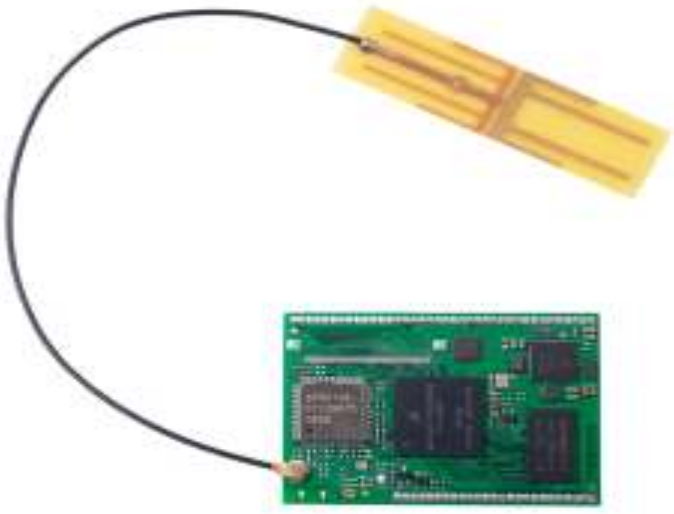
LinTech

Wireless Audio Module

HBM11

v6.4.1



Date: 2022-06-01
Revision: 1

Contents

1 ChangeLog 5
2 Introduction 10
2.1 Target Applications 10
2.2 Software Features 10
2.3 Board Features 11
2.4 Module Block Diagram 11
3 System Specifications 12
3.1 General 12
3.2 RF Specification 13
3.2.1 Wi-Fi 2.4 GHz 13
3.2.2 Wi-Fi 5 GHz 13
3.2.3 Bluetooth 14
4 Mechanical Specifications 15
4.1 Module Dimensions 15
4.2 Pin Layout 15
4.3 Pin Header 16
4.4 Antenna 16
5 Pin Descriptions 17
5.1 Overview 17
5.2 Pin Definitions 17
5.3 Pinout 19
6 Interfaces 20
6.1 I2S 20
6.2 Line-Out 21
6.3 Line-In 21
6.4 S/PDIF 22
6.5 Ethernet 22
6.6 USB 23
6.7 LEDs 23
6.8 Keys 24
6.9 GPIOs 25

6.10 Suspend Pin 25
6.11 Audio Status Pin 26
6.12 I2C 26
6.13 UART 26
7 Network 27
7.1 Ethernet 27
7.2 Bluetooth 27
7.3 WLAN 27
7.3.1 Operation 27
7.3.2 Configuration 29
7.3.3 Static IP 29
7.3.4 Roaming 29
7.4 Zeroconf 30
7.5 WPS 31
7.6 Tools 31
7.6.1 iPerf3 31
8 Audio 33
8.1 UPnP 33
8.1.1 Media Renderer 33
8.1.2 Media Server 33
8.2 OpenHome 34
8.3 AirPlay 34
8.4 AirPlay 2 35
8.5 Internet radio (powered by TuneIn) 35
8.5.1 Features 35
8.5.2 Playback 35
8.5.3 Playlists 35
8.6 Bluetooth 36
8.7 Spotify Connect 36
8.8 TIDAL Connect 36
8.9 Qobuz 36
8.10 Multi-Room 36
8.10.1 AirPlay and AirPlay 2 36

8.10.2 Songcast 38
8.11 Playback control 40
8.11.1 AirPlay 40
8.11.2 UPnP and OpenHome 40
8.11.3 Spotify Connect 40
8.11.4 TIDAL Connect 40
8.11.5 Bluetooth 40
8.12 Volume Control 40
8.13 Channel Setting 41
8.14 Equalizer 41
8.15 Surround Sound 41
9 AirLino® App 42
10 Customization 43
10.1 Device 43
10.2 Wi-Fi 43
10.3 Audio 44
11 Sample Applications 45
11.1 Serial-to-WiFi Bridge 45
11.1.1 Block Diagram 45
11.1.2 Workflow 45
11.1.3 AT Commands Reference 46
11.2 Line-In PCM-Recorder 49
11.2.1 Basic Setup 49
11.2.2 PCM Format 49
11.2.3 Example TCP Server application 50
11.3 Line-In Multi-Room Player 51
11.3.1 Basic Setup 51
11.3.2 PCM Format 51

List of Figures

Figure 1 – HBM11 Hardware Components 11
Figure 2 – Board Dimensions 15
Figure 3 – Recommended Pin Layout 15
Figure 4 – Pin Header 16
Figure 5 – Pin Group Overview 17
Figure 6 – HBM11 Pinout 19
Figure 7 – IS2 connection 20
Figure 8 – I2S Audio Format 20
Figure 9 – Analog Audio Output of the internal DAC 21
Figure 10 – Analog Line-In 21
Figure 11 – SPDIF I/O schematic 22
Figure 12 – Recommanded usage of the Ethernet interface 22
Figure 13 – USB connection 23
Figure 14 – Recommanded usage of the Reset Pins 24
Figure 15 – Recommanded usage of the Audio Pins 24
Figure 16 – Connecting an external power management controller to the SUSPEND pin 25
Figure 17 – Usage of the SUSPEND pin 25
Figure 18 – Connecting an external power amplifier to the AUDIO_STATUS pin 26
Figure 19 – Usage of the AUDIO_STATUS pin 26
Figure 20 – Network startup procedure 28
Figure 21 – Steps to setup a wireless network 29
Figure 22 – Example of an enhanced network configuration with multiple known network 30
Figure 23 – UPnP devices 33
Figure 24 – UPnP Media Server started when plugging in a USB pendrive ord USB disk 34
Figure 25 – Example of an AirPlay server distributing an audio stream with iTunes 37
Figure 26 – Examples of iPhone only supporting a single AirPlay stream 38
Figure 27 – Example of a Songcast Multi-Room Group of one Sender and four Receivers 39
Figure 28 – Example of Songcast Sender broadcasting an audio stream from Spotify 39
Figure 29 – Screenshots of the AirLino App 42
Figure 30 – Sample Application Serial-to-Wifi Bridge 45
Figure 31 – Basic Setup of the Line-on PCM-Recorder Sample Application 49
Figure 32 – Basic Setup of the Line-In Multi-Room Player Sample Application 51

1 ChangeLog

Date	Version	Changes	Author
2022-03-09	6.4.1	1. Fix suspend mode for HBM11 modules with FN-Link-based wifi chips 2. Fix audio status pin behaviour when using TIDAL Connect	Jörg Krause
2022-03-09	6.4.0	1. Add TIDAL Connect support 2. Fix Bluetooth apt-X streaming in multiroom mode 3. Improved radio station single mode playback	Jörg Krause
2021-04-29	6.3.5	1. Fix AirPlay 2 audio distortation caused when synchronizing the audio stream 2. Improved Airplay 2 volume handling: Volume is not fixed to 20 % anymore, but set to the current volume of the HBM11 3. Improved AirPlay 1 audio stream when playback begins: The audio device is not opened, closed for some hundred of milliseconds and then reopened anymore when audio streaming begins. This behaviour might have caused some small “popping” noises.	Jörg Krause
2021-04-12	6.3.4	1. Fix multiroom grouping issue (6.3.3 regression)	Jörg Krause
2021-03-23	6.3.3	1. Fix compatibility with Windows Media Player 2. Fix action “status” of HTTP API v22 endpoint “player” causing iOS app to crash in rare conditions	Jörg Krause
2021-03-01	6.3.2	1. Fix minor UPnP Media Server issues	Jörg Krause
2020-12-22	6.3.1	1. Fix non-function analog line-in (6.3.0 regression) 2. Fix HTTP-API “bluetooth” endpoint if Bluetooth is not available	Jörg Krause
2020-12-17	6.3.0	1. Feature: Optional support of AirPlay 2 2. Fix: Changing device name now also affects Bluetooth name 3. Fix: Improved UPnP album art 4. Security: Fix UPnP “Callstranger” vulnerability 5. Security: Fix Spotify vulnerability 6. API: New HTTP API v22	Jörg Krause
2020-01-27	6.2.1	1. Fix static Ethernet configuration 2. Fix support for compressed digital surround sound audio using the S/PDIF interface 3. Support using non-standard mime types for MP3, like “audio/mp3”	Jörg Krause
2020-01-08	6.2.0	1. New HTTP API v21: a. Add channel configuration support (stereo, both left, both right, mono) b. Add 10 band equalizer support c. Add support for static Ethernet configurations 2. Add aptX decoder support for Bluetooth A2DP 3. Fix static Wi-Fi configuration	Jörg Krause
2019-12-02	6.1.5	1. Fix Wifi/Bluetooth-Coex issue (6.1.4 regression)	Jörg Krause
2019-11-12	6.1.4	1. Fix Wi-Fi issue where multicast, and therefore device and service discovery, does not work anymore until rebooting the device 2. Fix multiroom devices not grouping in some cases after a reboot	Jörg Krause
2019-09-30	6.1.3	1. Fix non-starting UPnP media server in certain cases 2. Improved Wi-Fi stability	Jörg Krause
2019-09-02	6.1.2	1. Fix switching between radio and Bluetooth in multi-room mode 2. Fix setting a device name with certain special characters 3. Fix TIDAL and Qobuz login using the BubbleUPnP app 4. Fix incorrect multi-room synchronization (6.1.0 regression) 5. Use the device name as OpenHome room name instead of “Main Room” to improve compatibility with Linn Kazoo	Jörg Krause

		6. Add “audio/x-mpeg” mime type to supported protocols in UPnP to fix playback issues with Linn Kazoo 7. Improved performance when playing large radio, TIDAL, Qobuz, or UPnP playlist	
2019-08-14	6.1.1	1. Fix enabling multi-room group in certain circumstances 2. Fix silent Bluetooth audio output in multi-room mode 3. Improved Wi-Fi channel scanning 4. Improved radio playback when streaming session is interrupted	Jörg Krause
2019-07-11	6.1.0	1. Support for Wi-Fi 2.4 / 5 GHz variant 2. Add USB Host support: Attaching an external USB pendrive or disk will automatically start a UPnP Media Server 3. Improved I2S interface a. Full support for 44.1, 88.2, and 176.4kHz (no internal hardware resampling required anymore) b. HBM10 compliant MCLK and BCLK 4. Improved AirPlay playback quality 5. New HTTP API v20: a. Support Bluetooth configuration b. Support analog and digital line-in	Jörg Krause
2019-04-15	6.0.2	1. Fix SPDIF volume control when using the AirLino Pro baseboard 2. Fix non-playing status volume after first boot 3. Fix mono playback only routed to one channel	Jörg Krause
2019-04-04	6.0.1	1. Fix Bluetooth name issue 2. Fix non-working multimedia keys when using Bluetooth and the module is not connected to a network 3. Fixed initial Bluetooth volume handling 4. Fix Bluetooth not visable and pairable after Bluetooth disconnect 5. Turn blue LED on, when Bluetooth device connects and module is not connected to a network	Jörg Krause
2019-03-13	6.0.0	1. Add support for new LinTech HBM11 module 2. Add support for Bluetooth A2DP	Jörg Krause
2019-02-18	5.1.3	1. Fix audio dropouts on some devices introduced in version 5.1.2	Jörg Krause
2019-02-06	5.1.2	1. Improved AirPlay latency for iOS 12 and macOS Mojave 2. Fix volume not always muted when set to 0 3. Security and stability updates	Jörg Krause
2018-06-12	5.1.1	1. Fix AirPlay compatibility with Synology Audio Station version 6.4.1	Jörg Krause
2018-05-09	5.1.0	1. Bump Linux kernel version to 4.14 LTS 2. New HTTP-API v19 a. Add station info to network info command b. TIDAL and Qobuz playlist support c. New API endpoint “player” for remote control 3. Improved AirPlay client a. Better synchronization with newer version of macOS and iOS b. Increased stability playing Youtube audio on macOS and iOS c. Revised resynchronization logic improves performance on congested networks d. Added or improved compatibility with AirFoil, TuneBlade, Swinsian, and AirAudio 4. Fix Songcast Sender not playing audio stream after updating from version 5.0.1 5. Fix audio track interruption when switching stations in Songcast Sender mode	Jörg Krause
2018-04-23	5.0.4	1. Fix possible audio pops in AirPlay and Spotify when changing playback mode or switching tracks 2. Fix processing of wifi passwork keys containing special characters	Jörg Krause

		3. Fix 24-bit issue with Audirvana music player	
2018-03-05	5.0.3	1. Allows AirPlay single device selection in multi-room mode 2. Fix AirPlay volume in multi-room mode 3. Fix possible multi-room grouping issues introduced in version 5.0.2 4. Fix possible line-in crash 5. Improved switching between audio protocols and services	Jörg Krause
2018-02-19	5.0.2	1. Fix AirPlay volume curve 2. Fix compatibility with Windows Media Player 3. Fix multi-room compatibility with BubbleUPnP 4. Fix no sound in multi-room mode with Spotify App 5. Fix multi-room grouping issues after power-up	Jörg Krause
2017-12-22	5.0.1	1. Improved Songcast multi-room protocol 2. Improved Spotify Connect multi-room group login 3. Fix updating firmware versions before 4.2.0 4. Reduce the timeout value for the factory reset to four seconds	Jörg Krause
2017-11-30	5.0.0	1. New multi-room protocol Songcast in addition to AirPlay 2. Line-In support 3. Add support for Wi-Fi Protected Setup (WPS) 4. Enhanced Network Support a. Add support for static network configuration b. Add support for hidden network 5. New HTTP-API v18	Jörg Krause
2017-10-17	4.6.3	1. Security fixes (WPA2)	Jörg Krause
2017-09-20	4.6.2	1. Fix issue introduced in 4.6.0 where AirPlay, Spotify, and UPnP might not be available by default after boot in certain circumstances 2. Minor improvements and security fixes	Jörg Krause
2017-07-04	4.6.1	1. Fix issue with AAC audio streams	Jörg Krause
2017-06-13	4.6.0	1. New HTTP-API v17 • Allows to enable/disable AirPlay, Spotify, and UPnP at runtime 2. New Customizations options: • Allows to enable/disable AirPlay, Spotify, and UPnP by default on startup	Jörg Krause
2017-05-12	4.5.0	1. Improved Spotify Connect client 2. Fix of audio streams not being stopped when going into standby mode (bug introduced in version 4.4.0)	Jörg Krause
2017-04-18	4.4.3	1. Fix low audio output level introduced in version 4.4.1	Jög Krause
2017-02-08	4.4.2	1. Fix non-persistent SSID set by “setapmode”	Jörg Krause
2017-01-18	4.4.1	1. Fix wifi after waking up from standby mode 2. Improved network performance 3. Improved playback of 24-bit high resolution audio files	Jörg Krause
2016-12-21	4.4.0	1. Add Spotify Connect combatibility 2. Bump to latest Linux Kernel 4.9	Jörg Krause
2016-11-30	4.3.0	1. Add standby mode for low power consumption 2. Improved switching between AirPlay and UPnP/Radio 3. Improved AUDIO_STATUS pin behavior to be more accurate when an PCM stream is opened and closed 4. Improved wifi network performance 5. New HTTP-API v16 • Advanced AP mode: ○ Setup an encrypted AP ○ Change SSID and channel • Support for multiple network configurations • Set a favourite playlist	Jörg Krause

		• Add actions “playpause”, “next”, “prev” to navigate through a playlist 6. Discard toggling between AP and station mode using pin 1 and enable it for factory reset solely 7. Fix Ethernet connection issue after factory reset or firmware update 8. Fix missing field “name” when calling the radio “query” command after a restart 9. Fix missing download status when issueing a “querystatus” command 10. Fix I2S clock for sample rates above 96 kHz	
2016-07-25	4.2.1	1. Fix not going into AP mode if configured network is not found 2. Fix compatibility with “HTC Connect” 3. Fix AUDIO_STATUS (pin 8) not going to low state after power-up	Jörg Krause
2016-06-27	4.2.0	1. Improved volume control and AirPlay audio playback 2. Auto disable WLAN interface if Ethernet link is detected 3. New radio playback features: • Play and navigate with the multimedia keys through a playlist • Define a favourite station • Enhanced *query* command 4. New sound features: • Get and set the master volume • Get and set the volume for the status tones 5. Improved network toggle button handler 6. Measure network bandwith with iperf3 7. Fix some network issues 8. New HTTP API v1.5	Jörg Krause
2016-05-02	4.1.2	1. Fix duplicated Ethernet MAC addresses 2. Fix LED blinking 3. Fix I2S BLCK	Jörg Krause
2016-04-11	4.1.1	1. Fix streaming issue with some mp3 playlists 2. Fix streaming issue with Windows Media Player	Jörg Krause
2016-03-21	4.1.0	1. Add support for setting the brightness of the LEDs 2. Improve support for MP3 audio files 3. Improve update over the air 4. Fix factory reset sent via HTTP-API 5. Fix storing volume	Jörg Krause
2016-03-02	4.0.1	1. Fix HTTP command “Radio Stop” 2. Fix an AirPlay bug, causing unnecessary resynchronizations 3. Fix an incompatibility bug with some UPnP media control points	Jörg Krause
2016-02-29	4.0.0	1. Add support for remote control of audio playback with keys 2. Improve performance and stability 3. Feature “Serial to WifiBridge” is now optional and not enabled by default	Jörg Krause
2016-02-15	3.3.6	1. Fix storing volume changes followed by a power-cut 2. Fix wrong AirPlay volume after changing UPnP/Radio volume	Jörg Krause
2016-01-27	3.3.5	1. Improve switching between different audio streams by temporarily turning AirPlay off, when UPnP or radio playback is about to begin 2. *HBM10-ETH*: Fix UPnP not available on Ethernet	Jörg Krause
2016-01-15	3.3.4	1. Fix radio playback after network reconnection 2. Fix ”Next Track”-Bug using AirPlay on iOS 9.2 3. Disable DHCP server in network client mode	Jörg Krause
2015-12-23	3.3.3	1. Fix radio unintentionally starting a stream after reboot although radio stream was stopped	Jörg Krause

2015-12-14	3.3.2	1. Fix UPnP stuttering in case a Control Point becomes suddenly unreachable 2. Fix audio noise in quite passages	Jörg Krause
2015-12-01	3.3.1	1. HTTP API v1.3 2. Expose API version in Zeroconf service description 3. Fix UART issue outputting characters twice 4. Fix UPnP not working on Windows 5. Fix AUDIO_STATUS for internet radio 6. Improve network management for wifi + ethernet 7. Fix missing HTTP response for action setconfig, in case of changing the device name only without setting a network configuration	Jörg Krause
2015-10-21	3.3.0	1. Internet radio support 2. HTTPS support 3. New HTTP API v1.2	Jörg Krause
2015-10-19	3.1.3a	1. Fix another power-cut issue	Jörg Krause
2015-09-30	3.1.3	1. Fix bricking issue in case of a power-cut during boot process 2. Fix hostname issue causing trouble with some routers 3. Improve HTTPs JSON parser robustness 4. Backport support for HTTP API v1.1 5. Bump Linux Kernel to 4.1 LTS	Jörg Krause
2015-08-18	3.1.2a	1. Append last four digits of the mac address to wifi ssid	Jörg Krause
2015-08-06	3.2.0	1. HBM10-ETH with Ethernet support 2. Serial to Wi-Fi bridge 3. New HTTP API v1.1	Jörg Krause
2015-07-06	3.1.2	1. Fix clock synchronization with AirPlay 2. Pin *AUDIO_STATUS* also works now for closing UPnP connections 3. Improve compatibility with WHAALE app 4. Change default device and SSID name to “HBM10”	Jörg Krause
2015-05-26	3.1.1	1. Fix non-working *AUDIO_STATUS* in certain cases when streaming with UPnP	Jörg Krause
2015-05-20	3.1.0	1. Add module pin *AUDIO_STATUS* (pin #8)	Jörg Krause
2015-05-19	3.0.0	1. Enable GPIOs 2. Add an icon for UPnP device rendering	Jörg Krause
2015-05-18	2.0.1	1. Fix issue when updating firmware with the iOS app	Jörg Krause
2015-05-04	2.0.0	1. Initial version	Jörg Krause

2 Introduction

The HBM11 is a low-cost and powerful System-on-Module (SoM) specially designed for high-quality audio streaming applications.

The vastly optimized embedded Linux system supports numerous audio streaming protocols such as Spotify Connect, AirPlay, UPnP, DLNA, OpenHome and making it suitable for all types of audio systems including A/V receivers, wired and wireless multi-room speakers and portable audio devices.

Thanks to its highly integrated connectivity, the HBM11 already includes all the necessary interfaces like certified WiFi, Bluetooth/BLE, Ethernet, analog and digital audio in/out, digital display and serial interfaces to minimize the need for external components.

Its complete reference design drastically reduces the development time for own applications on a customer-specific carrier board.

2.1 Target Applications

- Network music stations
- HiFi-systems
- Light and sound systems
- Portable audio system
- Network audio loudspeakers
- Wireless media adapters
- Complete radio and audio products

2.2 Software Features

- NXP Linux Kernel 4.14
- Bootloader (USB Firmware recovery)
- Buildroot rootfs (pre-installed in NAND Flash)
- Audio Streaming Protocols:
    - AirPlay and AirPlay 2
    - Spotify Connect
    - TIDAL Connect
    - UPnP, DLNA, OpenHome
    - Songcast (Multiroom)
    - Bluetooth A2DP
- Manage and listen to playlist from:
    - TuneIn
    - Qobuz
- Firmware Update over the Air
- Remote Control through HTTP API
- Internet radio player
- Audio playback control keys support
- Sample Applications
    - Serial-to-Wifi Bridge
    - Line-In PCM-Recorder
    - Line-In Multiroom-Player

2.3 Board Features

Hardware	HBM11 – 2.4GHz	HBM11 – 2.4+5GHz
CPU	NXP i.MX 6ULL (Arm® Cortex®-A7)	
RAM	256 MB	
Flash	2 Gbit	
Ethernet	10/100 Mbps	
Wi-Fi	IEEE 802.11b/g/n	IEEE 802.11a/b/g/n/ac
Bluetooth	V4.2 (HS) with integrated Class 1.5/Class 2 PA + Bluetooth Low Energy (BLE)	
Interfaces	I2S, I2C, SPI, UART, GPIOs	
Dimension	33.8mm x 49.5mm x 5mm	
Approvals & Certifications	CE, RED, BQB RF Testing Bluetooth Classic and Low Energy	

2.4 Module Block Diagram

The Figure 1 shows the main hardware components assembled on the HBM11 module.

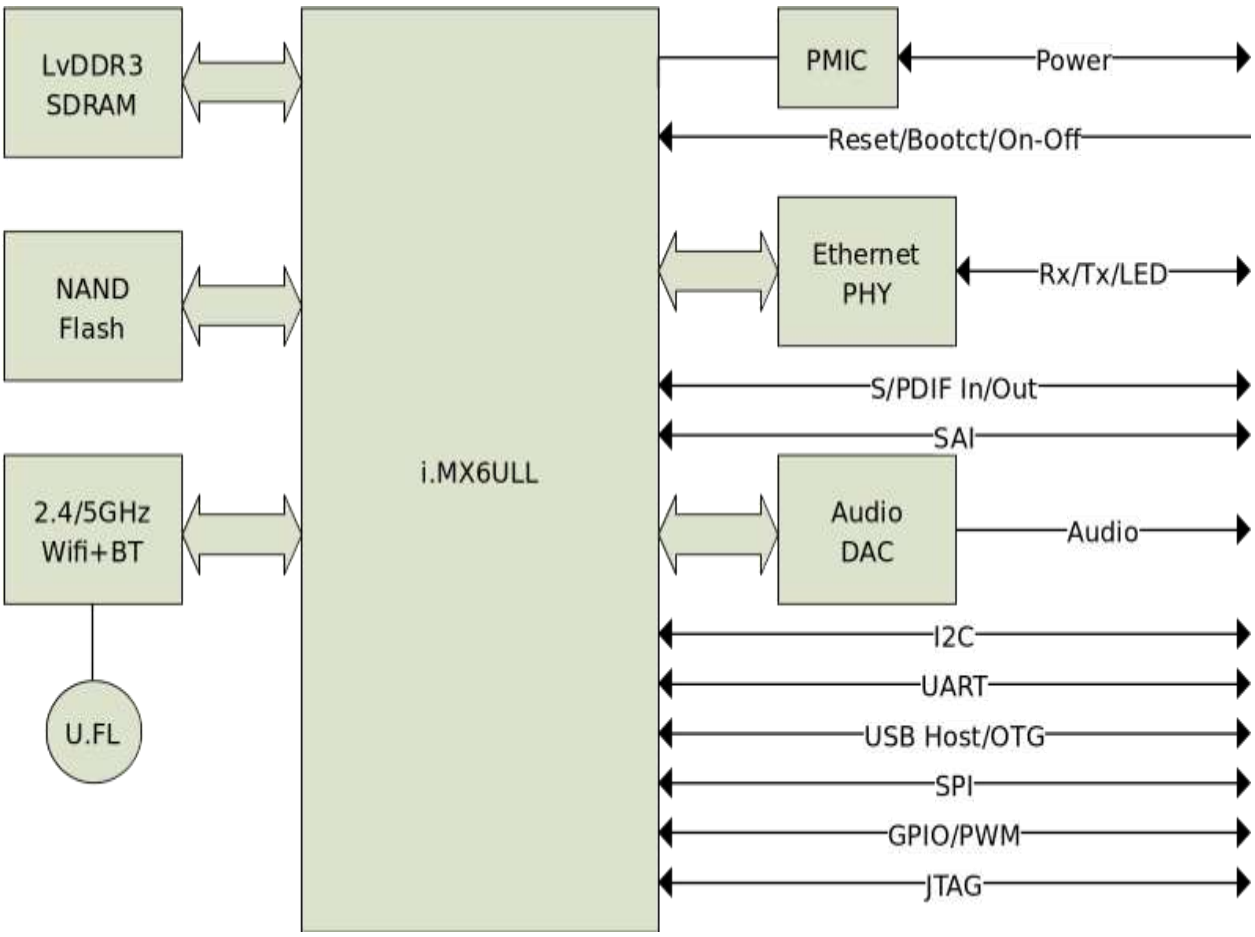


Figure 1 – HBM11 Hardware Components

3 System Specifications

3.1 General

Platform	*OS*	NXP Linux 4.14	
	CPU	NXP i.MX 6ULL (Arm® Cortex®-A7)	
Memory	*NAND FLASH*	2 Gb	
	RAM	256 MB	
Connectivity	*Wi-Fi*	2.4 GHz	2.4 + 5 GHz
	Bluetooth	V4.2 + BLE	
	Antenna	IPEX connector to external antenna (U.FL)	
Audio	*Protocols*	AirPlay, AirPlay 2, UPnP/DLNA, OpenHome, Spotify Connect, TIDAL Connect, HTTP, HTTPS	
	Formats	MP3, AAC, Vorbis, Opus, PCM, WMA, AC3, FLAC, ALAC, APE, WavPack	
	Container	MP4, MKV, OGG, WAV, AIFF, ASF	
	Audio Data Lengths	16 and 24 bit	
	Sampling Frequency	8 to 192 kHz	
	SNR	> 112 dB	
Interfaces	*UART*	1x	
	USB 2.0	1x OTG 1x Host	
	Audio	1x I2S 1x Line out (R, L)	
	I2C	1x	
	Power	*Input*: 5 V *Output*: 3.3 V	
	Reset	1x	
	GND	4x	
	LED	2x	
	KEYS	8x	
	PWM	1x	
	ANT	1x	
	GPIO	Up to 11x	
Environment	*Operation Temperature*	-10 to 70 ℃	
	Operation Humidity	10 to 90 %	
	Storage Temperature	-40 to 100 ℃	
	Storage Humidity	5 to 95 %	

Performance	*Boot Strap*		~15 Sec
	Power Dissipation	*Streaming*	2.2 W
		Idle	1.9 W
	Current Consumption	*Active*	typ: 200 mA @ 5V, max: 800 mA @ 5 V
		Suspend	<25 mA @ 5V
Operating Condition	*VDD*		5 V ± 5%
	VDD_DAC		3.3 V ± 3%

3.2 RF Specification

3.2.1 Wi-Fi 2.4 GHz

Conditions: VDDIO=3.3V, Temp=25°C

Protocols		IEEE 802.11b/g/n
Frequency Range		2.400 GHz ~ 2.497 GHz
Channels		1 - 14
Max data rates	*802.11b*	11 Mbps
	802.11g	54 Mbps
	802.11n	65 Mbps
Modulation	*802.11b*	DQPSK, DBPSK, CCK
	802.11g/n	OFDM/64-QAM, 16-QAM, QPSK, BPSK
Output Power	*802.11b*	16 dBm ± 1.5 dB @ EVM ≤ -9 dB
	802.11g	15 dBm ± 1.5 dB @ EVM ≤ -25 dB
	802.11n	14 dBm ± 1.5 dB @ EVM ≤ -28 dB
Receive Sensitivity	*802.11b*	-87 dBm @ 8 % PER @ 11 Mbps
	802.11g	-74 dBm @ 10 % PER @ 54 Mbps
	802.11n (20 MHz)	-71 dBm @ 10 % PER @ MCS=7
	802.11n (40 MHz)	-69 dBm @ 10 % PER @ MCS=7

3.2.2 Wi-Fi 5 GHz

Conditions: VDDIO=3.3V, Temp=25°C

Protocols		IEEE 802.11a/b/g/n/ac
Frequency Range		4.900 GHz ~ 5.845 GHz
Channels		See European standard EN 301 893
Max data rates	*802.11a*	54 Mbps
	802.11ac (20MHz)	78 Mbps @ MCS=8
	802.11ac (40MHz)	180 Mbps @ MCS=9

Modulation	*802.11a/n*	64-QAM,16-QAM, QPSK, BPSK
	802.11ac	256-QAM, 64-QAM,16-QAM, QPSK, BPSK
Output Power	*802.11a (R=3/4)*	12 dBm ± 2 dB @ EVM ≤ -25 dB
	802.11n (R=5/6)	11 dBm ± 2 dB @ EVM ≤ -28 dB
	802.11ac (R=3/4)	10 dBm ± 2 dB @ EVM ≤ -30 dB
	802.11ac (R=5/6)	10 dBm ± 2 dB @ EVM ≤ -32 dB
Receive Sensitivity	*802.11a (20MHz)*	-73 dBm @ 10 % PER @ 54 Mbps
	802.11n (20MHz)	-71 dBm @ 10 % PER @ MCS=7
	802.11ac (20MHz)	-66 dBm @ 10 % PER @ MCS=8
	802.11ac (40MHz)	-63 dBm @ 10 % PER @ MCS=9
	802.11ac (80MHz)	-59 dBm @ 10 % PER @ MCS=9

3.2.3 Bluetooth

Conditions: VDDIO=3.3V, Temp=25°C

Standard		V4.2
Frequency Band		2402MHz ~ 2480MHz
Number of Channels		79
Modulation		FHSS, GFSK, DPSK, DQPSK
Output Power	*Class 1.5*	8 dBm
	Class 2	2 dBm
Sensitivity	*GFSK (1Mbps)*	-92 dBm @ BER=0.1%
	DQPSK (2Mbps)	-92 dBm @ BER=0.01%
	DPSK (3Mbps)	-85 dBm @ BER=0.01%
Maximum Input Level		-20 dBm

4 Mechanical Specifications

4.1 Module Dimensions

The HBM11 module has the following dimensions, as depicted in Figure 2:

- L x W x H: 35.0 mm x 50.0 mm x 4.0 mm
- Tolerance: ± 0.2 mm

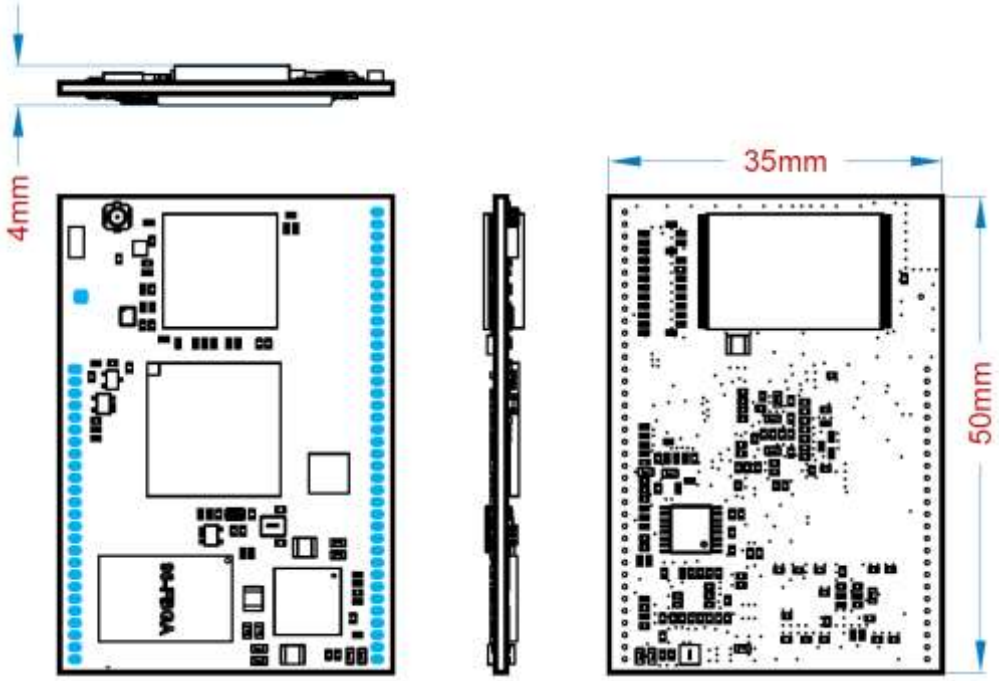


Figure 2 – Board Dimensions

4.2 Pin Layout

Figure 3 shows the recommended pin layout for the two 25-pins pitches of the HBM11 module.

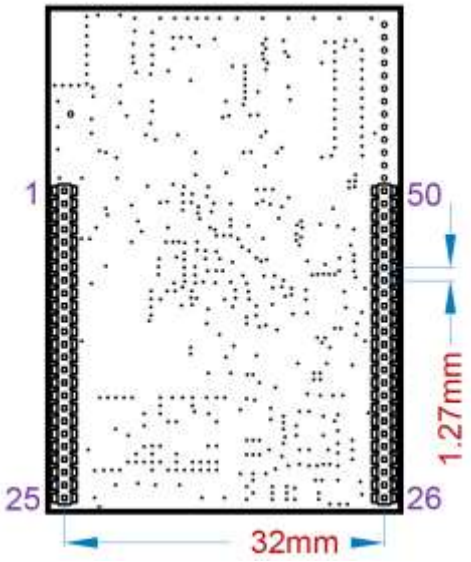


Figure 3 – Recommended Pin Layout

4.3 Pin Header

For connecting the HBM11 module on a host board two 1.27 mm pitch 25-pins headers are required. The specification for the pin header is as depicted in Figure 4.

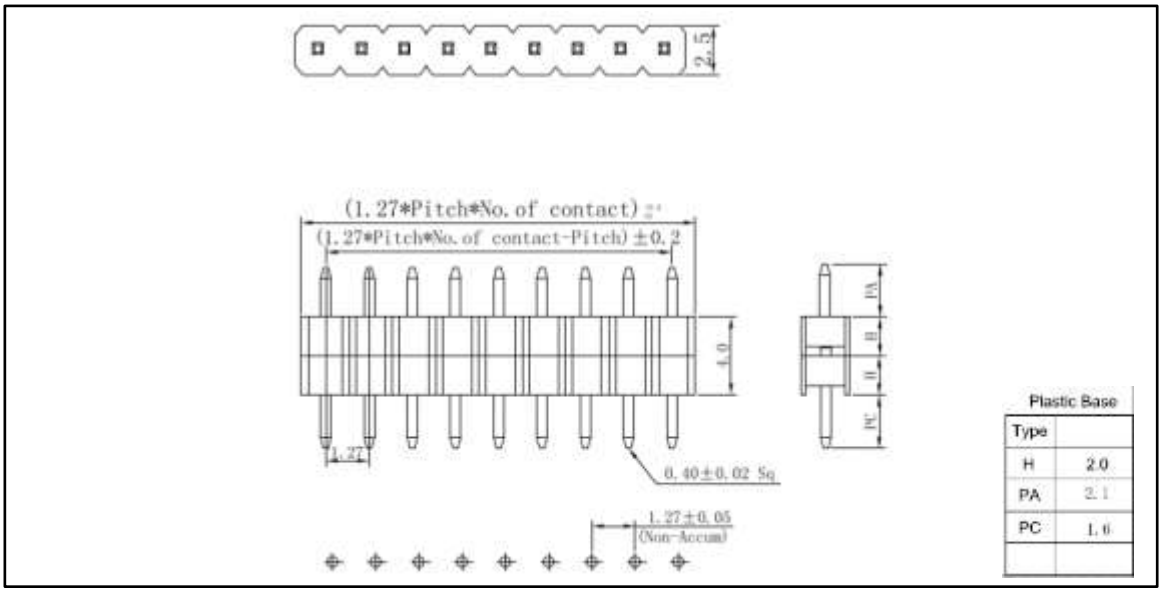


Figure 4 – Pin Header

4.4 Antenna

The HBM11 needs an external antenna attached to the U.FL connector. Please make sure that the radio can achieve its best RF performance by following:

1. Due to the surface mount antenna on the module, the green area on all layers of the customer circuit board should be free of any metal objects. Specifically, there should be no ground plane, traces or metal shield case.
2. The wireless signal including Wi-Fi applications is mostly affected by the surrounding environment, such as trees, and other obstacles. Metal absorbs a certain radio signal. In practical application, the data transmission distance is affected.
3. Do not use metal housings.

5 Pin Descriptions

5.1 Overview

Figure 5 shows an overview of the available pin groups exposed by the 50-pin header.

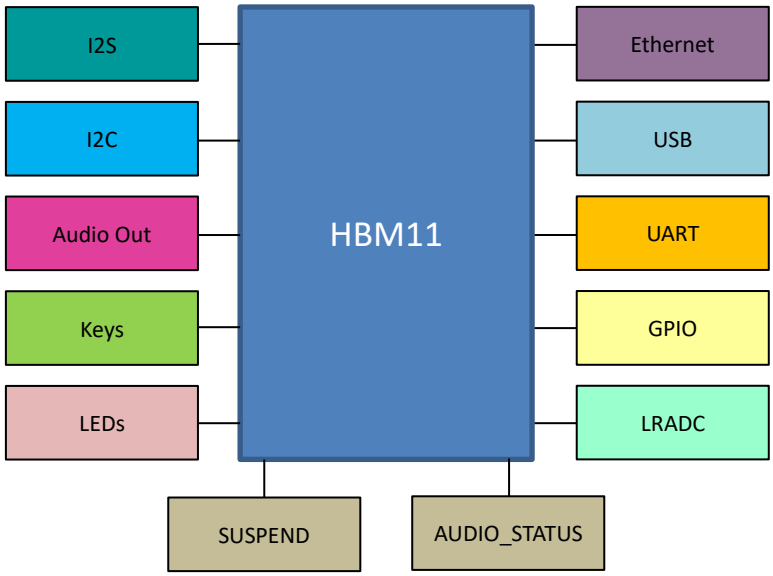


Figure 5 – Pin Group Overview

The “External Audio Codec”, “Power Manager” and “Amplifier” are examples of possibly connectected external components.

5.2 Pin Definitions

Pin	Name	I/O	Function	Notes
1	KEY_WPS_FACTORY_RESET	I	WPS/Factory reset	7
2	HW RESET	I	HW_RESET	
3	FEC_LED	O	ETH	4
4	GND	#	Digital Ground	
5	GPIO12	I/O	GPIO	2
6	GPIO13	I/O	GPIO	
7	SUSPEND	I	Suspend to RAM	7
8	AUDIO_STATUS	O	Audio status	6
9	KEY_VOLUME_UP	I	Key – Volume up	5
10	GPIO1	I/O	GPIO	2
	USB1_ID	I	USB Host ID	
11	USB1_PWR	I/O	USB Host Enable	3
12	USB1_OC	I/O	USB Host Overcurrent	3
13	GPIO4	I/O	GPIO	2
	PWM5_OUT	O	PWM	
14	GND	#	Digital Ground	
15	GPIO5	I/O	GPIO	1, 2

	SPDIF_RX	I	S/PDIF	
16	GPIO6	I/O	GPIO	1, 2
	UART1_TXD	O	UART – Tx	1
17	GPIO7	I/O	GPIO	1, 2
	UART1_RXD	I	UART – Rx	1
18	GPIO8	I/O	GPIO	1, 2
	UART1_CTS	O	UART – CTS	1
19	GPIO9	I/O	GPIO	1, 2
	UART1_RTS	I	UART – RTS	1
20	DCDC_3V3	O	I/O voltage for GPIO	
21	LED1	O	Status LED 1	
22	LED2	O	Status LED 2	
23	GPIO10	I/O	GPIO	1, 2, 9
	I2C2_SDA	I/O	I2C – SDA	1, 3, 9
24	GPIO11	I/O	GPIO	1, 2, 9
	I2C2_SCL	I/O	I2C – SCL	1, 3, 9
25	GND	#	Digital Ground	
26	SYS_4V4	#	nc	
27	VSYS	I	5V Power In	
28	AUDIO_G	#	Analog Ground	
29	AUDIO_R	O	Line out Right	
30	AUDIO_L	O	Line out Left	
31	KEY_VOLUME_DOWN	I	Key – Volume down	5
	SPDIF_TX	O	S/PDIF	
32	SAI1_SYNC	O	I2S – LRCLK	
33	SAI1_MCLK	O	I2S – MCLK	
34	SAI1_BCLK	O	I2S – BITCLK	
35	SAI1_TXD	O	I2S – DOUT	
36	SAI1_RXD	I	I2S – DIN	
37	FEC_3V3	#	ETH – 3V3 supply	
38	ETH1_RXP	I	ETH – RX+	
39	ETH1_RXN	I	ETH – RX-	
40	ETH1_TXP	O	ETH – TX+	
41	ETH1_TXN	O	ETH – TX-	
42	USB0_D_N	I/O	USB OTG – D-	
43	USB0_D_P	I/O	USB OTG – D+	
44	USB1_D_N	I/O	USB Host – D-	3
45	USB1_D_P	I/O	USB Host – D+	3
46	GND	#	Digital Ground	
47	KEY_PLAY_PAUSE	I	Key – Play/Pause	5
48	KEY_STOP	I	Key – Stop	5
49	KEY_NEXT	I	Key – Next	5
50	KEY_PREV	I	Key – Previous	5
51	LICELL	#	nc	8
52	VSYS	#	nc	8
53	USB_OTG1_VBUS	#	nc	8

54	USB_OTG2_VBUS	#	nc	8
55	ONOFF	I		8
56	JTAG_NTRST	I/O	JTAG/I2S2/GPIO	8
57	JTAG_TDI	I/O	JTAG/I2S2/GPIO	8
58	JTAG_TMS	I/O	JTAG/I2S2/GPIO	8
59	JTAG_TCK	I/O	JTAG/I2S2/GPIO	8
60	JTAG_TDO	I/O	JTAG/I2S2/GPIO	8
61	GPIO	I/O	GPIO	8
	PWM3	I/O	PWM	8
62	TFT_VLEDM	#	nc	8
63	TFT_VLEDP	#	nc	8

Notes:

1. Pins 15 – 19 and pins 23 – 24 are multiplexed.
2. All GPIOs can be fully customized, see ch 10.4 Customization GPIOs.
3. Optional, not enabled by default.
4. Reserved.
5. If used, a 10 kΩ pull-up resistor is necessary. If not used, a 10 kΩ pull-down resistor is recommended instead.
6. If used, a 10 kΩ pull-down resistor is necessary.
7. If used, a 10 kΩ pull-up resistor is necessary.
8. Optional, not used by default
9. Internal Pullup 1,5kOhm

5.3 Pinout

Figure 6 shows the complete pinout of the HBM11 module.

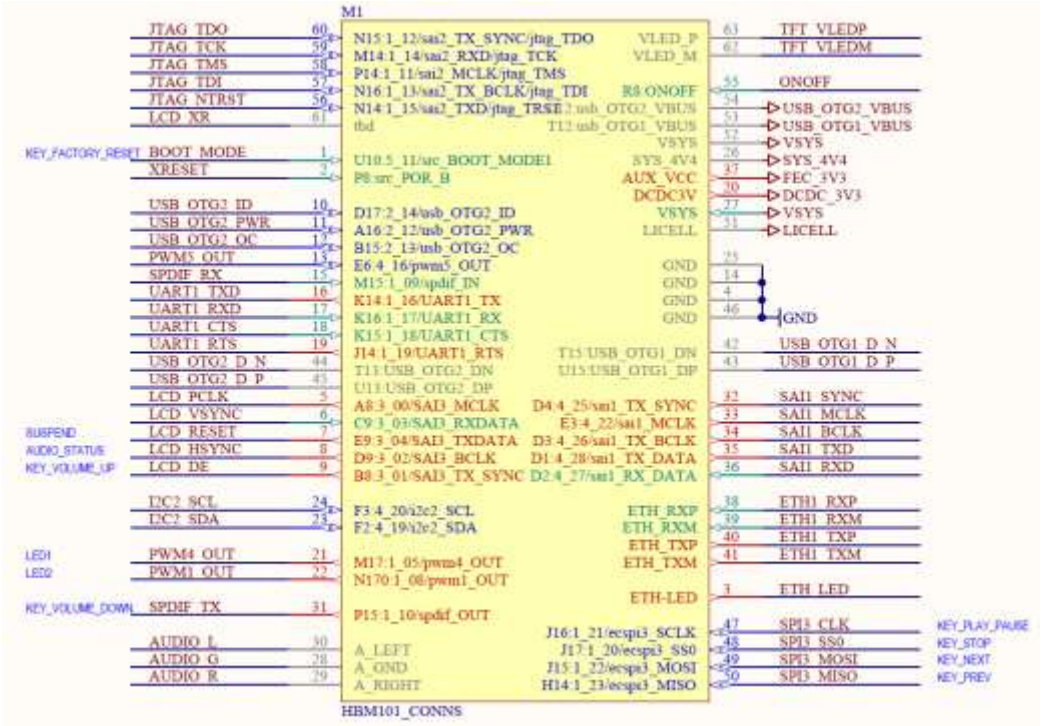


Figure 6 – HBM11 Pinout

6 Interfaces

6.1 I2S

The HBM11 module transmits the PCM audio data via the I2S interface, whereby the HBM11 is the controller of the Master Clock (MCLK). Figure 7 shows the I2S connection between the HBM11 module and any external audio codec.

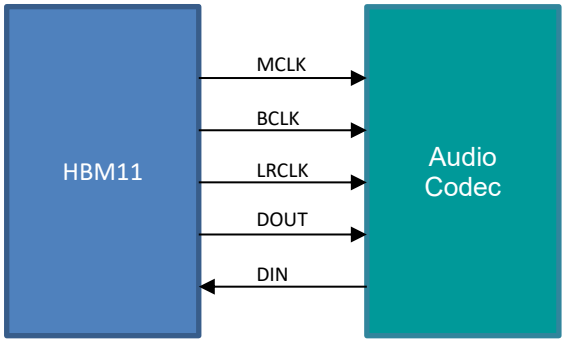


Figure 7 – IS2 connection

Audio data for each stereo channel is clocked with the BCLK signal. Data is time multiplexed with the LRCLK, indication whether the left or right channel data is present. The LRCLK is also used as a timing reference to indicate the beginning or end of the data words. The minimum number of BCLKs per LRCLK period is two times the number of 24 bits.

The MSB of the output data changes on the first falling edge of BCLK following an LRCLK transition, and may be sampled on the next rising edge of BCLK. LRCLK is low during the left samples and high during the right samples.

Figure 8 shows the I2S audio format used to transmit the PCM audio signal.

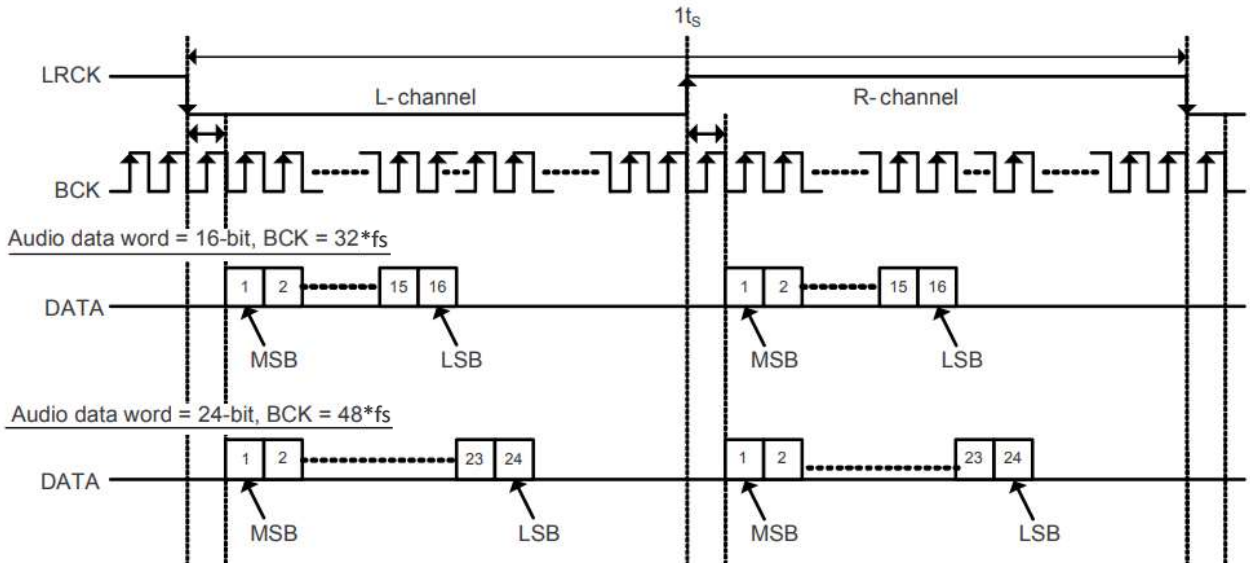


Figure 8 – I2S Audio Format

The I2S interface supports a MCLK to LRCLK ratio of 192*fs and 384*fs and sampling rates of 8kHz to 192KHz. The BCLK base rate is 32*fs for 16-bit audio data word length and 48*fs for 24-bit audio data word length.

Sampling Rate (kHz) LRCLK	MCLK (MHz)		BCLK (MHz)	
	192*fs	**384*fs**	**32*fs**	**48*fs**
8	–	3.072	0.256	0.384
32	–	12.288	1.024	1.536
44.1	–	16.9344	1.4112	2.1168
48	–	18.432	1.536	2.304
88.2	–	33.8688	2.8224	4.2336
96	–	36.864	3.072	4.608
176.4	33.8688	–	5.6448	8.4672
192	36.864	–	6.144	9.216

MCLK Frequencies and Audio Sample Rates

6.2 Line-Out

The HBM11 module is equipped with a Wolfson WM8524 Audio Stereo DAC. The analog output of the internal DAC is routed to the HBM11 pin interface as pin #30 (LINE_OUT_L) and pin #29 (LINE_OUT_R). Theses two pins can directly be connected to a line-out jack as shown in Figure 9.

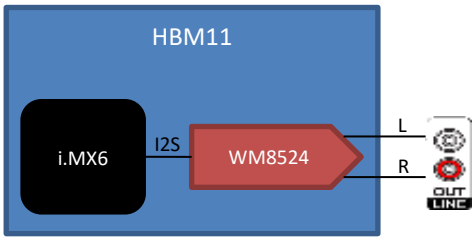


Figure 9 – Analog Audio Output of the internal DAC

6.3 Line-In

For analog input an external I2S ADC is required. The external ADC needs to fulfill the requirements for the I2S interface as described in chapter 6.1. The HBM11 module has builtin support for the Cirrus CS5343 ADC.

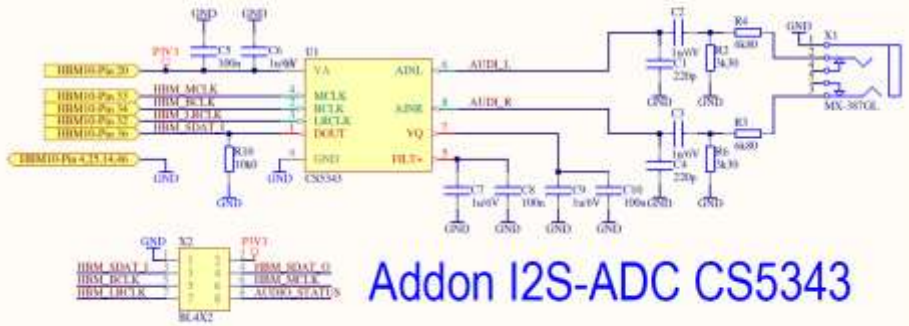


Figure 10 – Analog Line-In

6.4 S/PDIF

Digital input is accessible via the S/PDIF RX pin #15, digital output is accessible via the S/PDIF TX pin #31. For using the S/PDIF interface the only requirement is to add an optical TOSLINK receiver jack on the circuit board, as shown in Figure 11.

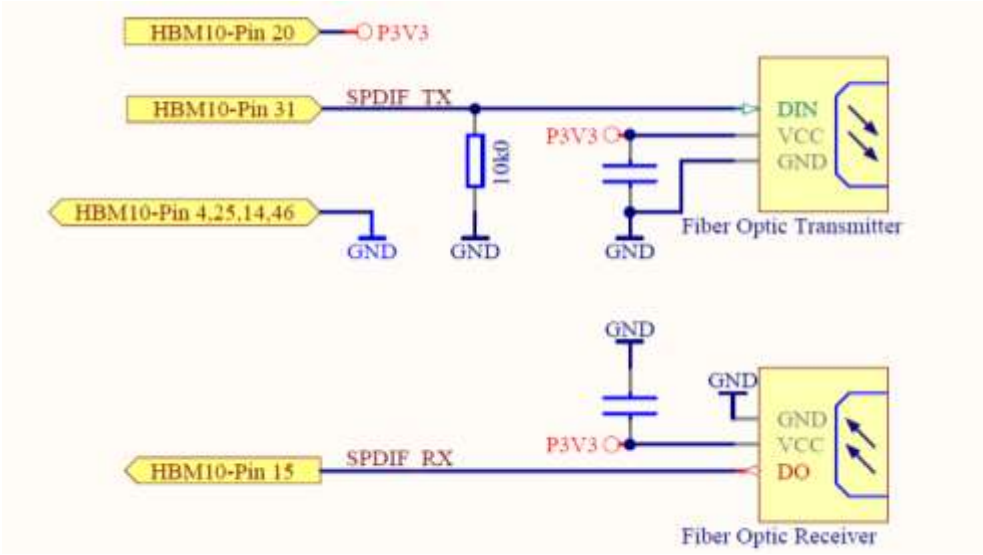


Figure 11 – SPDIF I/O schematic

6.5 Ethernet

The HBM11 module is equipped with a 10BASE-T/100BASE-TX IEEE 802.3 Compliant Ethernet Transceiver chip. When using the Ethernet functionality a layout as shown Figure 12 is recommended.

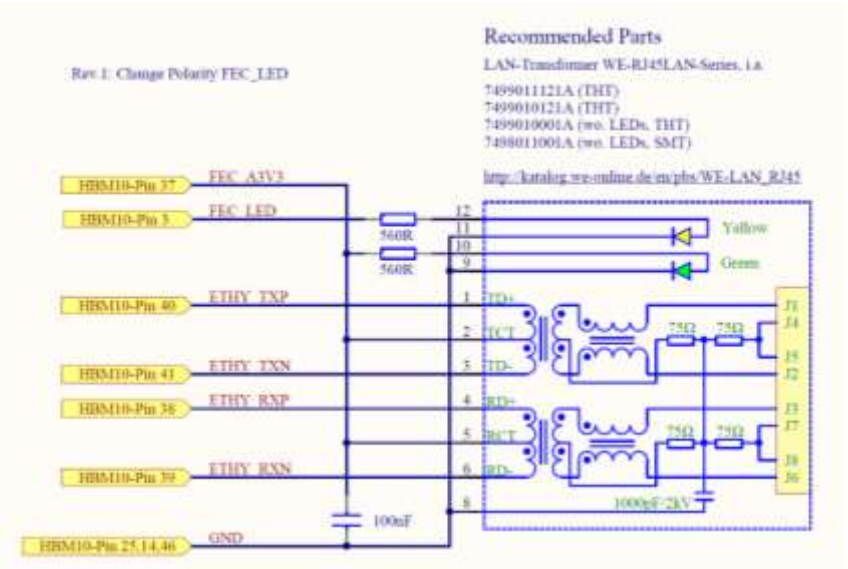


Figure 12 – Recommanded usage of the Ethernet interface

6.6 USB

The i.MX6 processor used on the HBM11 includes two Universal Serial Bus (USB) version 2.0 controllers:

- USB0 is OTG-capable
- USB1 is host-only

USB0 is capable of supplying VBUS, while USB1 is not.

The USB controller can operate as either a USB device or a USB host. Note that a dual-device configuration is not supported. As a host controller, it can enumerate and control USB devices attached to it. The USB controller supports eight endpoints: one control, one bulk-out, on bulk-in, and five flexible endpoints. Using the OTH features, the USB controller can negotiate with another OTG system to be either the host or the device in a peer connection.

The USB controllers operate either in USB 2.0 high-speed mode at 480 Mbps or full-speed mode mode at 12 Mbps.

The i.MX6ULL processor already integrates USB 2.0 PHYs providing a USB Transceiver Macrocell Interface (UTMI) allowing the USB_DP and USB_DN pins to be connected directly to a USB connector as shown in Figure 13.

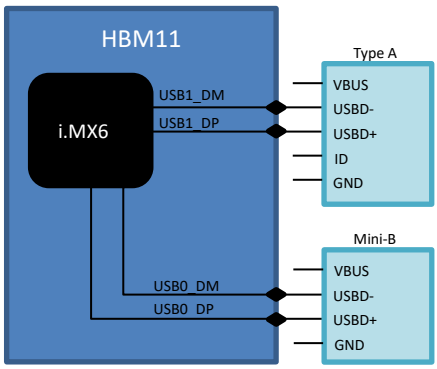


Figure 13 – USB connection

6.7 LEDs

Two status LEDs “LED1” and “LED2” are used to indicate the current device status:

Status	LED	
	LED1	**LED2**
Bootloader initializes hardware	On	On
Bootloader starts Linux	On	Off
Booting Linux	Regular Double Flash	Off
Device Mode		
AP mode	Off	On
Station mode	On	Off
Connecting to network	Regular Single Flash	Off

6.8 Keys

The HBM11 module provides eight user key pins:

Pin	Name	Type	Description
1	KEY_WPS_FACTORY_RESET	Reset	The functionality of the key depends on the press and hold duration: • Short press event (< 4 seconds): WPS • Long press event (> 4 seconds): Factory reset
2	HW_RESET	Reset	POR reset that leads to a reboot.
9	KEY_VOLUME_UP	Audio	Volume up
31	KEY_VOLUME_DOWN	Audio	Volume down
47	KEY_PLAY_PAUSE	Audio	Play/Pause
48	KEY_STOP	Audio	Stop
49	KEY_NEXT	Audio	Next
50	KEY_PREV	Audio	Previous

Pushbuttons are recommended for using the pins.

When using the reset pins, a pull-up resistor of 10 kΩ is required for pin KEY_WPS_FACTORY_RESET. However, no such resistor is required for pin HW_RESET, as show in Figure 14.

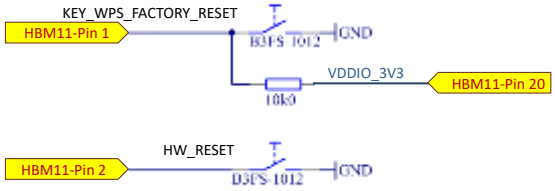


Figure 14 – Recommanded usage of the Reset Pins

When using the audio keys, a pull-up resistor of 10 kΩ is required, as shown in Figure 15.

Figure 15 – Recommanded usage of the Audio Pins

6.9 GPIOs

The HBM11 module exposes up to 11 GPIOs, whereby five GPIOs are always available and six GPIOs are muxed with the UART and I2C function.

Pin	Name	Mux
5	GPIO12	
6	GPIO13	
10	GPIO1	
13	GPIO4	
15	GPIO5	
16	GPIO6	UART
17	GPIO7	
18	GPIO8	
19	GPIO9	
23	GPIO10	I2C
24	GPIO11	

The GPIOs can be used in a custom design to add additional functionality. For support please contact the LinTech team.

6.10 Suspend Pin

Note: It is recommended to connect a 10 kΩ pull-up resistor to this pin.

The SUSPEND pin (HBM11 pin #7) can be used to put the module into a “Standby” mode to reduce power consumption. In this mode, the CPU is placed in the Wait-For-Interrupt (WFI) state, and the DRAM is placed in Self-Refresh mode.

The suspend pin can be connectected to some external power management controller to control the power consumption of the HBM11 as shown in

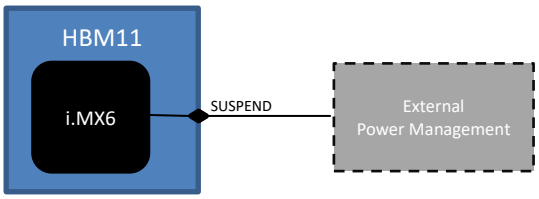


Figure 16 – Connecting an external power management controller to the SUSPEND pin

Figure 17 shows the usage of the suspend pin.

Figure 17 – Usage of the SUSPEND pin

To suspend the module a transition from high to low on pin #7 is necessary. Thereafter, the module will stop any streaming applications and disconnect itself from the network before shutting down the Wi-Fi and the internal DAC chip.

To wakeup the module a transition from low to high on pin #7 is necessary. The module will thereafter turn on the Wi-Fi and internal DAC chip and starts the network connection procedure. If no known network is found the module will switch to AP mode. After the network setup is done all streaming applications will be started again.

6.11 Audio Status Pin

Note: It is recommended to connect a 10 kΩ pull-down resistor to this pin.

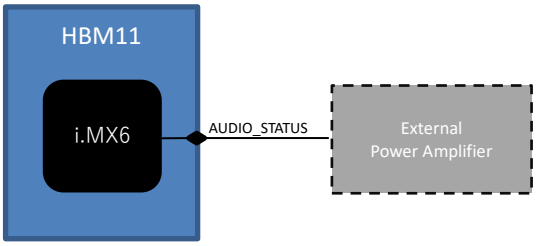


Figure 18 – Connecting an external power amplifier to the AUDIO_STATUS pin

The AUDIO_STATUS pin (HBM11 pin #8) represents the audio output status of the module:

Level	Function
Low	No audio output
High	Audio output PCM device is active

The level toggles from low to high state when a streaming application access the audio output PCM device, e.g. the "Play" button is pressed on a remote device. The level toggles from high to low when a streaming application releases the audio output PCM device, e.g. audio streaming is stopped by pressing the "Stop" button.

Figure 19 – Usage of the AUDIO_STATUS pin

Note, that between switching tracks the state might went from high to low for a short period of time. Furthermore, it might take time some time after a track is finished until the state goes to low. This behavior is very dependent of the streaming application and might vary between UPnP, AirPlay, Spotify Connect, and TIDAL Connect.

6.12 I2C

The I2C interface is a standard two-wire serial interface used to connect the HBM11 module with external peripherals. This interface provides a standard speed (up to 100 kbps), and a fast speed (up to 400 kpbs) I2C connection to multiple devices with the HBM11 acting either in master or slave mode. Typical applications for the I2C bus include:

- Audio Codec control
- LCD display control
- EEPROM

6.13 UART

The HBM11 module exposes one UART interface for using the application UART, featuring:

- serial-to-parallel conversion on data received from a peripheral device.
- parallel-to-serial conversion on data transmitted to the peripheral device.

The application UART offers similar functionality to the industry-standard 16C550 UART device, supporting flow control (RTS/CTS) and is high-speed capable of running up to 3.25 Mbps with 16-byte receive and transmit FIFOs.

7 Network

The HBM11 module supports two of the most important wireless standards – WLAN and Bluetooth. Also on bord is Ethernet, as the most widely local area network technology.

7.1 Ethernet

Setting up the Ethernet interface is as easy as plugging in a Ethernet cable into the jack connector. As soon as a link pulse is detected the HBM11 module will request an IP address automatically using DHCP. No additional setup is required at all to add the HBM11 module to an existing network.

7.2 Bluetooth

The HBM11 module is using the Bluetooth classical protocol Advanced Audio Distribution Profile (A2DP) for wireless audio streaming. A2DP is designed to unidirectionally transfer an audio stream in up to two-channel stereo from one device to another. The HBM11 module implements an A2DP sink.

Bluetooth operation can be controlled using the “bluetooth” HTTP API endpoint.

Bluetooth operation is enabled by default. After booting the operating system, the HBM11 module will by visible and pairable for 180 seconds. After 180 seconds, visibility and pairability will be turned off. Any device, which wants to connect to the HBM11 module using Bluetooth needs to enable the visibility and pairability using the “setproperties” API command.

After pairing with a A2DP source the Bluetooth device will disable the visibility and pairability, so only one Bluetooth device can connect at the same time. If another Bluetooth device wants to connect to the HBM11 module, the currently connected Bluetooth devices needs to disconnect from the HBM11 module first.

7.3 WLAN

7.3.1 Operation

The module can operate in access point (AP) mode or station mode. By default, if not configured, the module will boot into AP mode, except in case of the HBM11-ETH with Ethernet cable plugged-in - in that case it will boot into station mode. Without Ethernet, the module can be easily integrated into an existing network with a few HTTP requests using the HTTP API.

Furthermore, the device will boot into station mode, if any of the configured wireless networks is found when scanning the environment and the corresponding access point accepted the authentification.

After successful authentification the internal DHCP client will request a dynamic IP, so it is necessary that a DHCP server is running on the network. When requesting the DHCP client will use a distinct hostname for identifying itself on the network. The hostname is based on the model name and the last four digits of the MAC address, e.g. HBM11-9710.

The network configuration state is indicated by the modules status LEDs. When starting the network configuration process the *Status LED 2* (pin #22) turns off and the *Status LED 1* (pin #21) begins to flash periodically. If the authentification is successful and the module receives a dynamic IP from the DHCP server, the network configuration process is assumed as successful and the *Status LED 1* will turn on permanently.

If authentification with the access point or the DHCP request fails, the network configuration process is assumed as unsuccessful and the module will turn back into AP mode. In this case the *Status LED 1* will stop flashing and the *Status LED 2* will turn on permanently.

Some of the possible reasons for an unsuccessful network connection are:

- a wrong network key
- the DHCP client does not receive an IP address within 60 seconds
- the connection to the networks access point is somehow lost

In case of a successful connection to an access point, the module is integrated into the network and can be accessed by the IP address provided by the networks DHCP. The current IP address can be easily obtained by resolving the hostname using *Zeroconf*.

Note, that if the module is powered-up, but no known network is found, the modules switches into AP mode.

Note, that if the module is powered-up and connects successfully to a network and loses the connection to the access point later, the *Status LED 1* (pin #21) turns off and the module waits for the access point to become visable again. Beware, that is might need up to 180 seconds until the module connects to the access point again.

The Figure 20 shows the basic flow when the modules starts the network process.

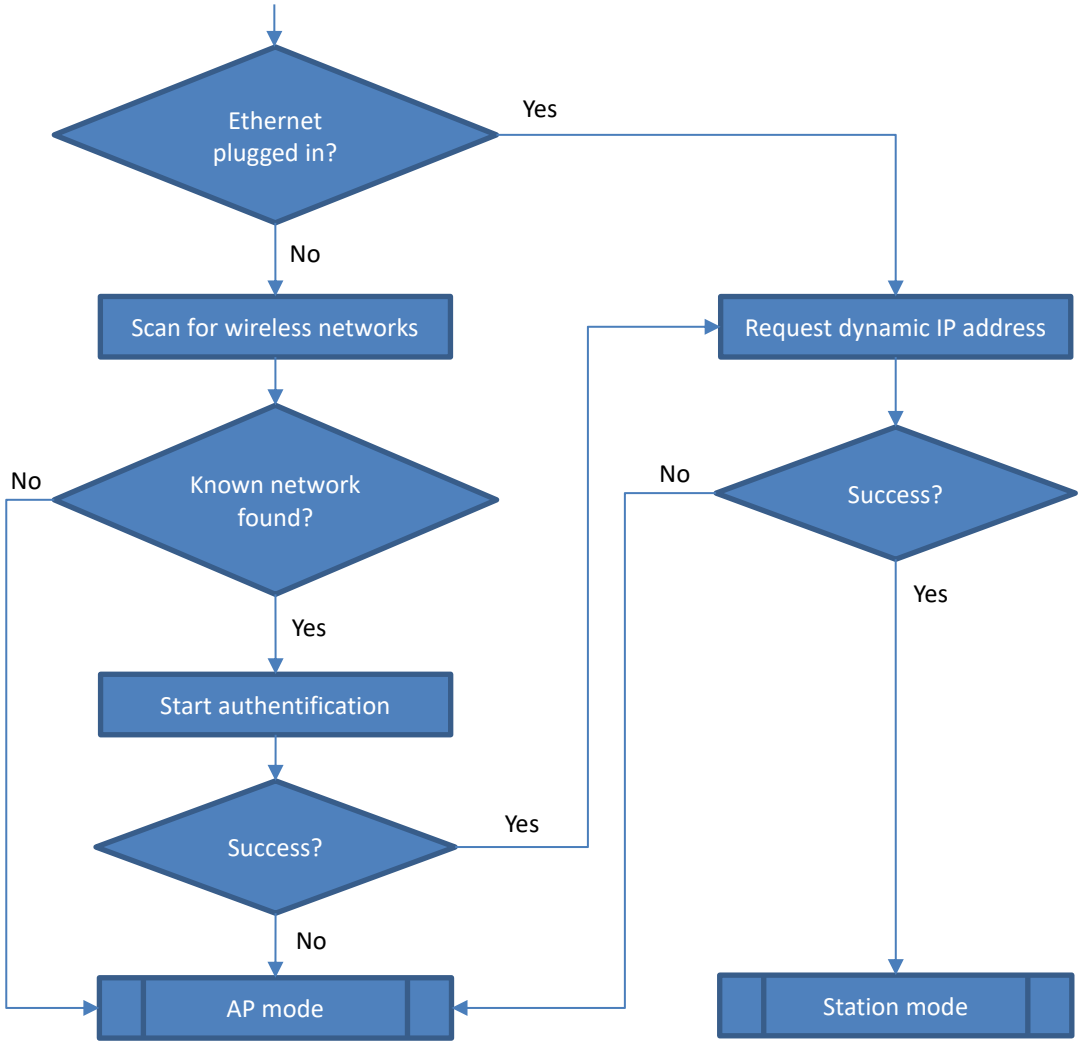


Figure 20 – Network startup procedure

7.3.2 Configuration

Setting up the wireless interface only needs a few HTTP requests. All necessary commands are described in the HTTP API. The basic workflow is as following:

Figure 21 – Steps to setup a wireless network

The “scan” action returns a list of access points identified by their BSSID[1], including additional network configuration parameters, like authentification and encryption parameters. To connect to one of the access points from this list the corresponding network configuration is added to a list of known networks by the “add” action. To finally start the connection procedure the “select” action is used.

If the connection to the access point is established successfully the HBM11 module will start requesting an IP address automatically using DHCP.

7.3.3 Static IP

It is possible to assign the HBM11 module one ore more static IP address configurations by providing the following information:

- Static IP address
- Subnet mask
- Gateway IP address

7.3.4 Roaming

The HBM11 module has network roaming support to make travels between different networks as smoothly as possible. Therefore every network added by the basic setup is put into a list of known networks. Besides the already mentioned “scan”, “add” and “select” actions, the “list” and “remove” actions complementary the actions to easily manage the list of known networks.

The HTTP API allows an enhanced configuration of multiple networks with different settings. Figure 22 shows an example of three known networks “Home”, “Uni” and “Work”. Both the “Home” and the “Work” networks have been assigned a separate static IP from the static IP list whereas the “Uni” networks uses the DHCP client to obtain an dynamically assigned IP address.

[1] Basic Service Set Identification

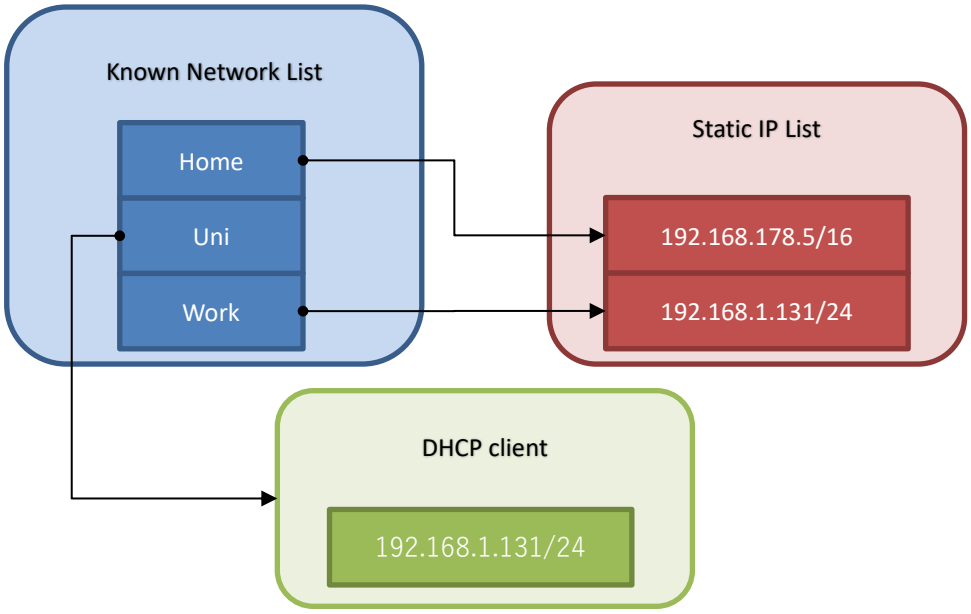


Figure 22 – Example of an enhanced network configuration with multiple known network

If the current network is removed, the module will perform a new scan and will search for any other enabled networks from the known network list. If one or more known network are found by scanning, the module will associate with the access point having the best signal strength automatically or switching to AP mode otherwise. If all networks are removed from the list, the module will switch to AP mode likewise.

Note, that the “id” property returned by the “list” action might change after issueing a “remove” action. That’s why, it is recommend to run the “list” command after emitting a “remove” action to update the network list.

7.4 Zeroconf

The module announces its HTTP service using DNS-SD[2]. The service type‘s name is _dockset._tcp. The instance name is the same as the device name. Note, if not set by the user, the device name defaults to concatenating the model name and the last for the digits of the MAC address with a hyphen, e.g. HBM11-BDDF.

The modules response to an query with a SRV, TXT and address (type A) record. While the SRV record includes the port number of the HTTP server, the address records contains the IPv4 address of the module. The TXT records contains additional information:

TXT record	Description	Example
api	The HTTP-API version.	v22
model	The model type.	HBM11
swver	The firmware version.	6.3.0

Using the IPv4 address, the port number and the api version string obtained from the mDNS response, the module can easily be configured using the HTTP API.

[2] Domain name service – service discovery

7.5 WPS

The HBM11 module supports Wi-Fi Protected Setup using the push button method. The push button method requires the user to press the WPS button on both the Access Point and the HBM11 module within a two-minute interval.

After pushing the WPS button, the HBM11 module disables any previous network configuration and starts the discovery mode indicated by flashing the blue LED fast with an on/off-cycle of 250 ms.

In case of a successful connection, the blue LED will switch to flash with normal on/off-cycle of 500 ms indicating the IP configuration phase. When the HBM11 module has been assigned an IP address, it stops flashing the blue LED and is turning it on permanently.

In case of a WPS timeout or error, the module will re-enable its previous network configurations. If the module was operating in AP mode before the WPS procedure, it will become AP again. If it was already operating in station mode being connected to another network, it will reconnect to that network again.

7.6 Tools

One of the most common network problems is insufficient or unreliable bandwidth. Bandwith limitation can cause packet loss, delays, and jitters. In addition, if the required sending and receiving bit rates exceed the bandwith limitations of the network, network congestion will occur and eventually results in a poor audio experience.

7.6.1 iPerf3

iperf is a commonly used network testing tool to measure the bandwidth of a network. The HBM11 module runs an iperf3 server on port 5201. Note that the iperf3 server has to be enabled using the HTTP API.

Running an iperf3 client on another device in the same network allows to measure the bandwidth between the two endpoints.

This is a snippet of a measurement when the HBM11 is connected via the Ethernet interface:

```
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  11.2 MBytes  93.8 Mbits/sec    0    122 KBytes
[  5]   1.00-2.00   sec  11.2 MBytes  94.4 Mbits/sec    0    123 KBytes
…
[  5]   8.00-9.00   sec  11.2 MBytes  93.8 Mbits/sec    0    124 KBytes
[  5]   9.00-10.00  sec  11.7 MBytes  98.0 Mbits/sec    0    185 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   113 MBytes  94.5 Mbits/sec    0             sender
[  5]   0.00-10.00  sec   112 MBytes  94.2 Mbits/sec                  receiver
```

The example below shows a snippet of the results when the HBM11 is connected wirelessly connected to a IEEE802.11n network:

```
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  2.67 MBytes  22.4 Mbits/sec    0    375 KBytes
[  5]   1.00-2.00   sec  2.61 MBytes  21.9 Mbits/sec    0    485 KBytes
…
[  5]   8.00-9.00   sec  1.99 MBytes  16.7 Mbits/sec    0    488 KBytes
[  5]   9.00-10.00  sec  2.11 MBytes  17.7 Mbits/sec    0    488 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  22.3 MBytes  18.7 Mbits/sec    0             sender
[  5]   0.00-10.00  sec  22.1 MBytes  18.5 Mbits/sec                  receiver
```

The example below shows a snippet of the results when the HBM11 is connected wirelessly connected to a IEEE802.11a network:

```
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  7.02 MBytes  58.9 Mbits/sec    0    494 KBytes
[  5]   1.00-2.00   sec  7.02 MBytes  58.9 Mbits/sec    1    373 KBytes
…
[  5]   8.00-9.00   sec  6.15 MBytes  51.6 Mbits/sec    0    469 KBytes
[  5]   9.00-10.00  sec  7.15 MBytes  60.0 Mbits/sec    1    352 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  70.0 MBytes  58.7 Mbits/sec    2             sender
[  5]   0.00-10.00  sec  70.7 MBytes  59.3 Mbits/sec                  receiver
```

8 Audio

8.1 UPnP

UPnP is a family of network protocols, designed for easily connecting devices in home networks. The name UPnP ("Universal Plug and Play") was chosen because little or no configuration is needed for the different elements to discover each other and play together.

The UPnP audio/video (UPnP AV) section of the protocol implements a way for data stored on a media server (PC, NAS, etc.) to be played on a media streamer, e.g. the HBM11 audio module.

Generelly, there are three building blocks in a network audio solution as shown in figure :

- A media server stores the audio data and makes it available to media renderers. It also implements a directory/tags manager which extracts audio tags from the media files and builds a database used for searching or browsing the collection.
- Media renderers receives the audio data from the media server and actually plays the audio file by passing the audio data to the DAC (Digital Analog Converter).
- The remote control point is mostly a user interface application running on a PC, tablet or smartphone. This application is used to select the media server and renderer as well as choosing songs from a database and controlling the audio playback.

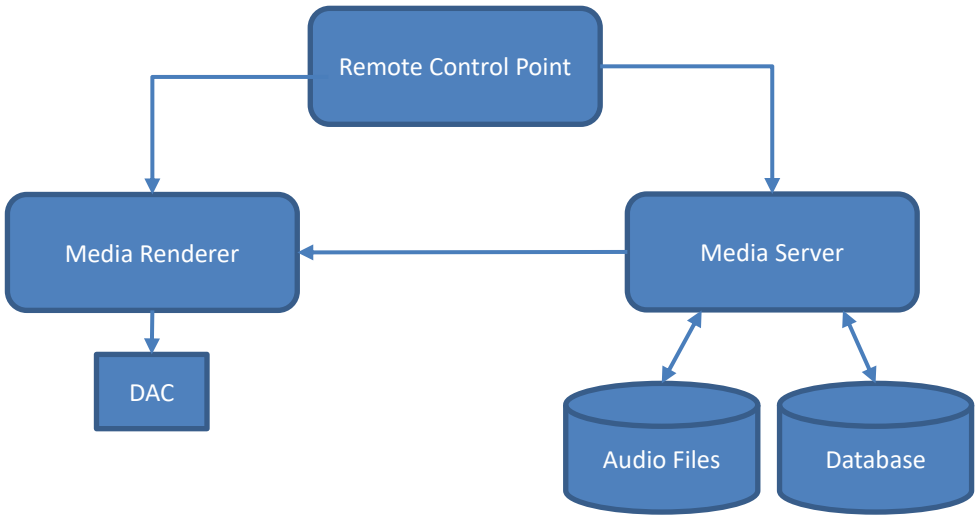


Figure 23 – UPnP devices

8.1.1 Media Renderer

The HBM11 module implements a UPnP media renderer according to the "MediaRenderer:1" specification[3]. Any supported audio file can be played back by selecting the HBM11 as a Media Renderer in the Remote Control Point. The suffix "-UPnP/AV" is appended to the name of the UPnP media renderer, e.g. "HBM11-7264-UPnP/AV".

8.1.2 Media Server

The HBM11 automatically starts a UPnP Media Server when a USB pendrive or external USB disk is plugged into the USB host port. The external pendrive or disk needs to be formatted in FAT16, FAT32 or NTFS format.

[3] http://upnp.org/specs/av/UPnP-av-MediaRenderer-v1-Device.pdf

The HBM11 Media Server can be selected as playback source using any UPnP control point, e.g. BubbleUPnP.

Figure 24 shows the UPnP Media Server services provided by the HBM11 module when attaching a USB pendrive or disk to the USB host port.

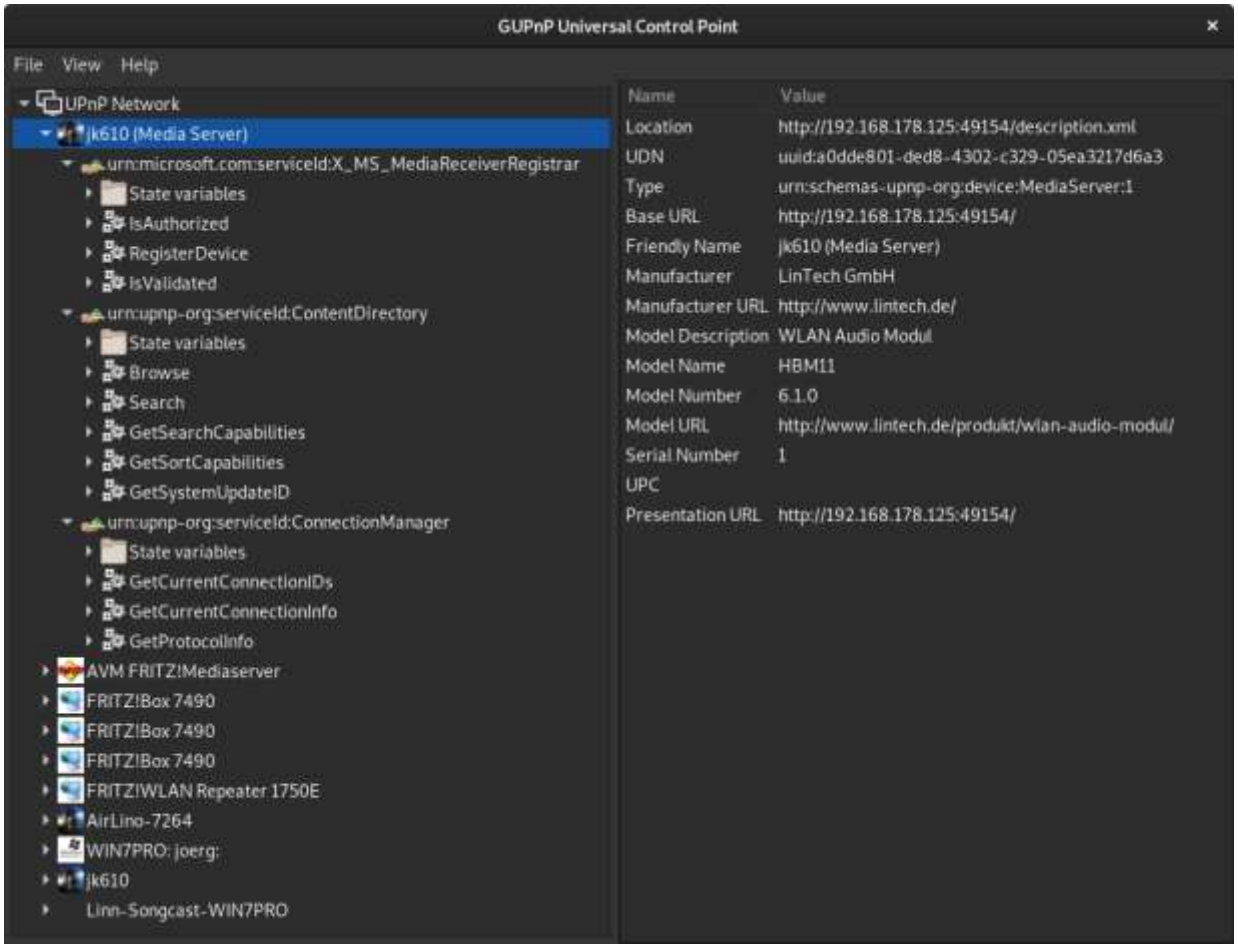


Figure 24 – UPnP Media Server started when plugging in a USB pendrive ord USB disk

8.2 OpenHome

OpenHome is an open standard and software platform for streaming digital audio from a variety of sources. The platform is built upon the UPnP standard, but simplifies access to services and enhances to capabilities of UPnP to provide improvements such as highly synchronized multi-room playback, managed playlists and gapless playback.

The name of the OpenHome device is the same name es the HBM11s device name.

8.3 AirPlay

AirPlay is a proprietary protocol stack developed by Apple Inc. that allows wireless streaming between devices of audio, video, screens, and photos. Originally, it was called AirTunes and only supported streaming of audio. Actually, the HBM11 module runs an audio-only receiver implementing the AirTunes v2 protocol. Any AirPlay Sender compatible to that protocol can directly stream to the HBM11.

The AirTunes v2 protocol uses UDP for streaming audio ans is based ont the RTSP network control protocol. The streams are transcoded using the Apple Lossless codec with 44100 Hz, 16-bit, 2 channels encrypted with AES. The stream is buffered for approximately 2 seconds before playback begins, resulting in a small delay before audio is output after starting an AirPlay stream.

AirPlay is the default AirPlay protocol.

8.4 AirPlay 2

AirPlay 2 offers some advantages of AirPlay:

- Improved audio buffering: instead of buffering only some seconds, buffered audio streaming allows to buffer minutes, depending on the streaming application
- Multiroom in iOS: similar to iTunes it is possible to stream to multiple AirPlay 2 receivers in iOS

AirPlay 2 can be enabled by the user using the HTTP API.

8.5 Internet radio (powered by TuneIn)

The HBM11 module provides the ability to listen to streaming audio from radio stations worldwide via TuneIn's powerful database.

8.5.1 Features

The main radio features of the HBM11 module are:

- play HTTP streams encoded with MP3 or Ogg/Vorbis
- manage radio stations in playlists and save them on the device
- automatic playback of the last radio station you listened to after turning on
- automatic playback of a favourite radio station or playlist after turning on
- remote control through the HTTP API

8.5.2 Playback

Playing a radio station is very simple. All you have to do is send a valid URL and optionally a radio station name with the **play** command to the HBM11 module. The **stop** command immediately stops the radio stream.

If the module is turned off while a radio station is playing, it will automatically restart the stream the next time it is turned on.

To always start with a specific radio station after switching on the module, regardless of what was played last, a favourite station can be defined via the **setfavouritestation** command.

The current playback status can be retrieved with the **query** command.

8.5.3 Playlists

Radio stations can be stored as playlists directly on the HBM11 module. This gives you access to all your playlists no matter which remote client you are using.

A playlist is saved using the **saveplaylist** command. Each playlist has a unique ID in the range of 0 to 128. You can give each list a short description, e.g. "News" or "Rock". Each playlist can contain up to 128 radio stations.

A playlist can be retrieved with the command **getplaylist**. The playlist is specified by its ID. If the playlist exists, the HBM11 module will return the corresponding playlist with its description and a list of stations.

The **playplaylist** command starts playback of a previously saved playlist.

If the module is turned off during playlist playback, it will automatically restart the next time it is turned on.

To always start with a playlist after switching on the HBM11 module, a favourite playlist can be defined with the command **setfavouriteplaylist**.

A playlist is removed from the HBM11 module with the command **rmplaylist**.

8.6 Bluetooth

The HBM11 module supports the A2DP source role. The following codecs are supported:

- SBC
- AAC
- aptX
- MP3

8.7 Spotify Connect

Spotify Connect allows the HBM11 audio module to stream music from Spotify directly, rather than via the phone. The Spotify app on the phone acts as the remote control.

By comparison, AirPlay - which also uses Wi-Fi - and Bluetooth stream music directly from the phone to the receiving device, which can wear down the battery more quickly.

For using Spotify Connect you need a Spotify account.

The easiest way to use Spotify Connect is with the Spotify app. You don't have to perform a set up routine to stream to the HBM11 module; if the HBM11 is on the network, your phone should be able to see it.

Once the app detects the HBM11 module, a "Devices Available" icon appears at the bottom left of the app. Pressing this will bring up a list of the players on your network. Once you choose the HBM11 to stream to, your music will begin playing. If you have a speaker group created within the AirLino App, you can now stream to all of them together using Spotify as well.

8.8 TIDAL Connect

Following in the footsteps of Spotify with Spotify Connect, TIDAL has released the TIDAL Connect, that allows its users to stream music directly to connect devices.

To listen to music on the HBM11 audio module via TIDAL connect select the content you want to play and then select the connect icon on the Now Playing page. This will bring up a list of nearby devices to stream through. Locate the HBM11 module and select to pair. Once paired you can use your TIDAL app as a remote control.

8.9 Qobuz

The HBM11 module supports playing songs from the commercial streaming service Qobuz. The HTTP API provides an easy way to listen to Qobuz songs and playlists. The main feature commands are:

- Pass the Qobuz credentials to the HBM11 module
- Play a Qobuz track
- Manage a list of Qobuz tracks in a playlist
- Play a list of Qobuz tracks

8.10 Multi-Room

The HBM11 module offers perfectly synchronized multi-room support by the following protocols:

- AirPlay and AirPlay 2
- Songcast

8.10.1 AirPlay and AirPlay 2

AirPlay is a proprietary family of protocols developed by Apple Inc. that allows wireless streaming between of audio, video, device screens, and photos, together with related metadata.

There are two types of AirPlay devices:

- AirPlay server that sends the audiovisual content
- AirPlay clients receiving that conent and rendering it on displays and speakers

The HBM11 module implements the audio parts of the AirPlay and AirPlay 2 client. It is compatible to AirPlay and AirPlay 2 server devices including:

- computers running iTunes (available on macOS and Windows)
- iOS devices such as iPhones, iPods, and iPads
- third party streamers such as Tuneblade, WHAALE or WHD Multiroom

Grouping

The AirPlay protocol does not have a concept of multi-room grouping. However, it is possible to stream to multiple devices synchronously.

Streaming

Figure 25 shows an example of an AirPlay setup with a MacBook running the iTunes desktop application. The operating system macOS running on the MacBook implements the AirPlay server part whereas the AirLinos implements the AirPlay client part.

The AirPlay protocol only supports unicast connections. iTunes is receiving the audio stream from some music network server. The AirPlay server running on the MacBook distributes the audio data as an AirPlay stream to every selected AirPlay client in the network using a point-to-point connection as shown in Figure 26..

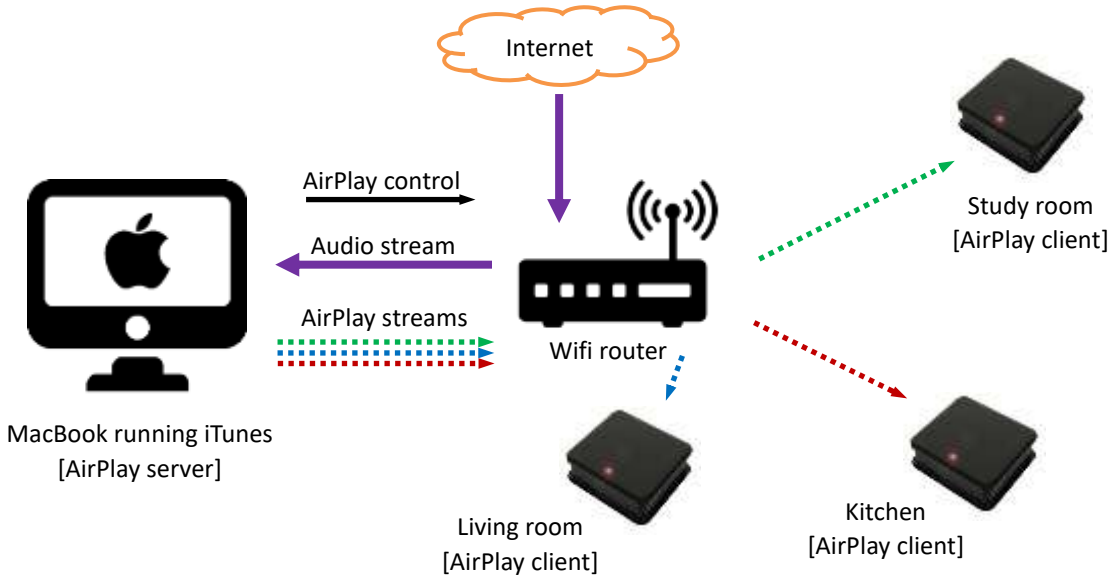


Figure 25 – Example of an AirPlay server distributing an audio stream with iTunes

AirPlay on iPhones only allows AirPlay connections to a single AirPlay client at the same time as shown in Figure 26. This limitation is removed with AirPlay 2, which support iOS to stream to multiple AirPlay 2 clients synchronously.

Because of the limitations of the AirPlay protocol the HBM11 audio module includes another multi-room protocol to offer all the freedom of distributing an audio stream to several audio devices.

8.10.2 Songcast

Songcast is an open protocol for broadcasting an audio stream in a network. It allows playing the same audio source synchronized on multiple players.

The protocol links two types of entities:

- A Songcast Sender broadcasting the audio stream
- Songcast Receivers receiving and playing the audio stream

In contrast to the receiver-only implementation of the AirPlay protocol, the HBM11 module can be run as a Songcast Sender and/or Receiver. This allows to set up a multi room group with one HBM11 module acting as a master and several modules acting as slaves.

If all members of a group are connected wirelessly to the network up to four receivers can be linked to the sender. If more receivers are to be used, the sender device must be connected to the network via Ethernet.

Grouping

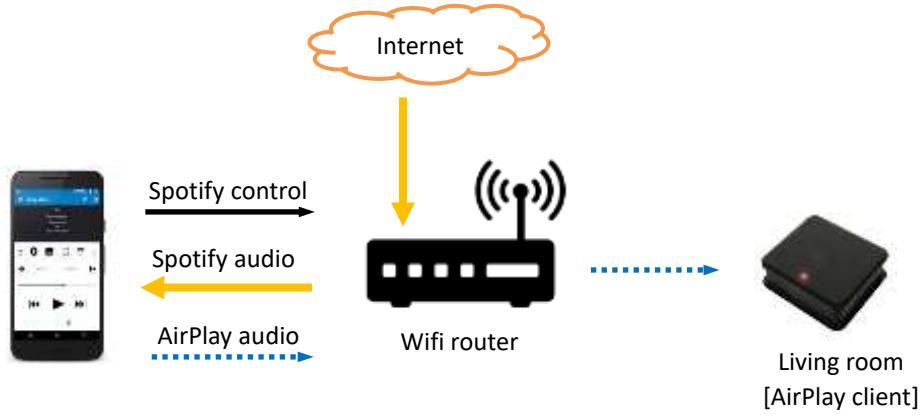


Figure 26 – Examples of iPhone only supporting a single AirPlay stream

By default, the HBM11 module operates in “Standalone Mode“, it is not a member of group.

To group multiple devices, e.g. like shown in Figure 27, each device must be configured using the songcast actions defined in the HTTP API.

Grouping is as easy as following two steps:

1. Enable the “Multi-Room Sender Mode” on the master device.
2. Link every receiver device to the sender to put the device into “Multi-Room Receiver Mode”.

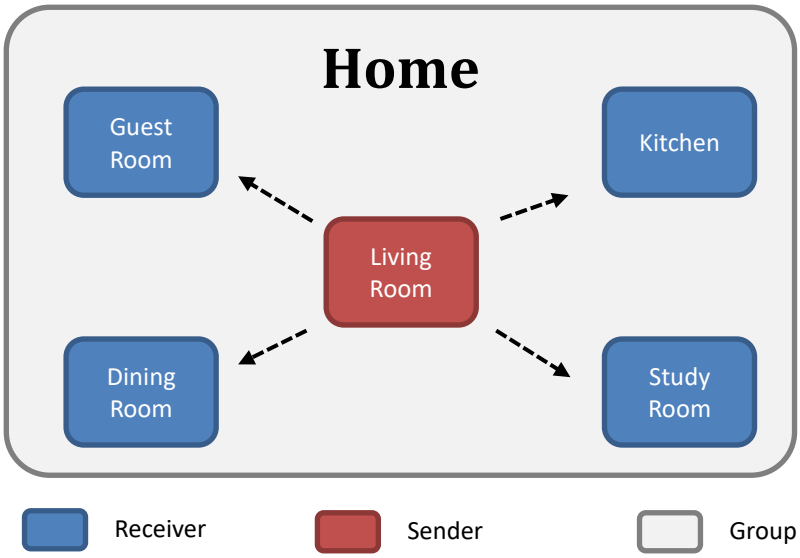


Figure 27 – Example of a Songcast Multi-Room Group of one Sender and four Receivers

For each receiver of the group, playback of the audio stream can be stopped and resumed individually. Furthermore, the volume of each receiver can be stopped individually, too.

Streaming

Figure 28 shows an example of a Songcast Sender broadcasting an audio stream received from a server using the Spotify Connect protocol to its connected Songcast Receivers. It contrast to AirPlay

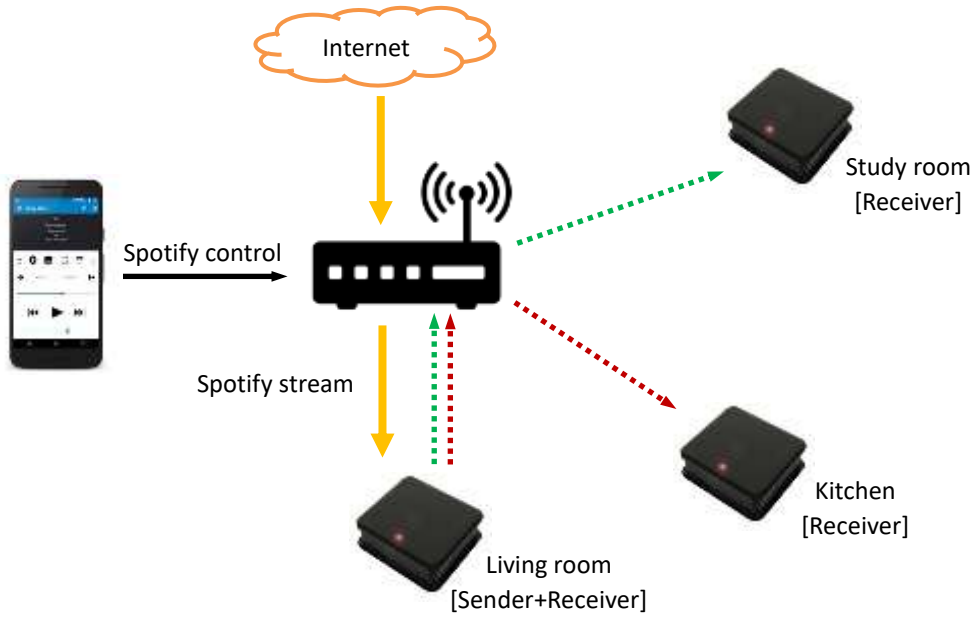


Figure 28 – Example of Songcast Sender broadcasting an audio stream from Spotify

8.11 Playback control

The HBM11 module can be extended with keys to control the audio playback:

- Volume Up
- Volume Down
- Play/Pause
- Stop
- Next
- Previous

8.11.1 AirPlay

Enabling remote control for AirPlay requires an AirPlay server supporting the Digital Audio Control Protocol (DACP). Once a key event has been triggered on the module, e.g. the “Stop” key was pressed by the user, an appropriate HTTP request is sent from the AirPlay client to the AirPlay server. Depending on the network connectivity it may need several dozen to hundred milliseconds until the server has received and processed the command sent by the client – for example in case of a “Stop” command, the server will end the playback stream.

8.11.2 UPnP and OpenHome

Besides controlling the audio playback with the keys any compliant UPnP media control point, e.g. BubbleUPnP can be used to. Therefore, the UPnP media control point has to be connected to the HBM11 who is acting as a UPnP media renderer.

8.11.3 Spotify Connect

Remote Control of a Spotify Player is possible if the player is connected to a HBM11 device.

Once a key event has been triggered on the module, e.g. the “Next” key was pressed by the user, an appropriate command is sent from the Spotify Connect client to the Spotify Player. Depending on the network connectivity it may need up to several hundred of milliseconds until the player has received and processed the command sent by the client.

Note, that the Spotify Player does handle a “Stop” command actually as a “Pause” command.

8.11.4 TIDAL Connect

Remote Control of a TIDAL Player is possible if the player is connected to a HBM11 device.

Once a key event has been triggered on the module, e.g. the “Next” key was pressed by the user, an appropriate command is sent from the TIDAL Connect client to the TIDAL Player. Depending on the network connectivity it may need up to several hundred of milliseconds until the player has received and processed the command sent by the client.

Note, that the TIDAL Player does handle a “Stop” command actually as a “Pause” command.

8.11.5 Bluetooth

Using the multimedia keys for controlling Bluetooth audio playback requires the Bluetooth AVRCP profile on the A2DP Bluetooth source.

8.12 Volume Control

The audio volume is controlled by a software control with a dynamic range from -51 to 0 dB with a resolution of 256 corresponding to a step of 0.2 dB.

The current volume is stored on the internal flash periodically. However, to make sure a volume change withstands an abrupt power-cut a delay of 10 seconds is necessary.

8.13 Channel Setting

The HBM11 module supports two physical audio channels. The following channel settings are supported:

- Stereo [L/R] (*Default*)
- Left [L/L]
- Right [R/R]
- Mono

The two settings “Left” and “Right” are particularly useful for mult-room applications.

The settings can be changed via the HTTP API.

8.14 Equalizer

The HBM11 module has a built-in 10-band software equalizer.

Frequency	Gain (dB)		
	min	default	max
31.25 Hz	-48	0	24
62.5 Hz	-48	0	24
125 Hz	-48	0	24
250 Hz	-48	0	24
500 Hz	-48	0	24
1 kHz	-48	0	24
2 kHz	-48	0	24
4 kHz	-48	0	24
8 kHz	-48	0	24
16 kHz	-48	0	24

The settings can be changed via the HTTP API. The default value is 66 for all bands.

8.15 Surround Sound

The S/PDIF interface allows to carry compressed digital audio for surround sound as defined by the standard IEC 61937. This mode can be used to connect the S/PDIF output of the HBM11 module to a home theatre amplifying receiver that supports Dolby Digital or DTS.

For correctly transmitting the encoded audio streams such as Dolby Digital the following conditions must be met:

- The volume must be set to the maximum value, which is 255,
- The equalizer must be set to its default settings, which means all bands must be set to 66.

9 AirLino® App

The AirLino® App is an easy to use application designed to run on iOS and Android mobile devices. Its key features are:

- Setup wizard to integrate the AirLino into the home network
- Graphical radio interface for managing favourite radio stations
- Firmware update over the air
- Control and modify a wide range of settings

The Figure 29 shows some screenshots of the latest AirLino app: a view of a list of radio stations, a detailed view of the station currently played, and a view of the settings page.

The AirLino app utilizes most of the functionality provided by the HBM11 HTTP.

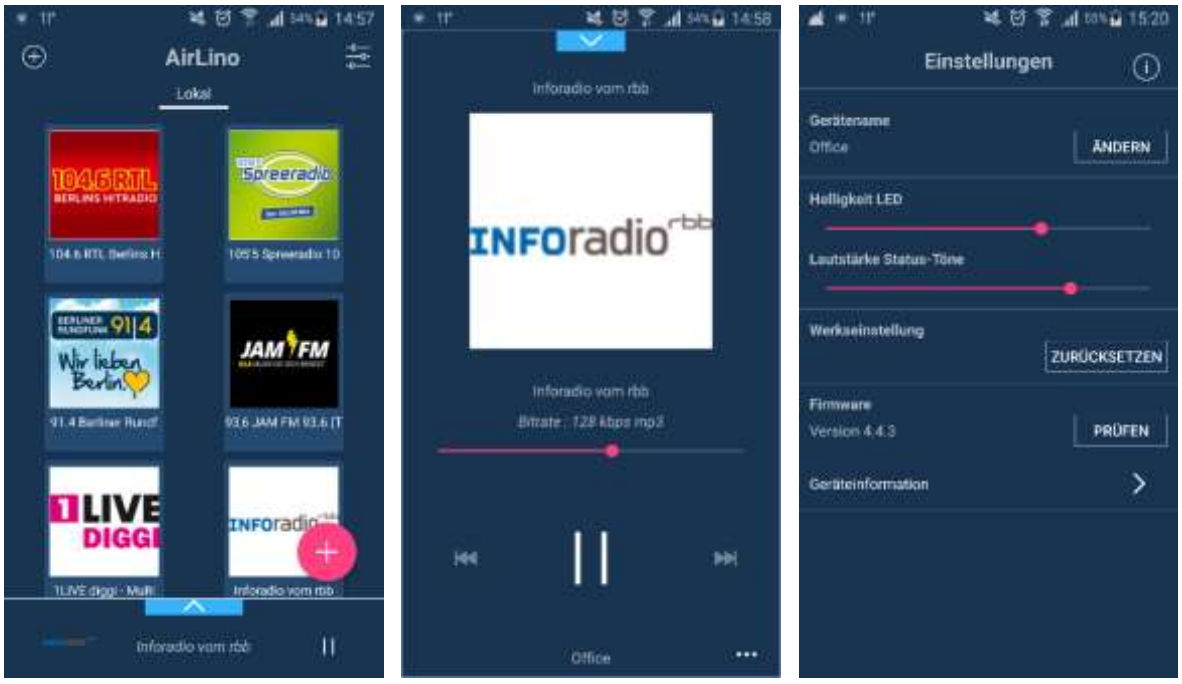


Figure 29 – Screenshots of the AirLino App

10 Customization

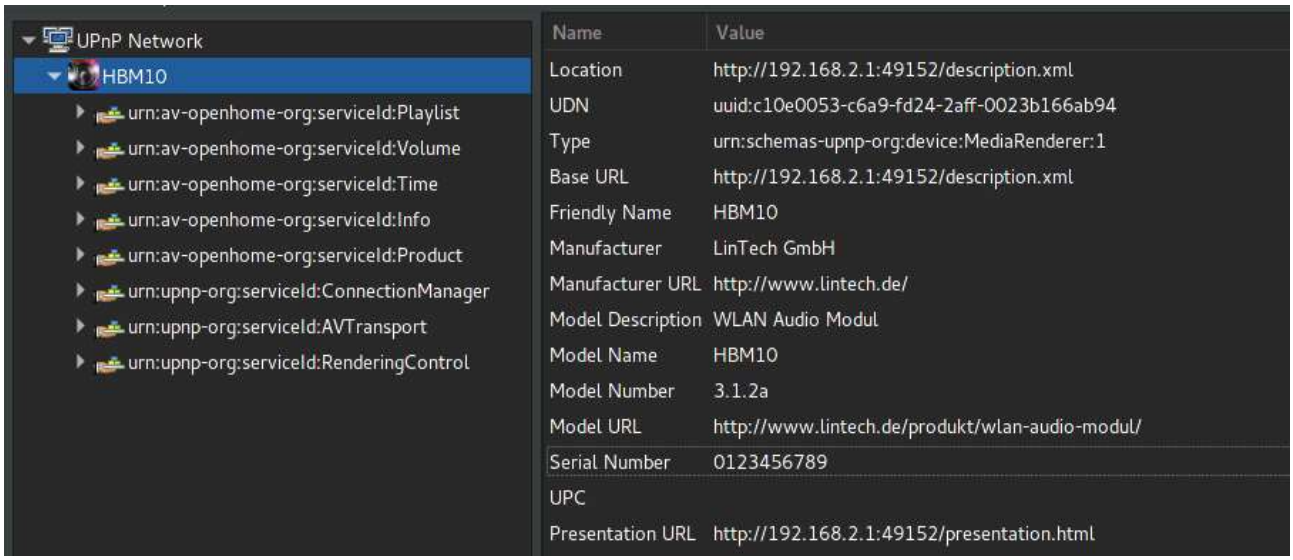
The device can be customized to represent your application. For customizing any of the values please contact the LinTech support team: lintech@lintech.de.

10.1 Device

The device parameters are mainly used for service description, e. g. by the UPnP protocol.

Parameter	Default	Description
Name	HBM11	A friendly name for identifying the device within the network.
Manufacturer		
Name	LinTech GmbH	The manufacturer name for the device.
URL	http://www.lintech.de	The manufacturer URL.
Model		
Name	HBM11	A model name for the device.
Description	WLAN Musikempfänger	A short description of the model.
URL	http://www.lintech.de/produkt/airlino-wlan-airplay-dlna-musikempfaenger/	The model URL.

These values are advertised by the device and can be viewed by most UPnP control points:



10.2 Wi-Fi

Currently, the SSID (Service Set Identifier) is the only customizable Wi-Fi parameter.

Parameter	Default	Description
SSID	HBM11-XXXX	The SSID (Service Set Identifier) name used as Access Point appended with the last four digits of the MAC address, eg. HBM11-A97B.

10.3 Audio

The status tones used for signaling the end of the boot up process and the state of the network connection can be turned off.

Furthermore, the state of the audio clients for AirPlay, Spotify, and UPnP/OpenHome can be configured. This way, certain audio services can be turned off for a certain model by default, e.g. if only internet radio is desired.

Parameter	Default	Description
Status Tones	on	Turn status tones on/off.
AirPlay	on	Enabled/disable the AirPlay client at startup.
Spotify	on	Enabled/disable the Spotify client at startup.
UPnP/OpenHome	on	Enabled/disable the UPnP/OpenHome client at startup.

11 Sample Applications

The sample applications are optional extensions to the HBM11 firmware and are not enabled by default. All samples can be modified or enhanced, depending on your very own requirements. Please contact the LinTech support team at lintech@lintech.de if you are interested in any of the samples or if you want to realise your own applications.

The following sample applications are ready to use:

- Serial-to-WiFi Bridge
- Line-in PCM-Recorder
- Line-in Multi-Room Player

11.1 Serial-to-WiFi Bridge

The HBM11 module can be used as a transparent bridge to carry serial (UART) traffic over an 802.11 wireless link. AT commands as described in the AT Command Reference are used to manage the configuration.

11.1.1 Block Diagram

The HBM11 is running a TCP Data Server listening on port 8990 and a TCP Control Server listening on port 8991.

The Data Server transparently bridges data between the UART interface and the TCP port 8990. The data is sent to the other interface as received – no processing or formatting is done. The default setting for the UART interface is 96008N1.

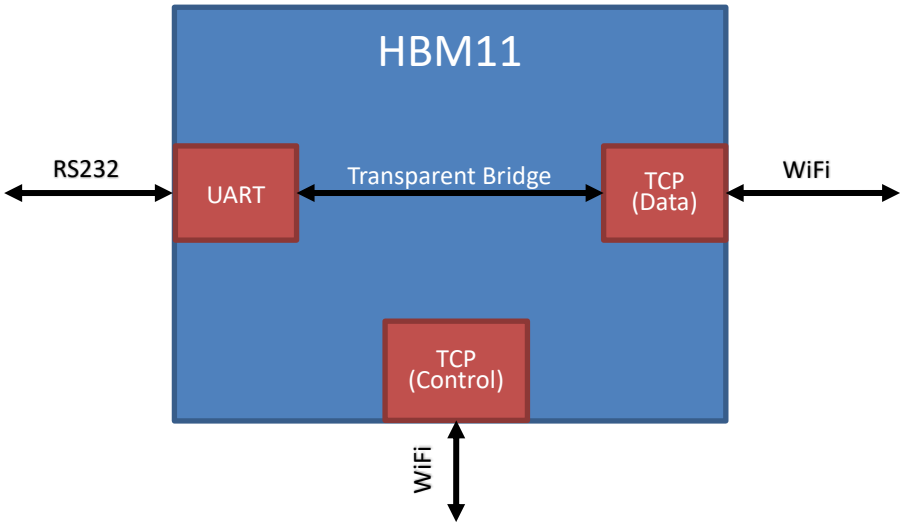


Figure 30 – Sample Application Serial-to-Wifi Bridge

The Control Server listens on TCP port 8991 for incoming AT commands as described in the AT Command Reference.

11.1.2 Workflow

The UART interface is opened and ready to user after the board powers up. If present, the UART interface uses the serial settings stored by the user otherwise the default settings for initialization.

To use the UART interface as a transparent bridge a TCP client has to establish a connection to the Data Servers port. Any data sent to the UART interface before a TCP connection is established will be lost.

If any configuration is requested, a TCP client has to establish a connection to the Control Servers port. This can be done at any point of time after the board powers up.

11.1.3 AT Commands Reference

11.1.3.1 AT Command Syntax

The “AT” or “at” prefix must be set at the beginning of each command line. To terminate a command line enter <CR>. Throughout this document, only the command lines are presented, <CR> is omitted intentionally.

Commands are usually followed by a response – <CR><LF>*response*<CR><LF>. Throughout this document, only the responses are presented, <CR><LF> are omitted intentionally.

The AT commands are case-insensitive and may be entered in either uppercase or lowercase letters and even can be mixed. Therefore, the following command lines are equivalent:

```
AT+UART?
at+uart?
At+Uart?
```

11.1.3.2 Result Codes

Result codes are messages sent from the Control Server to provide information about the execution of an AT command and the occurrence of an event. Two types of result codes are used:

- Final result codes
- Unsolicited result codes

A final result code marks the end of an AT command response. It is an indication that the Control Server has finished the execution of a command line. Two frequently used final result codes are **OK** and **ERROR**. Only one final result code will be returned for each command line.

The OK Final Result Code

The OK final result code indicates that a command line has been executed successfully by the Control Server. It always starts and ends with <CR><LF>.

The ERROR Final Result Code

The ERROR final result code indicates that an error occurs when the Control Server tries to execute a command line. After the occurrence of an error the Control Server will not process any remaining AT command. Like the OK final result code, the ERROR final result code always starts and ends with <CR><LF>. For common errors an error code follows the string “ERROR”, separated by a <SPACE> character, e.g.

```
ERROR 1
```

The following error codes are supported:

Error Code	Description
1	*Unknown command.* The command is not supported or contains a typo.
2	*Syntax error.* The command syntax is wrong, e.g. not all necessary parameters are set.
3	*Invalid range.* One or more parameters are out of range.

Unsolicited Result Codes

Unsolicited result codes are currently not used, but may be introduced with a new AT command.

11.1.3.3 Standard AT Commands

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>AT</td><td>Test command.
Response with OK when the control server is running.</td></tr>
<tr><td>A/</td><td>Repeat the last AT command.</td></tr>
<tr><td>ATE[<echo>]</td><td>Echo command.
Parameters:
<table>
<tr><th>Parameter</th><th>Type</th><th>Description</th></tr>
<tr><td>echo</td><td>Enum</td><td>0: Incoming characters will not be echoed.
1: Incoming characters will be echoed.</td></tr>
</table>
If <echo> is omitted, it defaults to 0.</td></tr>
<tr><td>AT&F</td><td>Factory defined configuration.
All configuration settings impacted by the AT&W command are reset to their default value.</td></tr>
<tr><td>AT&W</td><td>Stores the current configuration settings in non-volatile memory.
Parameters impacted by AT&W command:
<table>
<tr><th>Command</th><th>Parameter</th><th>Default</th></tr>
<tr><td>ATE</td><td><echo></td><td>0</td></tr>
<tr><td>UART</td><td><baud>
<data>
<parity>
<stop></td><td>9600
8
N
1</td></tr>
</table>
</td></tr>
</table>

11.1.3.4 Serial AT Commands

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>AT+UART=
<baud>,<data>,
<parity>,<stop></td><td>
Apply new UART settings.
Usage:
<table>
<tr><th>Parameter</th><th>Type</th><th>Description</th></tr>
<tr><td>baud</td><td>Integer</td><td>Supported baud rates:
<table>
<tr><td>300</td><td>28800</td></tr>
<tr><td>1200</td><td>38400</td></tr>
<tr><td>2400</td><td>57600</td></tr>
<tr><td>4800</td><td>115200</td></tr>
<tr><td>9600</td><td>230400</td></tr>
<tr><td>14400</td><td>460800</td></tr>
<tr><td>19200</td><td>921600</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td>Note: Other baud rates may work too, but are not supported.</td></tr>
<tr><td>data</td><td>Integer</td><td>Supported data bits:
5, 6, 7 or 8</td></tr>
<tr><td>parity</td><td>Char</td><td>Supported parities:
N: No parity
O: Odd parity
E: Even parity</td></tr>
<tr><td>stop</td><td>Integer</td><td>Supported stop bits:
1 or 2</td></tr>
</table>
Example:
AT+UART=9600,8,N,1
</td></tr>
<tr><td>AT+UART?</td><td>
Read current UART settings.
Response:
+UART=<baud>,<data>,<parity>,<stop>
Example:
+UART=9600,8,N,1
</td></tr>
</table>

11.2 Line-In PCM-Recorder

The analog audio signal is recorded as a lossless PCM stream and transferred from a TCP client to a remote TCP server.

If the HBM11 is properly configured and is powered up or resumes from standby mode, it automatically reconnects to the AP, establishs a connection to the TCP server and starts transmitting the PCM data stream.

If the HBM11 resumes from standby mode, it takes about 4 seconds until the first PCM data is transmitted (assuming the TCP server is running and accepting the TCP connection from the HBM11).

11.2.1 Basic Setup

The basic setup consists of an external audio codec connected to the I2S interface of the HBM11 audio module.

The external audio codec provides the line-in. The HBM11 runs the PCM recorder, which reads the PCM data from the audio driver. The PCM data are streamed by the TCP client to a remote device, e.g. a Rasperry Pi, running the corresponding TCP server.

Figure 31 shows the basic setup for the Line-in PCM-Recorder sample application.

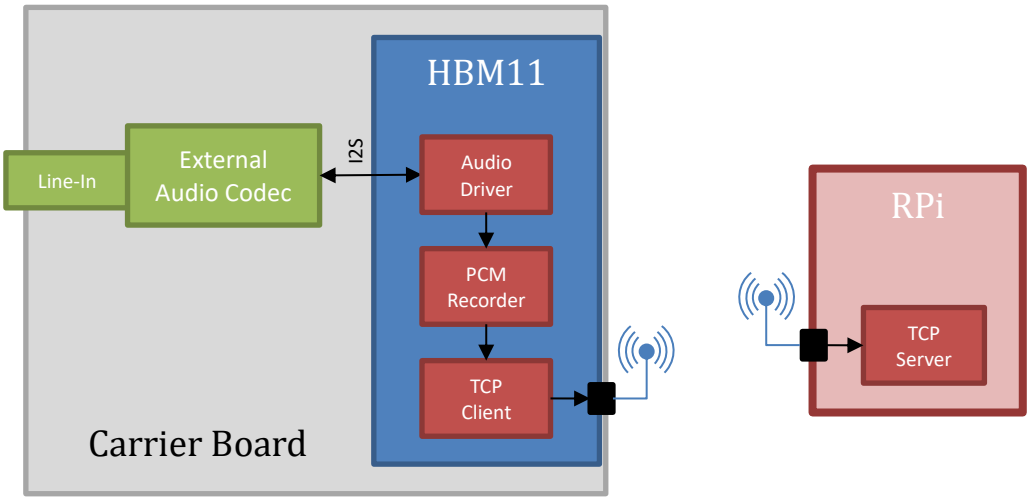


Figure 31 – Basic Setup of the Line-on PCM-Recorder Sample Application

11.2.2 PCM Format

By default, the recorded PCM has the following format:

- **Sampling frequency:** 48000 Hz
- **Bit depth:** 16 bit, little endian
- **Channels:** 1 (mono)

The PCM format can be modified.

11.2.3 Example TCP Server application

The following Node.js[4] script can be used for setting up a simple TCP server on a Linux system to easily evaluate the received PCM data:

```
/*jshint esversion: 6 */

const fs = require('fs');
const net = require('net');

const server = net.createServer((client) => {
  console.log('Client connected');
  fs.openSync('/tmp/sample.pcm', 'w');

  client.on('data', (data) => {
    fs.appendFile('/tmp/sample.pcm', data, (err) => {
      if (err) throw err;
    });
  });

  client.on('end', () => {
    console.log('Client disconnected');
  });
});

server.on('error', (err) => {
  throw err;
});

server.listen(9090, () => {
  console.log('Server bound');
});
```

The script will create a TCP server on port 9090. When the TCP client connects the received PCM data stream are appended into a file ‘sample.pcm’ in the ‘/tmp’ directory. This directory is on RAM on most Linux systems.

For evaluating the received PCM data stream the following command can be used on the Linux system:

```
aplay -t raw -c 1 -r 48000 -f S16_LE /tmp/sample.pcm
```

[4] The script was tested with the Node.js version 7.4.0.

11.3 Line-In Multi-Room Player

The analog audio signal is recorded as a lossless PCM stream and sent from the Songcast Sender to the connected Songcast Receivers.

11.3.1 Basic Setup

The basic setup consists of an external audio codec connected to the I2S interface of the HBM11 audio module.

The external audio codec provides the line-in. The HBM11 audio driver captures the PCM data from the line-in. The Songcast Sender streams the audio data lossless to all connected Songcast Receivers. Figure 31 shows the basic setup for the Line-in Multi-Room Player sample application.

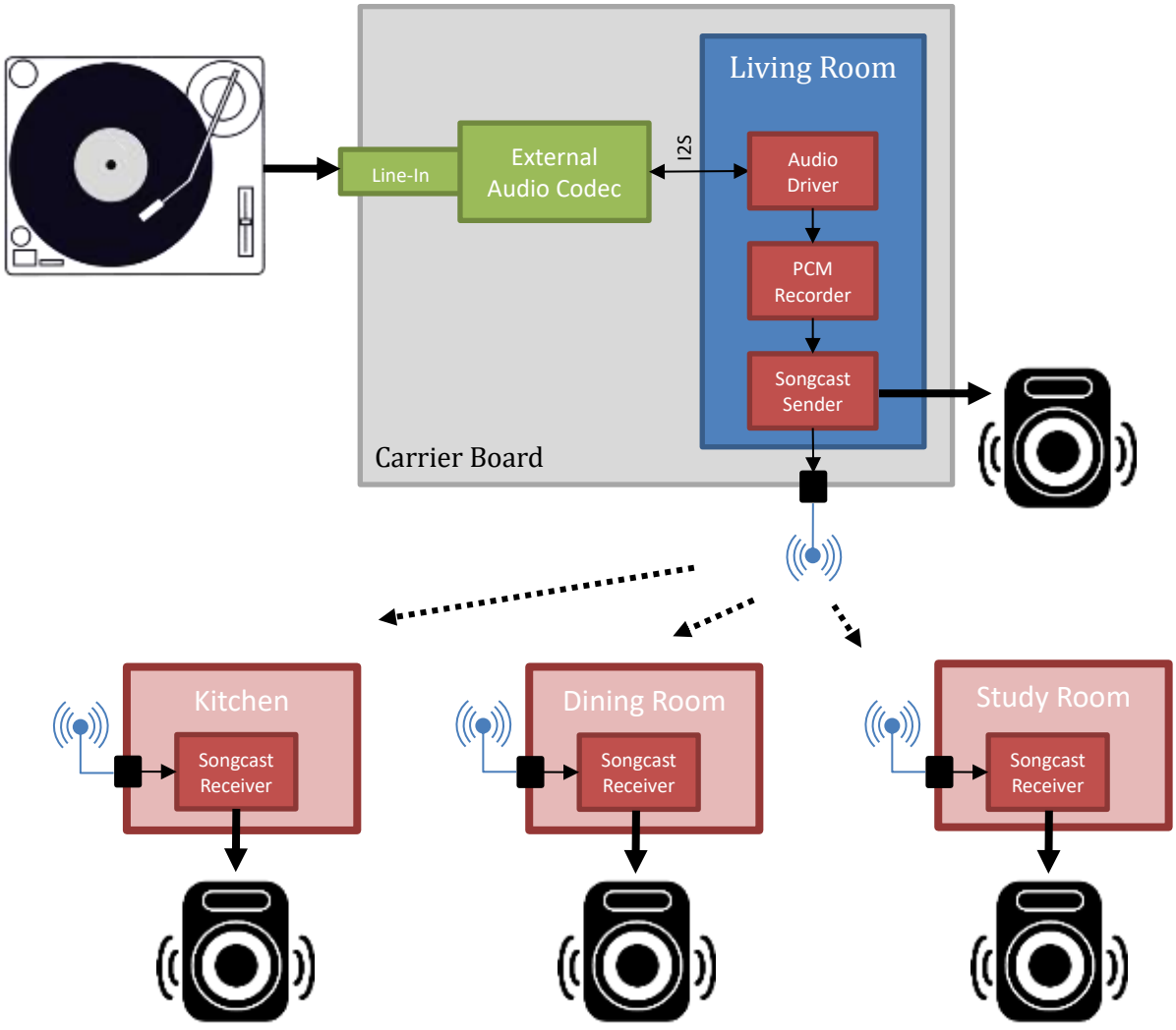


Figure 32 – Basic Setup of the Line-In Multi-Room Player Sample Application

11.3.2 PCM Format

By default, the recorded PCM has the following format:

- **Sampling frequency:** 44100 Hz
- **Bit depth:** 16 bit, little endian
- **Channels:** 2 (stereo)