Suggestions for P1722.1c

Authors:
- Florian Zeitz — m2lab Ltd.
- Marc Schettke — m2lab Ltd.

This document contains the author's suggestions for the upcoming revision of IEEE Std 1722.1-2021.

In Section 1 we outline new functionalities we deem useful. Currently we realize some of these as vendor specific extensions. In Section 2 we list corrections to the current standard text that we think are uncontroversial (e.g. typos, wrong section references). In Section 3 we list areas where we think clarifications to the standard are necessary. Some of them with concrete suggestions, some as starting point for working group discussion.

Contents

1. Additions

1.1. Add a Flag Indicating Unsolicited Sequence IDs Are Per Controller

ATDECC controllers usually want to ensure that their local state matches that of the controlled entity. Therefore, it is necessary to detect when an unsolicited notification was lost. To that end IEEE Std 1722.1-2021 already explicitly permitted sending consecutive sequence IDs per registered controller instead of utilizing a common counter. However, without vendor extensions, standard controllers have no way to determine whether this is the case. Consequently they cannot determine whether unsolicited notifications were lost.

We suggest adding a new flag to the REGISTER_UNSOLICITED_NOTIFICATION response indicating which strategy is being employed for tracking sequence IDs.

Additionally a system may be receiving unsolicited notifications intended for multiple different controllers. Therefore, controllers need a way to determine whether an unsolicited notification was intended for them to appropriately interpret the contained sequence ID.

For that reason we additionally suggest that this flag also governs that the **controller_entity_id** field is set to the entity ID of the controller receiving the unsolicited notification. Currently it might instead be set to the entity ID of the controller sending the original command (cf. Section 3.1).

1.1.1. Suggested text changes

Modify "Table 7-147—REGISTER_UNSOLICITED_NOTIFICATION Flags" to look as follows:

Bit	Field Value	Name	Description
31	00000001_{16}	TIME_LIMITED	The registration will automatically timeout and be removed if it is not renewed.
30	00000002_{16}	SEQ_ID_PER_CONTROLLER	The entity maintains a separate unsolicitedSequenceID per registered controller.
0-29	—	—	Reserved for future use.

At the end of section "7.4.37. REGISTER_UNSOLICITED_NOTIFICATION Command" insert a new subsection as follows:

> **7.4.37.3 Separate Sequence IDs per Controller**
>
> The SEQ_ID_PER_CONTROLLER flag is used to indicate to a controller that the **sequence_id** field in unsolicited notifications will be filled from a separate counter per controller. This means a controller can assume consecutive notifications will contain consecutive **sequence_id** values unless a notification is lost in transit.
>
> The SEQ_ID_PER_CONTROLLER flag shall only be set on a response and shall be ignored if present in a command. An entity shall set the SEQ_ID_PER_CONTROLLER flag if and only if it makes use of the option provided in 9.3.5.2.6. (*Note to the editor*: section number of "unsolicitedSequenceID" in AEM Entity State Machine) to use separate unsolicitedSequenceID counters per registered controller and sets the **controller_entity_id** field of the unsolicited notification to the registered controller.

1.1.2. Alternatives

Instead of adding a flag to REGISTER_UNSOLICITED_NOTIFICATION this could also be indicated by a new capability in the ADP **entity_capabilities** field. The benefit might be that this information is then also available to controllers receiving unsolicited notifications by acquiring an entity instead of registering for them. However, room for capability flags is scarce and already almost at capacity.

1.2. Control Types to Indicate Lock and Sync Status of Received Clock Signals

Entities supporting external clock sources may want to indicate whether a valid clock signal is being recognized and if it is in sync with the currently active clock source. To that end we propose adding two new control types, one for each of these purposes.

1.2.1. Suggested text changes

To table "Table 7-98—Control Types" add the following rows:

Value	Name	Description	Clause
$90e0f0000000001d_{16}$	CLOCK_LOCK	Clock Source's Lock State	7.3.5.35
$90e0f0000000001e_{16}$	CLOCK_SYNC	Clock Source's Sync State	7.3.5.36

In section "7.3.5. Control Types" insert the following two subsections:

> **7.3.5.35. Clock Lock State Control (CLOCK_LOCK)**
>
> The CLOCK_LOCK control is used to report whether available clock sources are currently locked to a frequency and if so which frequency.

A CLOCK_LOCK control shall be implemented using any linear value type with a sufficient range and resolution for the implementation. For each value the **minimum** and **maximum** shall represent the valid range of measurements. The **default** shall be zero (0). The **current** value shall be as currently measured and may be beyond the range specified by **minimum** and **maximum** indicating the clock source is not locked. An unlocked state should be represented by a value of zero (0). The **unit** field shall have a **multiplier** which provides the resolution required for the measurement and a **code** set to HERTZ.

A CLOCK_LOCK control shall be read only.

A CLOCK_LOCK control may be implemented as a child of a Unit, a Port or a Jack.

7.3.5.36. Clock Sync State Control (CLOCK_SYNC)

The CLOCK_SYNC control is used to report whether available clock sources are in sync with the clock currently active in the associated clock domain.

A CLOCK_SYNC control shall consist of one or multiple boolean values implemented using a CONTROL_LINEAR_UINT8 value type, with a **minimum** of zero (0), a **maximum** of 255, a **step** of 255, and a **unit** field with a **multiplier** of zero (0) and a **code** of UNITLESS.

A CLOCK_SYNC control shall be read only.

Zero (0) shall indicate that the clock source is out of sync, and 255 shall indicate that the clock source is in sync with the clock currently active in the associated clock domain.

A CLOCK_SYNC control may be implemented as a child of a Unit, a Port or a Jack.

1.3. Control Type for Controlling the Brightness of UI Elements

Entities may want to provide users with control of the brightness of various UI elements, such as metering LEDs, displays or power indicators. We propose adding a new control type to allow controlling such elements.

1.3.1. Suggested text changes

To table “Table 7-98—Control Types” add the following row:

Value	Name	Description	Clause
$90e0f0000000001f_{16}$	UI_BRIGHTNESS	Brightness of user interface elements	7.3.5.37

In section “7.3.5. Control Types” insert the following subsection:

7.3.5.37 User Interface Brightness Control (UI_BRIGHTNESS)

The UI_BRIGHTNESS control is used to disable or dim illuminations of various user interface components.

A UI_BRIGHTNESS control shall be implemented using any linear value type with a sufficient range and resolution for the implementation and a **unit** field with a **multiplier** of zero (0) and a **code** of UNITLESS or PERCENT.

A value of zero (0) should indicate that illumination is disabled.

A UI_BRIGHTNESS control may be implemented as a child of any object that can contain a Control.

1.3.2. Alternatives

There is an existing BRIGHTNESS control type, however it is tailored to video signals. It could be expanded to encompass this use case.

2. Corrections

2.1. Typos in Definition of Audio Scale Control Type

The definition of the AUDIO_SCALE control contains multiple typos. The unit “dBu” is consistently misspelled as “dBU”. The reference to the definition of DBU points to the wrong section.

2.1.1. Suggested text changes

In section “7.3.5.32. Audio Scale Control (AUDIO_SCALE)” replace all occurrences of “dBU” with “dBu”.

Replace “DBU (7.3.4.23)” referencing “7.3.4.23. Level and Loudness quantity” with “DBU (7.3.4.7)” referencing “7.3.4.7. Voltage quantity”.

2.2. Remove mentions of “this corrigendum”

In section “7.3.6.2.7. Values format for Sample Rate Type (CONTROL_SAMPLE_RATE)” there is a note mentioning “this corrigendum”. This seems like an oversight from integrating the corrigendum of IEEE Std 1722.1-2013 as it makes no sense outside a corrigendum.

2.2.1. Suggested text changes

In section “7.3.6.2.7. Values format for Sample Rate Type (CONTROL_SAMPLE_RATE)” replace “this corrigendum” with “IEEE Std 1722.1-2013/Cor 1-2018”.

2.3. Typo in Frequency Control Definition

Section “7.3.5.66. Frequency Control (FREQUENCY)” is missing an “as”.

2.3.1. Suggested text changes

In section “7.3.5.66. Frequency Control (FREQUENCY)” replace “A FREQUENCY control is implemented any value type providing sufficient range and resolution” with “A FREQUENCY control is implemented *as* any value type providing sufficient range and resolution”.

2.4. Typos in TIME_LIMITED flag definition

The section name is not correctly title-cased and a stray “this” occurs in the text.

2.4.1. Suggested text changes

Rename section “7.4.37.2. Time limiting” to “7.4.37.2. Time Limiting”.

In that section replace “When the TIME_LIMITED flag is set *this* by the ATDECC Controller” with “When the TIME_LIMITED flag is set by the ATDECC Controller”.

3. Clarifications

3.1. Content of Fields in Unsolicited Notifications

The contents of unsolicited notifications generated by commands are currently unspecified. Of course it is reasonably easy to infer that the contents should equal those of the solicited response except for the **u** bit being set. We still think this should be explicit.

Additionally the contents of the **controller_entity_id** field are open to interpretation. It could either be set to the entity ID of the controller where the command originated, or to the entity ID of the controller intended as recipient of the unsolicited notification. Both have their merits. The former allows determining which controller was responsible for making a change. The latter allows controllers to

determine which notifications were meant for them, mostly important when multiple controller are running on the same system (cf. Section 1.1).

3.2. Representation of IP addresses

Currently various fields may contain IP addresses where each field definition repeats how to handle IPv4 addresses. There is also no explicit specification on how to encode IP(v6) addresses.

To avoid repetition and to add clarity we propose adding a new section specifying the encoding of IP addresses.

3.2.1. Suggested text changes

In section "4. Other Information" insert a new subsection as follows:

> **4.3. Encoding of IP addresses**
>
> Some PDU fields contain IP addresses. These shall always be encoded in IPv6 address format. I.e. as a 128 bit numbers with the most significant octet occurring first. Where the IP address is actually an IPv4 address the IPv4-Mapped IPv6 Address format specified in section 2.5.5.2 of RFC 4291 shall be used.

Remove the phrase "If the stream is using IPv4 then the IPv4 to IPv6 address translation of 2.5.5.2 of RFC 4291 is used to encode the address." throughout the following sections:

- 7.4.15. SET_STREAM_INFO Command
- 7.4.16. GET_STREAM_INFO Command
- 8.2.1. ATDECC Connection Management Protocol PDU

Alternatively the phrase could be replaced with a reference to the newly added section.

3.3. Content of mappings Field in ADD/REMOVE_MAPPINGS Error Responses

Many commands return the current value of a setting even in the error case. For ADD/REMOVE *MAPPINGS which are incremental the interpretation of the **mappings** field in the error case is not clearly defined. E.g. it would be just as plausible that the mappings returned are those that were still added/removed before the error occurred, as mappings always being ignored on error.

3.3.1. Suggested text changes

In the sections

- 7.4.45.1. (ADD_AUDIO_MAPPINGS Command and Response Format)
- 7.4.48.1. (ADD_VIDEO_MAPPINGS Command and Response Format)
- 7.4.51.1. (ADD_SENSOR_MAPPINGS Command and Response Format)

append the following text:

> If an error occurs an entity shall ensure that no mappings are added. In this case the **mappings** field in the response shall contain no mappings.

In the sections

- 7.4.46.1. (REMOVE_AUDIO_MAPPINGS Command and Response Format)
- 7.4.49.1. (REMOVE_VIDEO_MAPPINGS Command and Response Format)
- 7.4.52.1. (REMOVE_SENSOR_MAPPINGS Command and Response Format)

append the following text:

> If an error occurs an entity shall ensure that no mappings are removed. In this case the **mappings** field in the response shall contain no mappings.

3.4. Status Code for Commands Supporting Only Certain Descriptor Types

Various commands require that the **descriptor_type** field is set to one of a fixed set of descriptors. One could argue for returning either BAD_ARGUMENTS or NOT_SUPPORTED when this requirement is not fulfilled. We think it would be worthwhile to recommend one over the other, making the intended distinction between both status codes more clear.

3.5. Lack of Normative Language for Control Type Definitions

Definitions of control types in section "7.3.5 Control Types" generally lack normative language (*shall*, *may*, *should*). Definitions of the value type and unit to use, as well as interpretation of certain values most of the time use a plain "is".

3.5.1. Suggested text changes

Add wording using "shall" to definitions of value type, unit and value interpretation throughout section "7.3.5 Control Types".