **Set Zone-ID** Message-Box. This Message-Box can be used to set the zone-ID of the DDMS Groups before setting up Station-Master or Station-Client.

Zone-ID would be a combination of unique multicast address and a unique port number.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	104	NA	Variable	Length of Data	Zone-ID

Data Field contains a unique multicast address and a unique port number as mentioned below.

Example: Zone id: 259.255.255.251:3000

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	104	SUCCESS / FAILURE	Variable	Length of Data	Zone-ID

6.51.Message-Box 105 (DDMS SSID)

Message-Box 105 is DDMS SSID message-box. It is used for creating Multi-zone in Home network DDMS mode, with each zone having a unique SSID (name).Members in a zone will fall under the same SSID.

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET/GET	105	NA	Variable	Length of Data	DDMS concurrent zone SSID

Data Format: Data Field contains a unique DDMS concurrent zone SSID as mentioned below.

Example: DDMS concurrent zone SSID - DIRECT-LBMyTestZoneF3635E

Response

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET/GET	105	SUCCESS / FAILURE	Variable	Length of Data	DDMS concurrent zone SSID

6.52. Message-Box 106 (Speaker Type)

This Message-Box is used by LS-Module to define the speaker type.

This Message-Box is applicable for triggers from HOST-MCU and also from the LUCI Smartphone App.

Following are the Data field, which can be sent by HOST-MCU or by LUCI Smartphone App in this MID.

- **SETSTEREO:** Set the Speaker to Stereo mode
- **SETLEFT:** Sets the speaker into Left Channel only mode. The speaker will only play the left channel of 2 channel audio
- **SETRIGHT:** Sets the speaker to Right Channel only mode. Speaker will only play the right channel of 2 channel audio

For Example,

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	106	NA	Variable	Length of Data	DataField (as explained above)

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	106	SUCCESS / FAILURE	Variable	0	-

6.53. Message-Box 107 (Scene Name)

Message-Box 107 is Scene Name message-box. This message-box is used to edit the name of the scene. The maximum length of the data to be used is 128 characters.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	107	NA	Variable	Length of Data	Scene_Name

6.54. Message-Box 108 (Setup Stereo Pair) – LS9 Module Only

Message-Box 108 is Setup Stereo Pair message-box. This message-box is used to set-up a stereo pair using two speaker devices.

Command (HOST-MCU to LS)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	108	NA	Variable	Length of Data	Data

Data

HOST-MCU should send the below listed commands to LS-Module, to setup the speaker devices as stereo pair.

- **MASTERLEFT** – To set-up the speaker devices as a **Master** with **Left** channel audio-output.
- **MASTERRIGHT** - To set-up the speaker devices as a **Master** with **Right** channel audio-output.

- **SLAVELEFT** - To set-up the speaker devices as a **Slave** with **Left** channel audio-output.
- **SLAVERIGHT** – To set-up the speaker devices as a **Slave** with **Right** channel audio-output.

Example,

To setup a Stereo-Pair of Master (Speaker A) with Right Channel audio-output and Slave (Speaker B) with Left Channel audio-output, do as below.

Step 1. Set-up Speaker A as MASTER with Right Channel

- 1 HOST-MCU should send MASTERRIGHT command over MB #108 to Speaker A
- 2 On Successful set-up of the speaker LS-Sends 1 to HOST-MCU over MB #108

Step 2. Set-up Speaker B as Slave with Left Channel

- 1 HOST-MCU should send SLAVELEFT command over MB #108 to Speaker B
- 2 On Successful set-up of the speaker LS-Sends 1 to HOST-MCU over MB #108

Speaker A and Speaker B are now set as a Stereo pair in DDMS network.

Stereo Pair Time Out

- Once the speaker is set as Master either with Right or Left channel audio-output, speaker waits for 120 seconds for the appropriate slave speaker to join. If the Slave speaker does not join, then the speaker will become free and will be in stereo mode.
- Once the speaker is set as Slave either with Right or Left channel audio-output, speaker searches for the appropriate Master speaker for 120 seconds. If the Master speaker is not found, then the speaker will become free and will be in stereo mode.

Response (LS to HOST-MCU)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	108	NA	Variable	Length of Data	Data

Data

LS-Module sends the below response to HOST-MCU.

- **1** – Successful set-up of the speaker device according to the command.
- **0** – failure to set-up the speaker device according to the command.

6.55.Message-Box 111 (TCP/IP Tunneling Start)

TCP/IP Tunneling is a unique feature supported by LS modules.

Tunneling enables a HOST-MCU (connected over UART) to communicate with other network devices.

The MCU can build their own proprietary protocol using LUCI tunneling.

Below is a simple scenario description

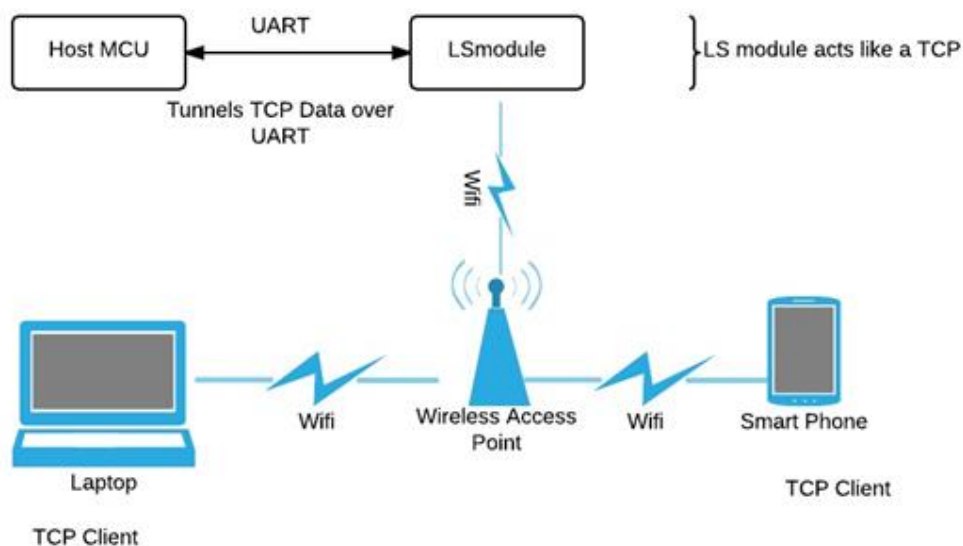


Figure 6-6: LUCI TCP Tunneling Scenario

To start Tunneling HOST-MCU has to specify the Port number using this Message-Box.

Once triggered LS module open a TCP listening socket and is ready to accept connection from network clients,

Following is the sequence.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	111	0	Variable	Length of Data	<portnumber>

Example of port number field can be like 4444.

All fields in data field is in ASCII representation.

Response

LS will respond to the command immediately as SUCCESS/FAILURE

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	111	SUCCESS / FAILURE	Variable	0	-

6.56. Message-Box 112 (Tunnel Data)

Once a Tunnel is created using the Message-Box 111 .LS opens a TCP listening socket and accepts connection from Remote Clients.

LS can accept more than one TCP connections with a MAX limit set to 10.

All data received from TCP socket from any of the connections will be passed to the MCU over this Message-Box.

MCU can also send Data to connected client using this Message-Box.

Below Sequence diagram shows details of steps involved for using TCP tunneling.

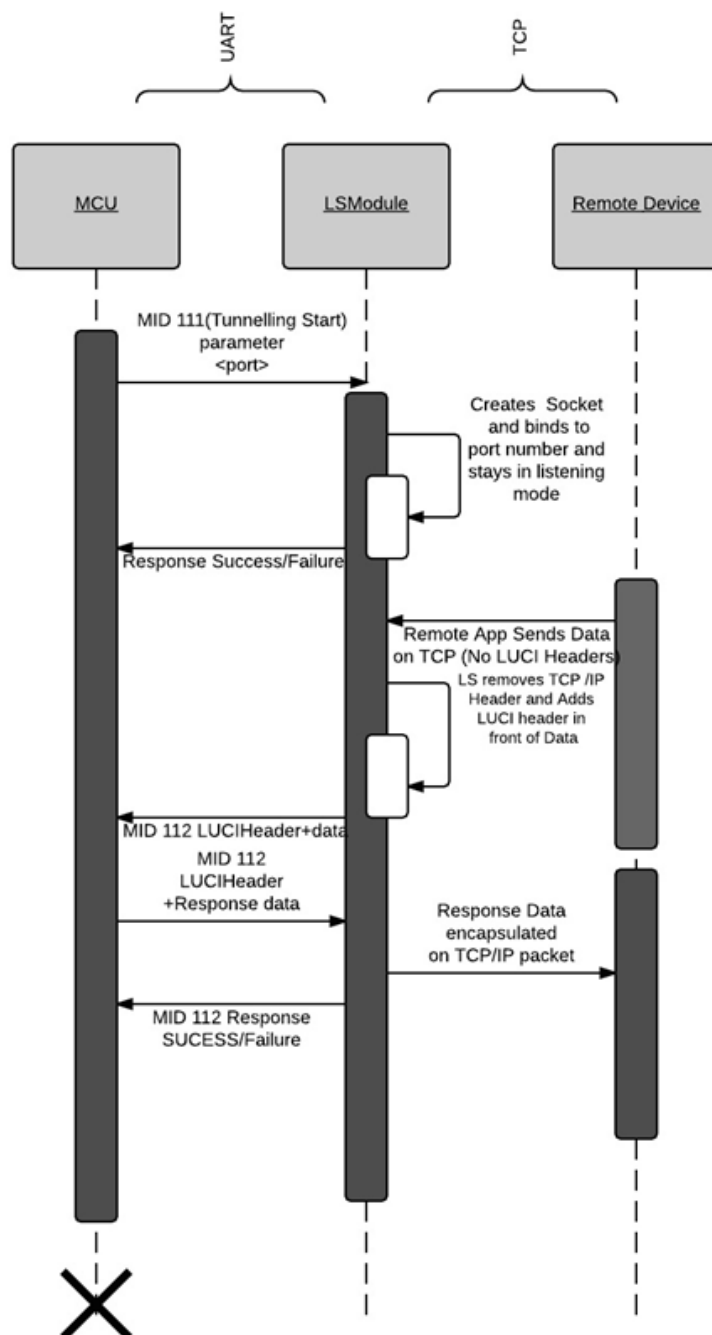


Figure 6-7: TCP / IP Tunneling Sequence

Both MCU and LS module can use this Message-Box to send TCP Data to each other in the below format

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	112	0	Variable	Length of Data	TCP Raw Payload

If MCU sets this Message-Box, then LS will respond back with Status.

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	112	SUCCESS / FAILURE	Variable	0	-

6.57.Message-Box 113 (Miracast Control)

Message-Box 113 is the **Miracast Control** Message-Box. This Message-Box is used by the HOST-MCU/Phone App to start/stop Miracast on LS platform.

After having received this command with cmd_type = 2 and data set to 1, LS should receive value 1(New Audio Source Allowed) from Message-Box # 11. Then LS triggers Miracast session and is discoverable on any of the source device which supports Standard Miracast feature and the source device can connect to LS module. After successful connection the source device can start the audio playback.

For ending the Miracast session, HOST-MCU/Phone App has to send cmd_type = 2 and data set to 0. This disconnects the Miracast session.



Note:

Miracast support is not available for LS-5. Also source device has to select audio playback files only for Miracast streaming.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	113	NA	Variable	Length of Data Field	data

Values of data field:

value	0	1	2
Description	Stop Miracast	Start Miracast HN-Mode	Start Miracast in SA Mode

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	113	SUCCESS / FAILURE	Variable	0	-

6.58.Message-Box 114 (Re-boot Request)

Message-Box 114 is the request for re-boot. This Message-Box will be sent from LS to Host-MCU requesting the reboot command.

LS-Device reboots, every time when it is switched from HN-Mode to SA-Mode and vice-versa. The sub-scenario's involved in this mode switching and functional behavior is described below:

Current State of the product and the User action	Functional Behavior
--	---------------------

HN (already configured to AP) to SA	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
HN (WAC Mode) to SA	No reboot
factory reset mode to SA	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
SA (Network Master) to HN (already configured to AP)	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
Factory Reset Mode → SA (Network Master) to HN	No reboot
SA (Network client) to HN (already configured to AP)	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
Factory Reset Mode → SA (Network client) to HN	No reboot.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	114	SUCCESS	Variable	Length of Data	reboot request

6.59. Message-Box 115 (Reboot Command)

Message-Box 115 is the **Reboot Command** Message-Box. Host-MCU sends the reboot command to LS module.

After reboot Host-MCU waits for Message-Box # 0xAAAA.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	115	SUCCESS	Variable	0	-

6.60.Message-Box 124 (Network Connection Status)

Message-Box 124 is the **Connection Status** Message-Box. LS informs the HOST-MCU about the current network status over various interfaces like WLAN, ETH, P2P, and various configuration modes such as WAC / SAC / LS-Connect.

When there is a change in Network Type or Network Status this message is sent to the HOST-MCU with Command Type: SET.

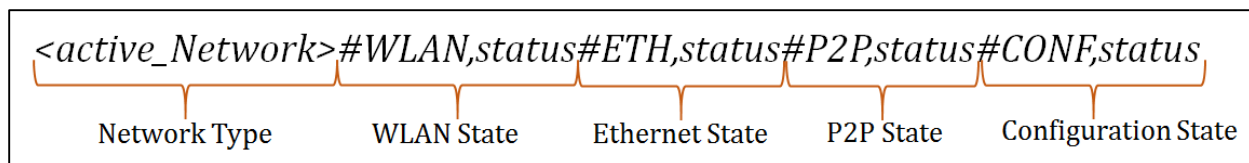
For Example,

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	124	0	Variable	Length of Data	Network_Status

The Data length field is always 17 Bytes. Data has the following format:

<active_Network>#WLAN,status#ETH,status#P2P,status#CONF,status



Where,

- **active_Network** is the current active network interface, values for the Network Type / Network Interface is as below.
 - 1 = WLAN
 - 2 = ETH
 - 3 = P2P
 - 4 = WAC/SAC/LS-Connect
- **WLAN,Status** is as below
 - 0 = WLAN is not active
 - 1 = WLAN is active
- **ETH, Status** is as below
 - 0 = Ethernet is not active
 - 1 = Ethernet is active
- **P2P, Status** is as below
 - 0 = P2P is not active
 - 1 = P2P is active
- **CONF, Status** is as below
 - 0 = WAC / SAC / LS-Connect configuration mode is not active.
 - 1 = WAC / SAC / LS-Connect configuration mode is active.

For Example,

If the current active network type / network interface is WLAN, and LS is connected to the Network over WLAN then, LS sets the data field of Message-Box#124 to:

- In Home-Network Mode or Infrastructure Mode, values will be
1#1,1#2,0#3,0#4,0
- In DDMS HN mode or Miracast, values will be
3#1,1#2,0#3,1#4,0
- In DDMS SA mode, values will be
3#1,0#2,0#3,1#4,0
- In Configuration mode (WAC/SAC/LS-Connect) values will be
4#1,0#2,0#3,0#4,1



Note: The data field is String type so the HOST-MCU has to interpret it
The HOST-MCU can always send a GET to the same Message-Box to know about the current status of connectivity

6.61.Message-Box 125 (Configure Network)

This Message-Box can be used by HOST-MCU or Libre App controller to make LS module connect to specific Access points in Station mode.

Both App developers and HOST-MCU can make use of this message-box to receive the credentials of Home AP from the user and send the same to LS module.

LS module will reconnect to this AP once it receives the credential.

HOST/APP->LS

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	125	0	Variable	Length of Data	DATA

DATA format: <SSID>,<Passphrase>

Example, If SSID of the home AP is JunkYardLab and Passphrase is hello123

Then DATA=JunkYardLab, hello123

Intermediate Response from LS module

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	125	Success/failure	Variable	0	-

Final Status of connection can be known by listening to Message-Box 124 or by discovering the device using LSSDP.

6.62. Message-Box 126 (i-Devices Wi-Fi Settings Sharing)

Message-Box 126 is **i-Devices Wi-Fi Settings Sharing** Message-Box. This Message-Box is used by Host-MCU to request LS to share the Wi-Fi settings information.

Command (Host-MCU to LS)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	126	NA	Variable	0	-

Response (LS to Host-MCU)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	126	NA	Variable	Length of Data	Sharing Status

Data

Data	Explanation
0	Waiting For connection
1	User Declined
2	Network Information Unavailable
4	Configuration Success
5	Configuration Failed

When the Host sets the MB#126 requesting for Wi-Fi Information Sharing, iOS device displays the message to the user asking to 'ALLOW' or 'Ignore' the sharing of Wi-Fi Information.



- When user selects Ignore , Accessory sends '1' to Host-MCU
- When user selects Allow, Accessory first sends '0' to Host specifying waiting for connection. Based on the status of connection Accessory then sends :
 - 4, if the Configuration Successful
 - 5, if the Configuration Fails

6.63. Message-Box 134 (LINKSTATUS) – LS9 Module Only

Message-Box 134 is LINKSTATUS message-box. Tis message-box is used to get Ethernet Link up and Link down status from HOST-MCU to LSx.

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	134	0	Variable	0	Data

Data

- LINKUP
- LINKDOWN

Response,

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	134	SUCCESS / FAILURE	Variable	NA	NA

6.64.Message-Box 140 (WPS Configuration status)

Message-Box 140 is the **WPS Configuration status** Message-Box. This Message-Box is used by LS to update the WPS configuration status to the HOST-MCU.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	0140	NA	Variable	Length of data field	wps_config_status

wps_config_status values:

Status	Value
WPS Failed	0
WPS Timeout	1
WPS Success	2

6.65.Message-Box 141 (WPS Configuration)

Message-Box 141 is the **WPS Configuration** Message-Box. This Message-Box is used by the HOST-MCU to trigger WPS Mode.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	0141	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	0141	SUCCESS / FAILURE	Variable	0	-

6.66.Message-Box 142 (Network Configuration)

Message-Box 142 is the **Network Configuration** Message-Box. This Message-Box is used by the HOST-MCU to trigger WAC Mode or to come out of DDMS-Mode and enter into STA-Mode.

Once WAC is initiated, either iOS device or device webpage can be used, to configure LS-Device to Home AP.

The status of Network Configuration is sent by LS-Module over Message-Box#143 to Host-MCU.

LS-Device reboots, every time when it is switched from HN-Mode to SA-Mode and vice-versa. The sub-scenario's involved in this mode switching and functional behavior is described below:

Current State of the product and the User action	Functional Behavior
HN (already configured to AP) to SA	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
HN (WAC Mode) to SA	No reboot
HN (Not in WAC mode & factory reset mode) to SA	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
SA (Network Master) to HN (already configured to AP)	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
SA (Network Master) to HN (Not in WAC mode & Factory reset mode)	No reboot
SA (Network client) to HN (already configured to AP)	LS sends the reboot-request over LUCI MB#114 to Host-MCU (only if hostpresent 1). Based on the reply by Host-MCU over MB#115, LS does reboot. Then does the actual switch to required mode.
SA (Network client) to HN (Not in WAC mode & Factory reset mode)	No reboot. It should enter into WAC mode



Note:

For Modules that support GCAST, when Switching from HN mode to SA mode, NV-Item "GoogelCast" should be set as "false"
For details on NV-Item usage refer to "***LibreWirelessTechNote -Non-Volatile_Items_in_LibreSync***" available in Libre Portal.

Scenarios:

Scenario 1

Precondition	LS-Enabled speaker is not connected to AP .
Action	WAC / SAC is triggered using the Setup Button (Press and Hold for three seconds) in LS EVK or through Message-Box 142.
Result	LS-Enabled Speaker enters WAC / SAC mode for Network Configuration.

Scenario 2

Precondition	LS-Enabled speaker is connected to AP .
Action	WAC / SAC is triggered using the Setup Button (Press and Hold for three seconds) in LS EVK or through Message-Box 142.
Result	LS-Enabled Speaker enters WAC / SAC mode for Network Configuration.

Scenario 3

Precondition	LS-Enabled speaker is not connected to AP .
Action	<ol style="list-style-type: none"> 1. Reboot the Speaker. 2. Configure the speakers to DDMS SA-Mode (refer to DDMS usability guide). 3. Trigger WAC / SAC using the Setup Button (Press and Hold for three seconds) in LS EVK or through Message-Box 142.
Result	LS-Enabled Speaker switches from DDMS Mode to WAC / SAC mode for Network Configuration.

Scenario 4

Precondition	LS-Enabled speaker is connected to AP .
---------------------	--

Action	<ol style="list-style-type: none"> 1. Reboot the Speaker. 2. Configure the speakers to DDMS SA-Mode (refer to DDMS usability guide). 3. Trigger WAC / SAC using the Setup Button (Press and Hold for three seconds) in LS EVK or through Message-Box 142.
Result	LS-Enabled Speaker switches from DDMS Mode and connects to the AP configured earlier.

Scenario 5

Precondition	LS-Enabled speaker is in DDMS SA-Mode and was not connected to AP earlier.
Action	<ol style="list-style-type: none"> 1. Reboot the Speaker and 2. Trigger WAC / SAC using the Setup Button (Press and Hold for three seconds) in LS EVK or through Message-Box 142.
Result	LS-Enabled Speaker switches from DDMS Mode and enters WAC / SAC mode for Network Configuration.

Scenario 6

Precondition	LS-Enabled speaker is in DDMS SA- Mode and was connected to AP earlier.
Action	<ol style="list-style-type: none"> 1. Reboot the Speaker and 2. Trigger WAC / SAC using the Setup Button (Press and Hold for three seconds) in LS EVK or through Message-Box 142.
Result	LS-Enabled Speaker switches from DDMS Mode and connects to the AP configured earlier.

Scenario 7

Precondition	LS-Enabled speaker is in DDMS HN-Mode and was connected to AP earlier.
---------------------	--

Action	1. Reboot the Speaker and 2. Trigger WAC / SAC using the Setup Button (Press and Hold for three seconds) in LS EVK or through Message-Box 142.
Result	LS-Enabled Speaker switches from DDMS HN-Mode and enters WAC / SAC mode for Network Configuration.

Example,

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	0142	NA	Variable	0	-

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	0142	SUCCESS / FAILURE	Variable	0	-

6.67.Message-Box 143 (Network Configuration Status)

Message-Box 142 is the **Network Configuration Status** Message-Box. This Message-Box is used by the LS to send update for the Network Configuration Status to the HOST-MCU.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	0143	NA	Variable	Length of data field	nw_config_status

nw_config_status values:

Status	Value
Initializing	0
Ready	1
Configuration Start	2
Configuration Received	3
Configuration Complete	4
Stop	5
Error	6
WAC_Timeout_Message_Timeout	7
WAC_Message_ConfigFailed	8
SAC Failed	9



Device may send WAC Stop when the WAC is triggered. This is because WAC triggered previously is not cleared.

6.68.Message-Box 144 (Stop WAC)

Message-Box 144 is used to **STOP WAC** which was already triggered. HOST-MCU/Phone App sends the message over Message-Box #144 and LS in turn responds back with the **SUCCESS/FAILURE** Status of **STOP WAC**.

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	144	NA	Variable	0	-

Response,

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	144	SUCCESS / FAILURE	Variable	0	-

6.69.Message-Box 150 (Factory Reset)

Message-Box 150 is the **Factory Reset** Message-Box. HOST-MCU/Phone App sends the message with cmd_type SET over MB#150 and LS reboots itself to the default factory settings.

For Example,

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	150	0	Variable	0	-

Response,

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	150	SUCCESS / FAILURE	Variable	Length of data	-

6.70.Message-Box 151 (RSSI Indicator)

Message-Box 151 is the **RSSI Indicator** Message-Box. HOST-MCU/Phone App sends the message with cmd_type GET over MB#151 and LS in turn responds back with the current RSSI level in dBm.

For Example,

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	151	0	Variable	0	-

Response,

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	151	SUCCESS / FAILURE	Variable	Length of data	data

Data field contains comma separated RSSI values in dBm for both the antenna.

6.71. Message-Box 207 (IOT Control Commands)

Message-Box 207 is the **IOT Control Commands** Message-Box. This Message-Box is used to set commands to control the LibreSync Device over Wi-Fi.

For Example,

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	207	0	Variable	Length of data Field	IOT Control Commands

Valid IOT Control Commands is as below

Wi-Fi Switch Control

- RELAY:ON - To Turn ON the device connected to plug
- RELAY:OFF - To Turn OFF the device connected to plug

RGB Lamp Control

- LED_DEVICE:ON - To Turn ON the RGB Lamp
- LED_DEVICE:OFF - To Turn OFF the RGB Lamp
- LED_RGB:XXYYZZ - To Control RGB Color
 - XX represents RED component value in hexadecimal
 - YY represents GREEN component value in hexadecimal
 - ZZ represents BLUE component value in hexadecimal
- LED_INTENSITY:XX - To Control Intensity
 - XX represents intensity component value in hexadecimal
 - Value range is 00 to 0A
- LED_DYNAMIC:ON – To Turn ON the Dynamic Mode
- LED_DYNAMIC:OFF – To Turn OFF the Dynamic Mode

- LED_FLASHING:ON – To Turn ON the Flashing Mode
- LED_FLASHING:OFF – To Turn OFF the Flashing Mode

6.72.Message-Box 208 (NV Read/Write)

Message-Box 208 is the **NV Read/Write** Message-Box. This Message-Box is used to read & Write the Device NV-items.

For Example to Read,

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	208	SUCCESS/ FAILURE	Variable	Length of Data Field	READ_ENVName

Data Format: In short, host need to write READ_<nv name> to message box 208 and on success response can read 208 to get NV-item value.

Example: getenv DeezerUserName

```

Command,

#getenv DeezerUserName

LUCI_local 208 READ_DeezerUserName


Return,

DeezerUserName : abcd123@librewireless.com

```

Response

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	208	SUCCESS/FAILURE	Variable	Length of Data Field	NV content

For Example to Write,

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	208	SUCCESS/FAILURE	Variable	Length of Data Field	WRITE_ENVName,ENVValue

Data Format: In short, host need to write WRITE_<nv name>,ENVValue to message box 208 and on success response can read 208.

"," is the separator between env name and env value.

Example: setenv netif eth0

Command,

setenv netif eth0

LUCI_local 208 WRITE_netif,eth0

Response

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	208	SUCCESS/FAILURE	Variable	Length of data	ENVName:ENVValue

6.73. Message-Box 209 (Bluetooth Control Command)

Message-Box 209 is the **Bluetooth Control Command** Message-Box. This Message-Box is used to set commands to control the LibreSync Device over Bluetooth.

For Example,

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	209	0	Variable	Length of data Field	Bluetooth Control Commands

Valid Bluetooth Control Commands is as below

- ON – To Turn ON the Bluetooth
- OFF – To Turn OFF the Bluetooth
- ENTPAIR - Makes the device discoverable. Disconnects current BT connection if already in connected state.
- EXTPAIR - Makes the device to enter non-discoverable mode.
- DISCONNECT - Disconnects the current Bluetooth connection.
- AUTOCON – Automatically connects to the paired Bluetooth device.
- QUERY – Sends a query to Bluetooth Device.

- CLEARALL – Clears all the BT paired connections
- SETLOCALNAME,BT_Librespeaker – To give a name to the BT Device.
- GETFWVERSION– To request the firmware version of LSx module
 - Response to this command is **FWVERSION,xxx**. Where, xxx is the version number.
- btconnect,xxxxxx – LS sends the MAC Id of the Bluetooth connected device to HOST-MCU in format **btconnect,xxxxxx**. Where, xxxxxx is the MAC ID of the device.
- btdisconnect – LS sends btdisconnect to HOST-MCU when the Bluetooth device is disconnected.
- GETREMOTENAME – To know the name of the remote device. This is applicable on to LS9 Module.
- GETLOCALBTADDR– LS sends the command to know the address of the local Bluetooth device.
HOST-MCU sends the address of the local Bluetooth device with LOCALBTADDR,xxxxxx. Where, xxxxxx is the MAC ID of the device.
- ENTCONNECTABLE: To set the device in connectable state.
- EXTCONNECTABLE: To set the device in not connectable state.
When this EXTCONNECTABLE command is sent, the device already paired with, also is not connectable.

6.74. Message-Box 210 (DMR restart)

Message-Box 210 is the DMR restart Message-Box. This Message-Box is used for restarting the DMR and hence it makes the device to go out of UPnP list temporarily and will appear back in the UPnP device list in short time.

For Example,

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	210	0	Variable	0	-

Based on this command, the LSx sends out UPnP *ssdp:byebye* packets and device will get de-listed on the UPnP controller end. Within short time, *ssdp:alive* packets will be sent, and device capabilities will be notified.

Response

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	210	SUCCESS/ FALIURE	Variable	0	-

6.75. Message-Box 211 (Start Firmware Upgrade)

Message-Box 211 is Start Firmware Upgrade Message-Box. On request for Firmware Upgrade from LS over Message-Box 69, HOST-MCU either starts the firmware upgrade process or denies the firmware upgrade.

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	211	NA	Variable	Length of data Field	data

Values of data field:

Value (Case Sensitive)	start	no	START	CANCEL
Description	When the firmware upgrade is via SD-Card, USB, Network and Internet, triggered from webpage	When the firmware upgrade is via SD-Card, USB, Network, and Internet, triggered from webpage	When the firmware upgrade is via Internet and LS sends notifications as 1/2/3 to HOST-MCU over Message-Box #69	When the firmware upgrade is via Internet and LS sends notifications as 1/2/3 to HOST-MCU over Message-Box #69

Steps involved in confirmation to start the firmware upgrade request over Message-Box # 69 is as below.

Web Server

SD-Card / USB / Network

- Step 1.** Firmware upgrade trigger for SD-Card / network / USB from webserver
- Step 2.** LS sends the "**RequestForFirmwareUpgrade**" to the HOST-MCU over Message-Box# 69
- Step 3.** HOST-MCU takes a decision whether it has to *Allow* or *Deny* the firmware upgrade.
- Step 4.** Send a confirmation on **Message-Box#211** with data field set to "**start**" / "**no**" to *Allow* or *Deny* the firmware upgrade.

End --

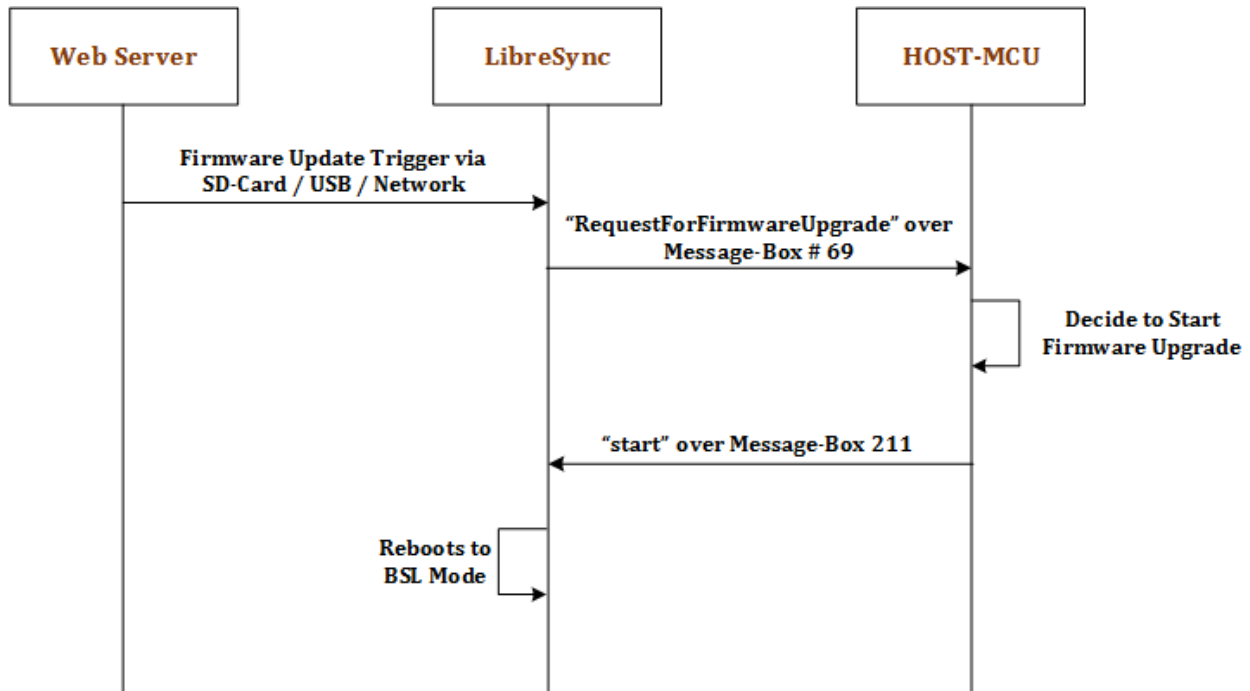


Figure 6-8 a: SD-Card / USB / Network Method of Firmware upgrade from Web Server

Internet

- Step 1.** Firmware upgrade trigger for Internet from webserver
- Step 2.** LS sends the "***RequestForFirmwareUpgrade***" to the HOST-MCU over Message-Box# 69
- Step 3.** HOST-MCU takes a decision whether it has to *Allow* or *Deny* the firmware upgrade.
- Step 4.** Send a confirmation on ***Message-Box#211*** with data field set to "***start***" / "***no***" to *Allow* or *Deny* the firmware upgrade.
- Step 5.** If the data is sent as ***start***, LS triggers **Message-Box #65** internally to start Internet method of upgrade.
- Step 6.** LS sends the following message over ***Message-Box 69*** to HOST-MCU
 - 1 LS sends "***2***" if the download of ***fwdownload.xml***, file fails.
 - 2 LS sends "***1***" if parsing of ***fwdownload.xml***, file fails.
 - 3 LS sends "***3, <NEW/OLD/SAME>, <New Fw Version>, <New Host Version>***" after successful completion of download and parsing of ***fwdownload.xml***, file.

Note: LS sends the notifications to HOST-MCU, only when the **fwinternet_host** NV-Item is set as **1** in the FENV.xml file.

Step 7. HOST-MCU takes a decision whether it has to *START/CANCEL* the firmware upgrade.

Step 8. HOST-MCU sends a confirmation on **Message-Box#211** with data field set to "*START*" / "*CANCEL*" to start or stop the firmware upgrade.

End --

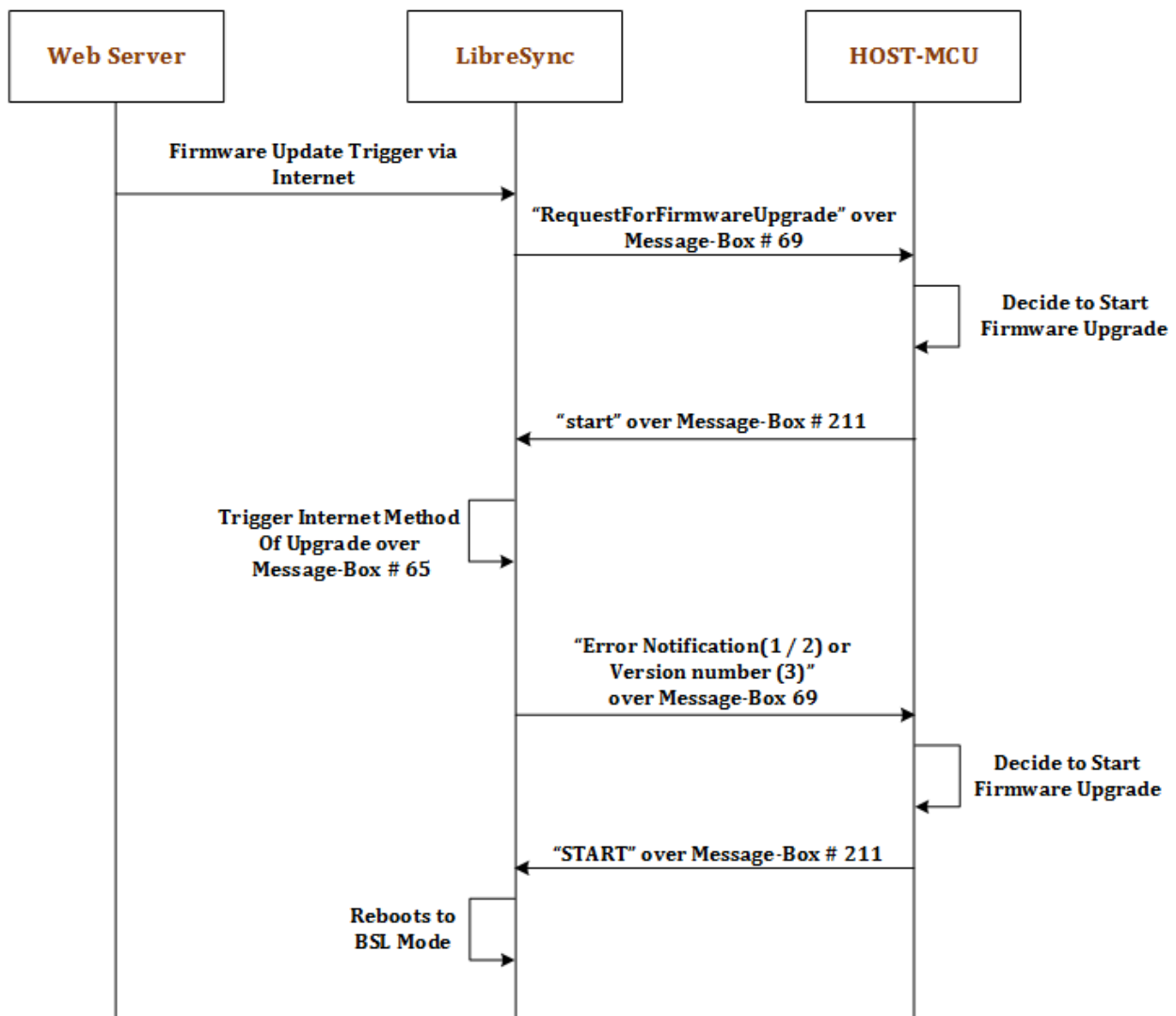


Figure 6-8 b: Internet Method of Firmware upgrade from Web Server

HOST-MCU

SD-Card / USB / Network

Step 1. Firmware upgrade trigger for SD-Card / network / USB from HOST-MCU over **Message-Box 65**

Step 2. LS reboots to BSL and starts the firmware upgrade process.

End --

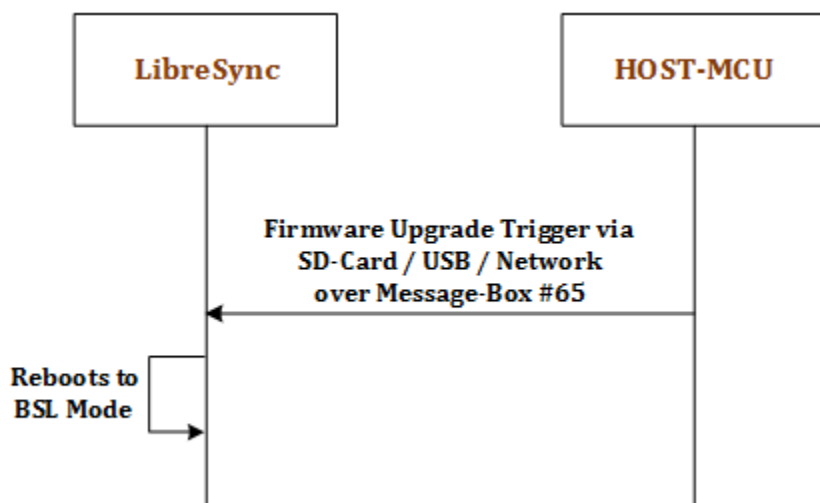


Figure 6-8 c: SD-Card / USB / Network Method of Firmware upgrade from HOST-MCU

Internet

Step 1. Firmware upgrade trigger for Internet from HOST-MCU over **Message-Box #65**.

Step 2. LS sends the following message over **Message-Box 69** to HOST-MCU

- 1 LS sends "2" if the download of **fwdownload.xml**, file fails.
- 2 LS sends "1" if parsing of **fwdownload.xml**, file fails.
- 3 LS sends "3, <NEW/OLD/SAME>, <New Fw Version>, <New Host Version>" after successful completion of download and parsing of **fwdownload.xml**, file.



LS sends the notifications to HOST-MCU listed in the above steps, only when the **fwinternet_host** NV-Item is set as **1** in the FENV.xml file.

Step 3. HOST-MCU takes a decision whether it has to *START/CANCEL* the firmware upgrade.

Step 4. HOST-MCU sends a confirmation on **Message-Box#211** with data field set to "*START*" / " *CANCEL*" to start or stop the firmware upgrade.

End --

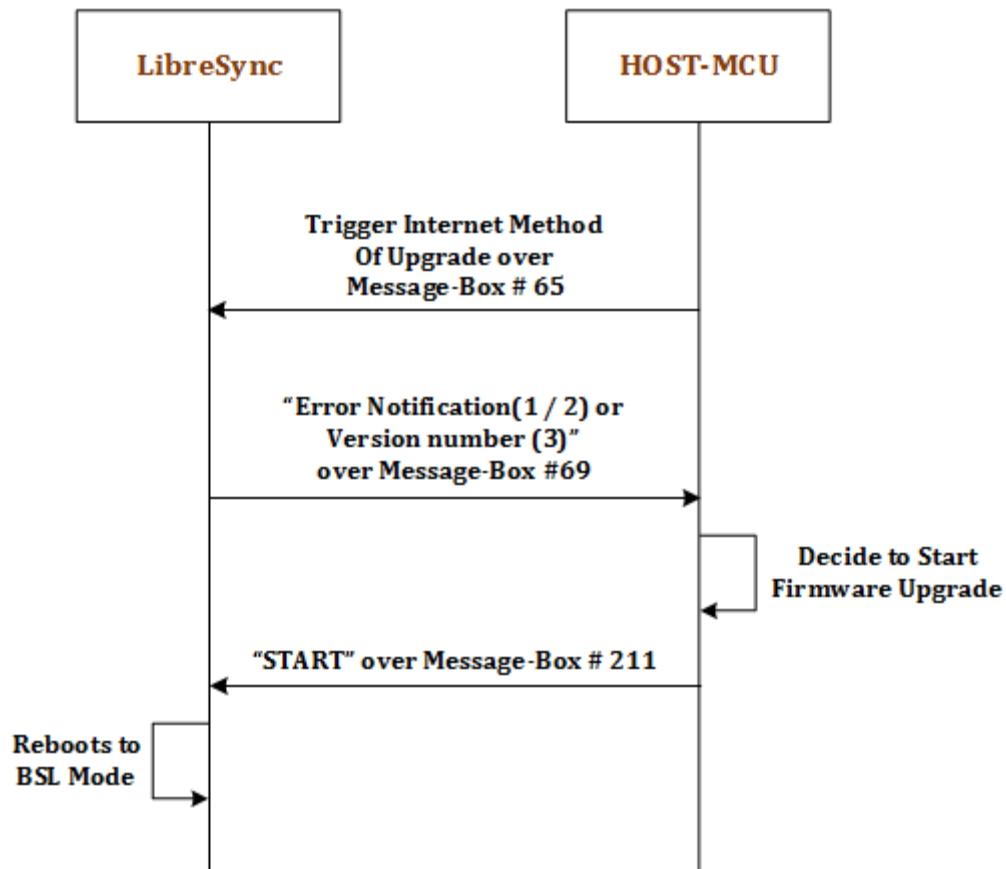


Figure 6-8 d: Internet Method of Firmware upgrade from HOST-MCU

6.76. Message-Box 213 (Name Password Notifier)

Message-Box 213, is **Name Password Notifier** message-box. This message-box is common for all the online music services. Message-Box is used to notify LS the user credentials such as User name and Password to access particular music service.

Currently this message-box 213 is used to notify the user credentials of Deezer and Tidal.

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	213	0	Variable	Length of Data	MUSICSERVICE_User Name MUSICSERVICE_Password

Data Format

MUSICSERVICE: Provide the name of music service.

User Name: End User login name

Password: End User login password.

Example,

DEEZER_divya@librewireless.com

DEEZER_Deezerlogin

Response

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	213	Success / Failure	Variable	0	-

6.77.Message-Box 214 (Enable-Share-Mode)

Message-Box 214 is **Enable-Share-Mode** message-box. This message-box is used to enable sharing of network credentials between two LS-enabled devices.

- Share mode can be triggered only when the LS-Device is Network-Master, otherwise share command is invalid.
- LS-Module will send SHARE_MODE_INVALID value over message-box 214, if Share is triggered on the Network-client device.
- Once Triggered Share Mode is active for duration of 2 minutes in LS5B modules and 30 minutes in LS6 Modules.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	214	NA	Variable	Length of Data	Data

Response

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	214	Success / Failure	Variable	Length of Data	Data

6.78.Message-Box 215 (Enable-Pair- Mode)

Message-Box 215 is Enable-Pair-Mode message-box. This message box enables the LS-enabled device to pair with another LS-enabled device (in Share-Mode) that broadcasts the network credentials.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	215	NA	Variable	Length of Data	1 or 0

Data

- **0**: Pairing-Device will look for Share-Device which is in Stand-Alone mode.
- **1**: Pairing-Device will look for Share-Device which is in Home-Network mode.

To know if the Pairing-Device is in Home-Network Mode or in Stand-Alone Mode, check Message-Box 124.

- If the device is in Home-Network Mode, then data to be sent by HOST-MCU to LS is **1**, over Message-Box 215.
- If the device is in Stand-Alone Mode, then data to be sent by HOST-MCU to LS is **0**, over message-box 215.

Response

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	215	Success / Failure	Variable	Length Of Data	1 or 0

6.79.Message-Box 216 (SLAVEINFO)

Message-Box 216 is used to share the Slave Information with the master. Message-Box 216 is used to receive the slave device name in case of Multi-Room Audio for Soptify.

The Master Device name will be combination of all device names in the audio group, starting with the Master's name. Slave speakers will not be discovered in Spotify App.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	216	NA	Variable	Length of Data	M+S1+S2+....

Data

Where,

M = Master Device Name

S1 = Slave 1 Device Name

S2 = Slave 2 Device Name, and so on.

6.80. Message-Box 219 (Zone Volume Control)

Message-Box 219 is Zone Volume Control Message-Box. This message-box is used to control the volume of a scene in DDMS.

Zone Volume Control uses scaling ratios to provide a good user experience and intuitive system response based on controlling volume levels for dynamic groups of speakers - called "Sound Scenes".

Zone volume control manages an overall Sound Scene's Volume. Once speakers are adjusted, the Sound Scene Volume will raise and lower each speaker's volume in a relative way keeping the relationship of volume level between speakers. However as you lower the Sound Scene's Volume level to zero, every speaker in the Scene will converge to zero at the same time.

If there are 2 devices in a Sound scene, and the HOST-MCU on the first device changes the zone volume or device volume, then LS sends the updated zone volume to both the hosts using the same Message-Box.

Other considerations.

- If all the device's volume is manually set to zero, then changing the zone volume will only affect the volume of the device which was the last to go to 0.
- If the device that last went to 0 is removed from the group, then changing the zone volume will affect only the group master.
- If all devices reboot with current volume as zero, changing the zone volume will only affect the volume of the group master.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	219	NA	Variable	Length of data field	Volume_level

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	219	NA	Variable	Length of data field	Volume_level

6.81. Message-Box 221 (Pair Status)

Message-Box 221 is Pair Status Message-Box. This message-Box is used to notify HOST-MCU progress of the Pair-Mode trigger.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	221	NA	Variable	Length of Data	Pairing Status

Data

- PAIR_IN_PROGRESS – LS sends PAIR_IN_PROGRESS when, pair-mode for the device is initiated.
- PAIR_RETRY – LS sends PAIR_RETRY when, the pairing fails and devices retries to connect to the master.
- PAIR_SUCCESS – LS sends PAIR_SUCCESS when, the devices successfully connects to the master.
- SHARE_IN_PROGRESS - Share functionality is triggered.
- SHARE_DISABLED - Share mode is active for specific time interval (LS5B: 7 minutes, LS6: 10 minutes) and will turn off after that. HOST-MCU will receive a message SHARE_DISABLED after the Share timeout.
- SHARE_MODE_INVALID – Share Mode Invalid message is sent by LS to HOST-MCU, when Share is triggered on the client device.
- PEER_DEVICE_CONNECTED – Peer device is connected to share device.

6.82. Message-Box 222 (Cast OTA Update) - LS9 Module only

Message-Box 222 is **Cast OTA Update** Message-Box. This Message-Box is used to send the status of the Cast Firmware update of the device via OTA Upgrade. LS- Device sends the data to HOST-MCU / APP over this message-box.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	222	NA	Variable	Length of Data	OTA Data

Data

- **4:** Force Upgrade, HOST-MCU should get ready for immediate reboot.
- **2:** Ready for Restart
- **1:** Downloading OTA image in background.

6.83. Message-Box 223 (Firmware Upgrade INTERNET) – LS9 Module only

Message-Box 223 is Firmware Upgrade INTERNET Message-Box. This message-box is used to send notification to the HOST-MCU and Network Controller APP, on the availability of firmware.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	223	NA	Variable	Length of Data	Data

Data

- **UPDATE_STARTED:** This data is sent when internet update has started automatically from a separate source (Not from LUCI/HOST command).

- **UPDATE_IMAGE_AVAILABLE:** This data is sent when the LS9 firmware image (83_IMAGE_network) for Internet method of upgrade is downloaded from the server.
- **NO_UPDATE:** This data is sent when the update is not available, or fails to download the URL.
- **CRC_CHECK_ERROR:** This data is sent when the downloaded image CRC and the CRC written in the *<crc32check>* of the XML file used for internet upgrade do not match.
- **DOWNLOAD_FAIL:** This data is sent when the 83_IMAGE_network download is failed.

6.84. Message-Box 224 (Cast is Enabled) - LS9 Module only

Message-Box 224 is CAST is Enabled message-box. Message Box 224 is used to start/stop GoogleCAST. On CAST start the HOST will receive a response with values ranging from 0-13.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	224	NA	Variable	Length of Data	Data

Data

Value	Description
START	To start CAST on LS-Enabled devices.
STOP	To stop CAST on LS-Enabled devices.

Value	Description
0	CAST Successful.
1	Unknown error.
2	CAST shell is not reachable.
3	CAST control is not compatible to CAST shell. CAST Control or CAST Shell should be updated.
4	Response for request has not arrived in time.
5	libcastcontrol has not implemented this request
6	A user action was reported when no media session was available.
7	The current application does not support the requested namespace.
8	Multi-zone is not enabled in CAST shell.
9	The request contained poorly-formatted or invalid data.
10	Response arrived but was formatted incorrectly
11	Receiver is not accepting resources.
12	Started, setup, but suspended by caller. CAST is NOT functional.
13	Ready to CAST. CAST is functional.
14	CAST state is not setup. CAST is not functional.
15	CAST state is offline. CAST is not functional.

6.85.Message-Box 226 (Google Cast Settings Info) – LS9 Modules Only

Message-Box 226 is Google CAST Settings Info message-box. This message-box is used to manage the Google CAST settings for TOS, Share Usage Data and Time Zone.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	226	NA	Variable	Length of Data Field	Data

Data

<TOS SHARE status>:<Timezone>

- <TOS SHARE status> indicates whether TOS and Share is accepted by user and is an integer which can have value between 0 and 3.
 - 0: No TOS and No Share is accepted
 - 1: only TOS is accepted
 - 2: only Share is accepted
 - 3: both share and TOS is accepted
- <TIMEZONE> indicates the time-zone in String format. Example: Asia/Calcutta.

APP can GET the current configuration values, and can inform the device when some settings is changed. Device will also notify the APP when the settings is changed.

APP should strictly follow Googles guidelines on implementing this.



On SET, its mandatory to provide the config values of all the fields. Say user changes <Timezone>.

on setting this, the current <TOS SHARE status> should also be provided along with the new <Timezone>. Device reads all the fields, compare the existing configuration and changes the updated ones.

6.86. Message-Box 227 (DMR STOP)

Message-Box 227 is **DMR STOP** message-box. This message box is used to stop DMR discovery. Message-Box 227 along with Message-Box 228 can be used to avoid hijacking by DMR during source switch.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	227	NA	Variable	Length of Data	STOP

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	227	Success / Failure	Variable	Length of Data	STOP

6.87. Message-Box 228 (DMR START)

Message-Box 228 is **DMR START** message-box. This message box is used to START DMR discovery, after DMR Discovery is stopped.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	228	NA	Variable	Length of Data	START

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	228	Success / Failure	Variable	Length of Data	START

DMR Discovery using MB# 227 and MB# 228

DMR discovery will happen in two scenarios

- Source switch from DMR to DMR
- Source switch from DMR to other Sources.

If current playing source is DMR and source switch is requested, HOST-MCU has to follow the below procedure to start and stop DMR Discovery.

- Step 1.** When new source is playing, LS-Enabled Device will send the new source to MB#10 (isAllowed).
- Step 2.** If the current playing source is DMR, HOST-MCU should send MB# 227.
- Step 3.** HOST-MCU should send allowed on MB# 11.
- Step 4.** When new source playback is started LS-Enabled device will update MB# 51 with playing status.
- Step 5.** On receiving MB# 51 HOST-MCU should send MB# 228.

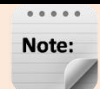
Source Switching between DMR and QPlay

- QPlay uses DMR discovery, so when the new audio source is QPlay and current playing source is DMR, DMR discovery should not be stopped.
- QPlay can hijack DMR; when source is switched from DMR to QPlay, DMR will not hijack again so there is no need to send MB# 227 and MB# 228 for this case.

6.88. Message-Box 229 (GET NTP Time)

Message-Box 229 is the **GET NTP Time** message-box. This message-box is used to share the NTP time to HOST-MCU from LS-Module. This message-box sends the data successfully only when the Internet connectivity and Network Interface is active.

Based on the EPOCH value sent from LS-Module, HOST-MCU should calculate the actual Date and Time.



This message-box is not applicable for LS9.

Command (HOST-MCU to LS)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	229	NA	Variable	Length of the data field	Data

Command Data

Value	ntp
Description	HOST-MCU sends the data as ntp to LS, to request the NTP time

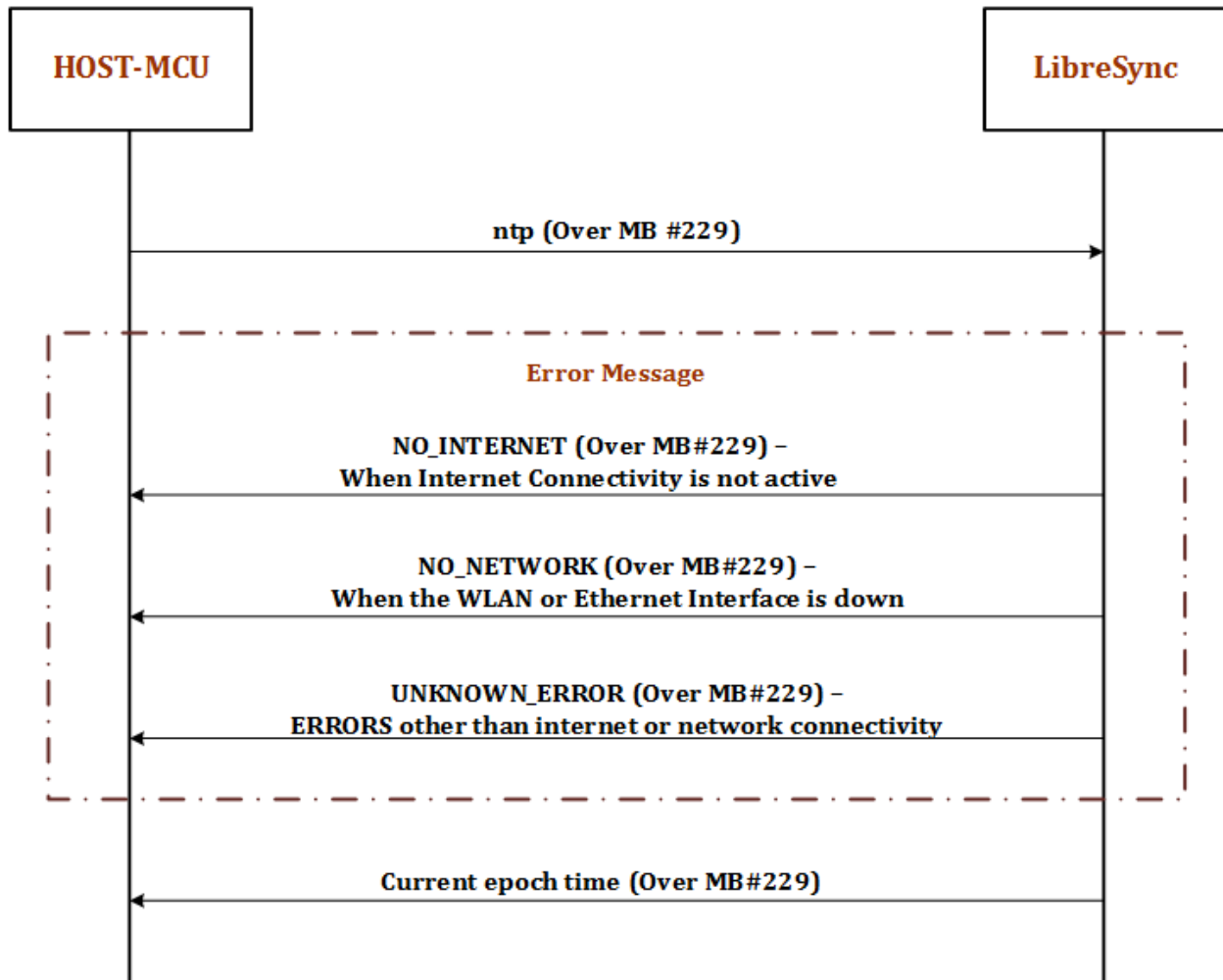
Response (LS to HOST-MCU)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	229	NA	Variable	Length of the data field	Data

Response Data

Value	<<Current epoch time>>	NO_INTERNET	NO_NETWORK	UNKNOWN_ERROR
Description	LS send the epoch time to HOST-MCU. HOST-MCU should calculate the actual Date and Time.	LS sends an error message NO_INTERNET if the internet connectivity is not active	LS sends an error message as NO_NETWORK when WLAN or Ethernet interface is down.	LS sends an error message as UNKNOWN_ERROR if there are any issues other than the internet or network connectivity.

Data Sequence



6.89. Message-Box 230 (Audio Output FS) – LS9 Module Only

Message-Box 230 is **Audio_Output_FS** Message-Box. This message-box is used by LibreSync-Module to notify the sampling frequency (I2S-LRClock) to HOST-MCU, every time the sampling frequency changes.

In LS9-Module

- LS-Module sets the I2S-LRClock according to the audio-track being played.
- The updated output sampling frequency is informed to the HOST-MCU.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	230	SUCCESS / FAILURE	Variable	Length of the data field	Data

Command Data

- 44100
- 48000
- 88200
- 96000
- 176400
- 192000

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	230	SUCCESS / FAILURE	Variable	Length of the data field	NA

6.90. Message-Box 231 (gCast Serial Number) – LS9 Module Only

Message-Bix 231 is GCAST Serial Number message-box. This message-box is used to read the unique GCAST serial number of the device.

Command (HOST-MCU to LS9 or Network Client to LS9)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	231	NA	Variable	Length of Data (Max 20 Char)	Text String

Response (LS9 to HOST-MCU or LS9 to Network Client)

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	231	Success / Failure	Variable	Length of Data (Max 20 Char)	Text String

6.91.Message-Box 232 (AC_POWERED) - LS9 Module only

Message-Box 232 is **AC_POWERED** message-box. This is applicable if the product is running in Battery. HOST-MCU can check either Battery threshold or AC power for product and set the MB#232 so that Firmware update can happen only if Battery threshold is more or AC powered. LS will use this message-box before firmware download/update starts. Firmware is applicable to both LS and HOST-MCU. By default, value of the message-box is TRUE.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	232	NA	Variable	Length of the data field	Data

Command Data

Value	Description
TRUE	The Speaker is AC powered and Firmware download/update can happen
FALSE	The Speaker is NOT AC powered (running in battery) and Firmware download/update cannot happen

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	232	SUCCESS / FAILURE	Variable	Length of the data field	Data

6.92.Message-Box 233 (MIC Control)



This Message-Box is applicable for LS6 Module only.

Message-Box 233 is **MIC Control** message-box. This message-box is used to send MIC Control messages between LS and HOST-MCU.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	233	NA	Variable	Length of Data	Data

Data

Data	Description	Data Flow
START_MIC	Indicates LS to start capturing the voice. Typical scenario would be like button press or Tap or when mic detects "Alexa" key word.	HOST-MCU → LS
MIC_STARTED	Indicates the HOST-MCU that LS has started listening to user's voice, HOST-MCU can use this command to show LED indication.	LS → HOST-MCU
STOP_MIC	Indicates LS to stop capturing the voice. Typical scenario would be like button press or when user releases the tap or when mic detects silence.	HOST-MCU → LS

Data	Description	Data Flow
MIC_STOPPED	Indicates the HOST-MCU that LS has stopped listening to user's voice, HOST-MCU can use this command to stop LED indication	LS → HOST-MCU

Response

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	233	Success / Failure	Variable	Length of Data	Data

6.93. Message-Box 234 (AVS_APP_SERVICE)

Reserved

6.94. Message-Box 236 (Cloud Tunnelling Request) – LS6 / LS5B Only

Message-Box 236 is Cloud Tunnelling Request message-box. This message-box is used by HOST-MCU to request LSx to download the content from remote server via HTTP/ HTTPS protocol. To use this feature, MCU shall send first URL of remote server, Http GET or POST command followed by GETDATA command in sequence.

Command

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	236	SUCCESS / FAILURE	Variable	Maximum 2560	HTTP/HTTPS protocol Commands

HTTP/HTTPS protocol Commands format

Command	command format	Description
URL	URL:Url,<Header1>,<Header2>,<data>	Url of the remote server, Header1 (optional), Header2 (optional), data to be uploaded (only for METHOD:POST)
METHOD	METHOD :<GET/POST>	HTTP/HTTPS request type GET or POST
GETDATA	-	Ready to receive the data from LS

URL Request Parameters

URL Parameter	Size in Bytes	Description
HTTP_REQUEST_MAX_SIZE	2560	URL length + URL data length + (2 x Headers size)
HTTP_RESPONSE_MAX_SIZE	1024	Maximum size of the response from server (data to be sent to HOST-MCU as a response to a request)
URL_LENGTH_MAX_SIZE	1024	Url with access token
URL_DATA_MAX_SIZE	1024	Maximum size of the data to be uploaded
HTTP_HEADER_MAX_SIZE	256	Maximum size of the header information
ACCESS_TOKEN_MAX_SIZE	512	Maximum size of the access token

Below are the few Example GET or POST Request (HOST-MCU to LSx)

- GET Request (Here Header1 and Header2 are not shown. Header1 or Header2 can be optional)

- Data:

URL:https://librewireless.com?access_token=inrbiU7iETEXnvdPtdJFbo7k6TgchbsMQSzOA3iK9EyaOxkjRZvXrjVGj66cEP3xk

METHOD:GET
- Response:

SUCCESS,<size of the content> or

ERROR,<error code> - Refer below table for types of error codes
- POST Request (Here Header2 is not shown. Header1 or Header2 can be optional)
 - Data:

URL:[https://librewireless.com/events?access_token=QSzOA3iK9EyaOxkjRZvXrjVGj66cEP3xk,HEADER1:Contenttype:application/json,DATA:{\"data 512\\\": \\\"ztwTC4ysdRdTVAtwcG8jFakQVVhwyWiubqRyPWG0nX00WQM0meo\\\"}](https://librewireless.com/events?access_token=QSzOA3iK9EyaOxkjRZvXrjVGj66cEP3xk,HEADER1:Contenttype:application/json,DATA:{\)

METHOD:POST
 - Response:

SUCCESS,<size of the content> or

ERROR,<error code> - Refer below table for types of error codes
- GETDATA Ready to receive the data from LS which is received from Cloud
 - Data:

GETDATA
 - Response:

Received in MB 237

Error Codes

Error Code	Error Name	Description
101	ERROR_HOST_REQUEST_FORMAT	Request from host is not in proper format
102	ERROR_SERVICE_NOT_SUPPORTED	Other Web Services

Error Code	Error Name	Description
103	ERROR_INVALID_PROTOCOL	WS or WSS Requests
104	ERROR_DNS_LOOKUP_FAIL	Error in resolving the HOST-MCU name
105	ERROR_DOWNLOAD_FAIL	Failed to download the data from server
106	ERROR_MEMORY_NOT_SUFFICIENT	Response size is greater than 1K
107	ERROR_NO_INTERNET_AVAILABLE	Device is not connected to network
108	ERROR_NO_NETWORK_AVAILABLE	No valid interface is up
201	ERROR_INTERNAL	Internal error
200	URL_REQUEST_SUCCESS	URL request Success

6.95. Message-Box 237 (Cloud Tunnelling Response) – LS6 / LS5B Only

Message-Box 237 is the Cloud Tunnelling Response Message-Box. This Message-Box is used by LS to send the downloaded URL content requested via MB#236 to HOST-MCU.

Command (LSx to HOST-MCU)

Unique Remote ID	Command Type	Message Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	237	SUCCESS / FAILURE	Variable	Maximum of 1024	URL content

Data

- Success,<size of the content>
- Error,<error code>

6.96. Message-Box 494 (CAST SETUP STARTED) - LS9 Module only

Message-Box 494 is **CAST SETUP STARTED** message-box. In this message-box LS-Enabled Device sends CAST SETUP started notification to HOST-MCU. On receiving this message HOST-MCU should change the source of LS-Enabled Device to Libre Wi-Fi.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	GET	494	NA	Variable	Length of Data Field	CASTSetupStart

6.97. Message-Box 43690 (0xAAAA) (UART Interface Ready)

Message-Box 43690 (0xAAAA) is the **UART Interface Ready** Message-Box. This is a special Message-Box and only applicable to UART.

Once LS platform boots up and initializes all the modules responsible for the LS framework to work, LS signals through this Message-Box that the HOST-MCU can initiate the UART communication. The data sent over this Message-Box from LS to HOST-MCU will be a fixed data '0xAAAAAAAAAAAA' (10 times hex A).

HOST-MCU can start the communication as soon as it reads 0xAAAAAAAAAAAA data in the data payload.

Message-Box returns SUCCESS in case it reads the predefined data and now onwards HOST-MCU can initiate the normal communication with LS framework.

Message-Box returns FAILURE in case the HOST-MCU does not get the predefined data.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	43690 (0xAAAA)	NA	Variable	Length of Data Field	0xAAAAAAAAAAAA

Response

The HOST-MCU must send a response for this over Message-Box#9 for the HOST-MCU present status.

6.98. Message-Box 43981 (0xABCD) (LS-IN-BSL Indication)

Message-Box 43981 (0xABCD) is the **LS-IN-BSL Indication** Message-Box. LS sends to HOST-MCU about LS module booted in BSL.

After reboot Host-MCU waits for Message-Box # 0xAAAA.

Command

Unique Remote ID	Command Type	Message-Box ID	Command Status	CRC	Data Length	Data
0xAAAA	SET	43981 (0xABCD)	NA	Variable	Length of Data Field	0xAAAAAAAAAAAA

7. Appendix

7.1. Acronyms and Abbreviations

For details on acronyms and abbreviations used in the document see

“LibreWirelessTechNote_Acronyms_And_Abbreviations_1.0”