Release 1.2.2

|  |  |
| --- | --- |
| Version |  |
| Date |  |
| Status |  |
| Document ID |  |
| Owner |  |
| Approved |  |
| Template | TSP00048 – Szczepan Piastka / Andrey Ponomarev – V2.0 |

Table of contents

[1 Introduction 3](#_Toc400973892)

[1.1 Portability 3](#_Toc400973893)

[1.2 High Performance 3](#_Toc400973894)

[1.3 Extensibility 3](#_Toc400973895)

[1.4 Conformance to Standards 3](#_Toc400973896)

[2 Release Details 4](#_Toc400973897)

[2.1 Reference Implementation 4](#_Toc400973898)

[2.2 Packaging and Availability 4](#_Toc400973899)

[2.3 Release Contents 4](#_Toc400973900)

[2.3.1 Software Components 4](#_Toc400973901)

[2.3.2 SDK 4](#_Toc400973902)

[2.3.3 Documentation 5](#_Toc400973903)

[2.4 New Features, Enhancements and Changes 5](#_Toc400973904)

[2.5 Issues Fixed 5](#_Toc400973905)

[2.6 Known Issues 5](#_Toc400973906)

[3 Description 5](#_Toc400973907)

[3.1 Test Report 6](#_Toc400973908)

[4 Usage Guide 6](#_Toc400973909)

[4.1 Build Information 6](#_Toc400973910)

[4.1.1 For target environment 6](#_Toc400973911)

[4.2 Code under GPL license 6](#_Toc400973912)

[4.3 Running the EAVB Stack 6](#_Toc400973913)

[4.3.1 Configuring the EAVB Stack 7](#_Toc400973914)

[4.3.2 Running the EAVB Stack 7](#_Toc400973915)

[5 Release History 7](#_Toc400973916)

[6 References 9](#_Toc400973917)

[7 Change History 9](#_Toc400973918)

# Introduction

The Symphony Teleca (STC) Ethernet AVB (EAVB or AVB) stack is a portable core implementation of EAVB protocols suitable for a wide range of hardware and software solutions. These release notes cover both the core STC EAVB stack and also the reference implementation on select target platforms that can be used for evaluation.

Portability, high performance, extensibility and conformance to standards are key overriding features of the STC EAVB stack.

## Portability

The EAVB core is implemented in highly portable C language suitable for the wide range of hardware and operating systems platforms. This core is isolated from the OS and hardware by an OSAL (Operating System Abstraction Layer) and a HAL (Hardware Abstraction Layer) which is customized for each HW/OS combination. See the "ST13-01059 Ethernet Audio Video Bridging (AVB) High Level Specification" [5] for more architecture details.

Here is a list of current ports of the STC EAVB stack.

* Linux (various kernel versions)
* Android (demo level)
* QNX
* FreeRTOS / OpenRTOS
* uC/OS III (Micrium)

## High Performance

AVB standards dictate requirements for delivery and transport latency which the STC EAVB stack fully meet. Real world solutions commonly require running many AVB streams on a single end-station. This is where performance becomes even more important. Please request our performance report for details of the STC EAVB stack performance and capacity.

## Extensibility

In addition to efficient porting, it is just as important that the STC EAVB stack integrate with media services on the hardware as well as into host applications. The STC EAVB stack defines two sets of APIs that allow these integrations without requiring source code to the core EAVB stack.

The "STC EAVB SDK" is the set of header files, samples and documentation that is used to integrate the STC EAVB stack into a host application. See the "ST13-00672 STC EAVB SDK Developer Guide" [1] for more details.

The "STC AVTP Interface Module SDK" is the set of header files, samples and documentation that is used to create modules (Interface Modules) that are source or sinks to the AVB data streams. See the "ST13-00669 STC AVTP Interface Module SDK Developer Guide" [2] for more details.

## Conformance to Standards

The following are the primary large feature sets supported in the STC EAVB stack.

| **Feature (or functionality)** | **Standard** |
| --- | --- |
| gPTP | IEEE 802.1AS |
| FQTSS | IEEE 802.1Q-2011, Clause 34 (Previously 802.1Qav) |
| SRP | IEEE 802.1Q-2011, Clause 35 (Previously 802.1Qat) |
| MAAP | IEEE1722 |
| AVTP | IEEE1722 and IEEE 1722a |
| AVDECC (internal beta) | IEEE1722.1 |

Testing details can be found in the document "STC\_test\_report\_L.1.2.2" [4].

# Release Details

## Reference Implementation

This STC core EAVB version is released as a reference implementation on these platforms:

* Texas Instruments Jacinto 5 (J5) with Linux OS
* Freescale i.MX 6 with Linux OS

## Packaging and Availability

This release is available as part of an evaluation license agreement or custom SoW agreement and it packaged as an archive that includes binaries as well as the SDKs previously mentioned.

## Release Contents

### Software Components

| **Component Name** | **Version** |
| --- | --- |
| STC Core EAVB Stack | 1.2.2 |

### SDK

| **Title** | **Version** |
| --- | --- |
| STC AVTP Interface Module SDK | 1.2 |
| STC EAVB SDK | 1.3 |

### Documentation

| **Title** | **Version** |
| --- | --- |
| STC AVTP Interface Module SDK Developer Guide[2] (Part of SDK) | 1.2 |
| STC EAVB SDK Developer Guide.doc [1] (Part of SDK) | 1.3 |
| Release Notes for Release L.1.2.2 | 1.2 |

## New Features, Enhancements and Changes

These are the primary feature enhancements and changes implemented since the last public release of the STC EAVB stack.

* Implementation of OSAL / HAL: Significant internal architecture changes to separate the platform independent elements of the core EAVB stack from those elements that are platform dependent. This results in reduced time to port to new platforms, improved stability on new platforms and reduced maintenance costs.
* AVTP Interface Module API changes: See the "ST13-00669 STC AVTP Interface Module SDK Developer Guide" [2] for details.

## Issues Fixed

These are issues that have been fixed since the last public release of the STC EAVB stack.

* Add PTP\_SETUP\_SIGNAL\_HANDLER
* Fix PTP shutdown
* Fixed AVB\_TRACE\_MODE\_FUNC\_TIME mode in trace system
* Eliminate mediaq and avtp\_time function pointers
* Change stcptp to stcPTP
* Move stcPtpSigHandler prototype
* Removed unused function stcPtpSigHandler()

## Known Issues

These are issues that may be noticed when running the STC EAVB core on a reference implementation platform. Note: For details of the protocol compliance tests see the separate document "STC\_test\_report\_L.1.2.2" [4]

|  |  |
| --- | --- |
| ID | 1. Description |
| 22 | On the TI J5 the talker end-station may report excessive CPU utilization. This does not appear to effect proper functioning of the AVB stack. Based on research it appears that some lower priority service is spinning either waiting for work or busy doing non-time sensitive background activities. |
| 54 | When using the ALSA device "plug:dmix" for mixing multiple audio stream the ALSA drive may experience under-flow errors. This is not related to the EAVB stack but also occurs when using this ALSA device for mixing multiple playbacks with the ALSA command line aplay tool. |

## Test Report

Testing details for this release can be found in the document "STC\_test\_report\_L.1.2.2" [4].

# Usage Guide

The following information relates to the construction and use of the STC Core EAVB stack on the Linux reference implementation.

## Build Information

This is a binary release, however, to run on the reference implementation target platform the Linux kernel needs to be patched.

### For target environment

Steps to build kernel for J5 and i.MX 6 board are described in document [3]

#### Additional information

| **Item** | **Value** |
| --- | --- |
| GIT TAG – tag that was added during integration process in Git respository | RELEASE\_L-1.2.2 |
| TI ezSDK patch version – patch used during building software | ezsdkL1.1.patch |

## Code under GPL license

There are some parts of code that are covered the Linux GPL license and are distributed as source code in every release, in accord with the GPL license. These files are attached to the official release package:

* Queuing discipline (files: sch\_avb.c; includes/avb\_sched.h)

All files covered by GPL license are found inside GPL\_code directory in the release package.

## Running the EAVB Stack

The STC EAVB stack is designed to have some amount of porting, customization and integration applied into final solutions. However, configuration and execution of the reference implementation is possible for test and evaluation purposes. There are two executables that need to be configured and run. These are the endpoint and the AVB host application.

The endpoint contains the SRP, MAAP and FQTSS components.

The AVB host manages the AVTP component including the stream lifecycle.

This section will outline steps needed to configure 2 end-stations, a talker end-station and a listener end-stations with a simple stream (echo stream) the sends text data from the talker to the listener end-station and outputs to the console on the listener end-station.

### Configuring the EAVB Stack

On the Linux reference implementation the configuration of the EAVB stack and streams are exposed via .ini files and a handful of command line parameters.

Note: the use of .ini files is only the method used to expose the configuration items for a binary build of the EAVB stack in the reference implementation. In custom implementations of the EAVB stack the configuration items can be programmatically set at run time.

1. On each end-station the .ini files, and all binary files, should be copied to ~/bin (/home/root/bin) on each board.
2. On each end-station the /etc/ld.so.conf file must be updated to include the location of the bin directory (/home/root/bin). After doing this the command ldconfig must be run to update the symbolic links.
3. On each end-station examine the "~/bin/endpoint.ini" file. If needed, change the "interface" setting to match the board's active Ethernet interface
4. On the listener end-station Inspect the "echo\_listener.ini" file; set the stream\_address to match the talker's MAC address

### Running the EAVB Stack

1. On each end-station start the endpoint.   
   $ ./stc\_avb\_endpoint endpoint.ini &
2. On the talker end-station start the host application  
   $ ./stc\_avb\_host echo\_talker.ini
3. On the listener end-station start the host application  
   $ ./stc\_avb\_host echo\_listener.ini
4. Look for text output on the listener console.

# Release History

|  |  |  |  |
| --- | --- | --- | --- |
| Release | Release Type | Release Date | Feature / Issue Fix |
| L-0.2 | Beta | 4/16/2013 | * gPTP- SW time stamping * AVTP – mpeg2ts, mjpeg * FQTSS – multi-queuing support |
| L-0.2.1 | Beta | 4/25/2013 | * gPTP – fixes for time stamping |
| L-0.3 | Beta | 6/5/2013 | * FQTSS- HW fifo support * AVTP – AVTP as shared library |
| L-0.3.1 | Beta | 6/17/2013 | * gPTP – fix for PTP not working correctly |
| L-1.0.0 | Alpha | 7/30/2013 | * Protocols implemented according to standards 802.1AS, 802.1Qat, IEEE1722, 802.1Qav, IEEE1722 |
| L-1.0.0 | Production | 8/6/2013 | * Logging updated * Memory usage corrected * Non-blocking RX and TX callbacks to the mpeg2ts\_gst |
| L-1.1.0 | Production | 9/4/2013 | * Logging cleanup and enhancements * Exceptional situation handling * Graceful shutdown * Additional statistics during work * SRP API for stream status * Operation in non-SRP environment * SRP works only on asCapable links |
| L-1.1.1 | Production | 10/23/2013 | * Changed AVTP and gPTP fields to follow the standard * Improvements in Talker advertise mechanism * Correction in LeaveAll event treatment |
| L-1.1.2 | Production | 11/19/2013 | * Corrections inside AVTP frames * Fixed problem with gPTP crashing * Release for i.MX 6 platform |
| L-1.1.3 | Production | 12/19/2013 | * Corrections for AVTP headers in MJPEG mapping |
| L-1.1.4 | Production | 05/07/2014 | * Added malloc NULL buffer handling * Created shared library with LGPL code * Changed handling of private contract between mpeg2ts\_map and mpeg2ts\_file\_intf * Streaming with proper bitrate reimplemented * Max bitrate computation moved to separate file * Added MPEG2-TS streaming solution * Added bitrate sending control feature * Config option intf\_nv\_enable\_proper\_bitrate\_streaming is enabled by default * Change FQTSS delay resolution to nanoseconds |
| L-1.2.2 | Production |  | * Implementation of OSAL / HAL * AVTP Interface Module API changes * Add PTP\_SETUP\_SIGNAL\_HANDLER, fix PTP shutdown * Fixed AVB\_TRACE\_MODE\_FUNC\_TIME mode in trace system * Eliminate mediaq and avtp\_time function pointers * Change stcptp to stcPTP * Move stcPtpSigHandler prototype * Removed unused function stcPtpSigHandler() * Added AVB\_LOG\_LEVEL\_STATUS. Convert some log INFO to STATUS |

# References

|  |  |
| --- | --- |
| [1] | ST13-00672 STC EAVB SDK Developer Guide |
| [2] | ST13-00669 STC AVTP Interface Module SDK Developer Guide |
| [3] | Kernel\_build.txt |
| [4] | STC\_test\_report\_L.1.2.2 |
| [5] | ST13-01059 Ethernet Audio Video Bridging (AVB) High Level Specification |

# Change History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Change from Previous | Name | Status |
| 1.0 | 8Aug2014 | Document created | Michal Chrostowski | Draft |
| 1.1 | 6Oct2014 | General edits and reorganization | Ken Carlino | Draft |
| 1.2 | 8Oct2014 | Proofread and edit | Gordon Bechtel | Draft |
| 1.3 | 13Oct2014 | Added configuration and startup information | Ken Carlino | Draft |