

SLIMbus Audio Channel Setup

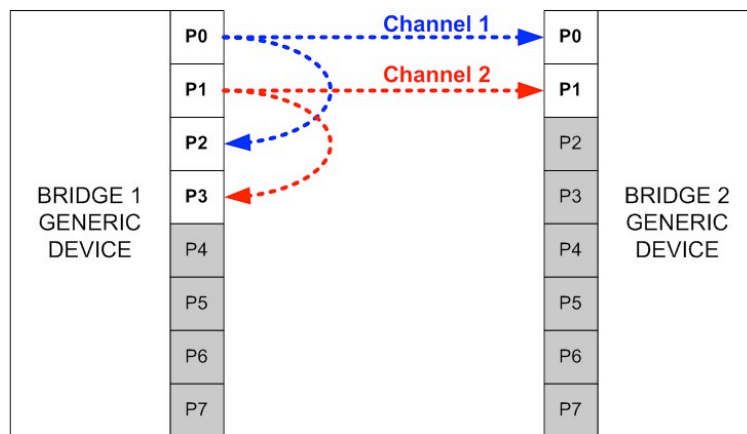


Table of Content

1. Purpose of the document	2
2. Presentation of the equipments	3
3. Collecting all the necessary information	4
3.1. Identifying the SLIMbus devices	4
3.2. Identifying the Data Ports	5
3.3. Bus Configuration	6
4. Practical Experimentation	9
4.1. What do we want to do?	9
4.2. Finding the Enumeration Addresses	9
4.3. Finding the right data ports	12
4.4. Configuring the bus	14
4.5. Creating the Data Channels	14
4.6. Script review	18

1. Purpose of the document

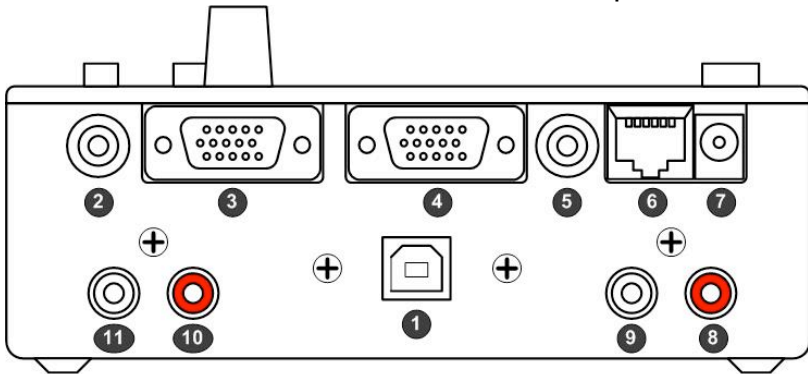
This document will guide the user through all the necessary steps to establish an audio stream between two SLIMbus devices. The procedure will list all the information that are necessary to be able to feed the right parameters in the SLIMbus messages and to put the SLIMbus in the right configuration.

We will use the **LnK** SLIMbus Audio Bridge as a SLIMbus audio device for the purpose of the exercise.

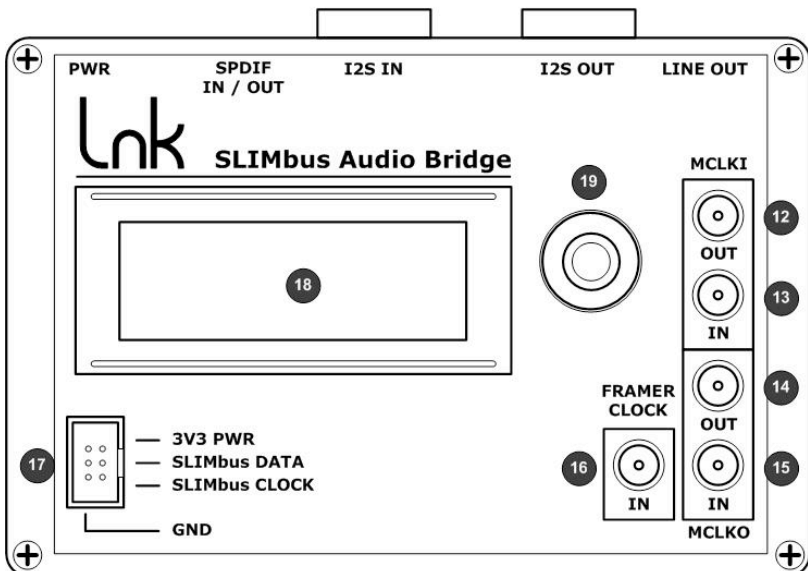
A basic knowledge of the SLIMbus specification and of the operation of the SLIMbus Audio Bridge is required to follow the various explanations given in this document. Refer to the SLIMbus Specification document and to the SLIMbus Audio Bridge HW and SW user manuals for more information.

2. Presentation of the equipments

The tests described in this document will require the SLIMbus Audio Bridge. An audio analyzer will also be required for the audio tests. Any kind of audio analyzer is suitable. However, the APx500 models from AudioPrecision are especially suited for these tests. For detailed information about these tools, please refer to their respective user manual.



- 1 - USB connector
- 2 - Analog line output (mini-jack)
- 3 - I2S multichannel output
- 4 - I2S multichannel input
- 5 - SPDIF IN / OUT (mini-jack)
- 6 - Factory Service
- 7 - Power supply
- 8 - SPDIF OUT
- 9 - SPDIF IN
- 10 - Analogue OUT (R)
- 11 - Analogue OUT (L)



- 12 - Audio Master IN clock out
- 13 - Audio Master IN clock in
- 14 - Audio Master OUT clock out
- 15 - Audio Master OUT clock in
- 16 - Framer clock input
- 17 - SLIMbus connector
- 18 - Display
- 19 - Rotary knob + push

The SLIMbus Audio Bridge will transmit (or receive) digital audio streams on (from) SLIMbus. The bridge features SPDIF input and output and I2S multichannel input and output. The I2S interface use the Left Justified data format and 64 bits per frame. The I2S IN/OUT multichannel interface is pin/signal compatible with the AudioPrecision DSIO interface of the APx500 machines when using the special cables delivered with the bridge.

The 8 port bridge needs a power supply and needs to be connected to a PC through USB to be able to feed the SLIMbus messages on the bus.

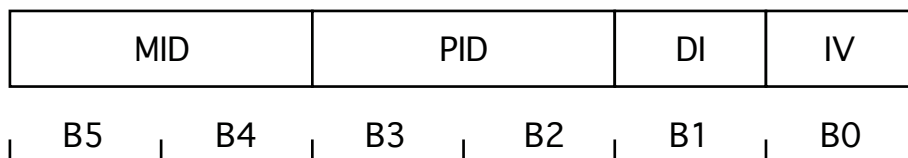
3. Collecting all the necessary information

In this section, we will review in detail all the information that are necessary to be able to setup an audio stream.

3.1. Identifying the SLIMbus devices

In order to communicate with a SLIMbus device, its Enumeration Address must be known. The Enumeration Address (**EA**) is a unique 48 bits value, often shown in the form of an hexadecimal value of 12 characters. The EA is split into 4 fields:

- Manufacturer ID (**MID**). This is a 16 bits value that uniquely identifies the manufacturer of a component.
- Product ID (**PID**). This is a 16 bits value that uniquely identifies a specific product in the portfolio of a Manufacturer.
- Device Index (**DI**). This is a 8 bits value that uniquely identifies the SLIMbus device inside a SLIMbus component.
- Instance Value (**IV**). This is a 8 bits value that uniquely identifies multiple instances of the same component in a system.



Note that a SLIMbus Component may contain many SLIMbus devices, each with their own **EA**. Therefore, it is very important to identify which of these devices will be involved in the audio streaming. For a given SLIMbus component, the **MID**, the **PID** and the **IV** fields will be common to all the devices of that component.

The LnK SLIMbus Audio Bridge has the following devices:

Device	MID	PID	DI	IV	EA
Interface	0x01C1	0x0001	0x00	0x00	0x01C100010000
Framer			0x01		0x01C100010100
Generic			0x02		0x01C100010200

The Audio Bridge has the possibility to use 0 or 1 for the InstanceValue, allowing two bridges to be connected on the same SLIMbus. When the Instance Value (Component Address) is set to 1, the Enumeration of the bridge devices will be:

Device	MID	PID	DI	IV	EA
Interface	0x01C1	0x0001	0x00	0x01	0x01C100010001
Framer			0x01		0x01C100010101
Generic			0x02		0x01C100010201

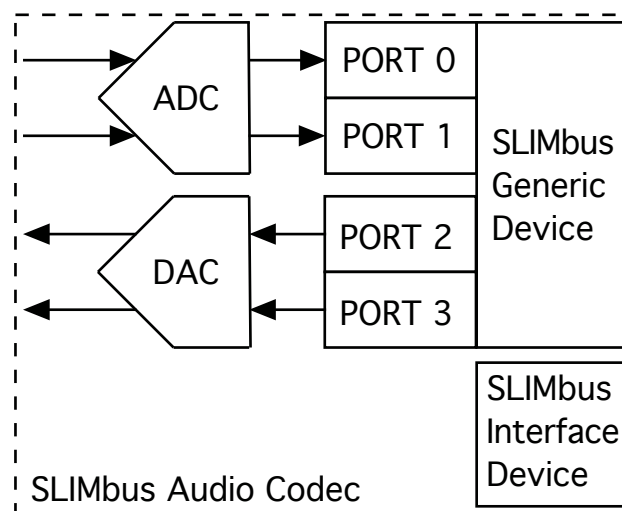
We need to identify which of these devices will be able to stream or receive audio channels. This information is usually provided in the data sheet of the chipset.

In the particular case of the SLIMbus Audio Bridge, the Generic device is capable of transmitting or receiving audio samples through SLIMbus data channels. The SLIMbus Audio Bridge Generic device has 8 data ports.

3.2. Identifying the Data Ports

A SLIMbus device can only transmit or receive data through data ports. It is therefore crucial to know about the device data ports. The device data sheet must give all information about the data ports and the type of function using or producing these data.

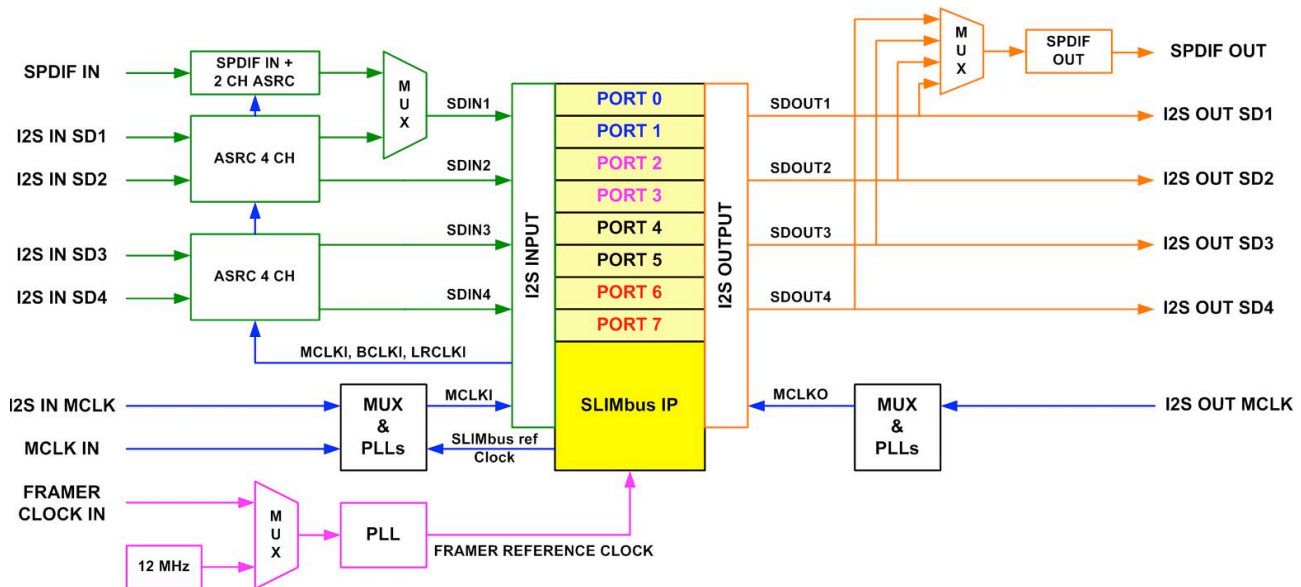
The following figures shows an possible implementation of a SLIMbus Audio Codec. We can see that the data ports are actually fulfilling a well defined task. Port 0 and Port 1 are attached to an ADC and will transmit audio samples. Port 2 and Port 3 are attached to a DAC and will therefore receive audio samples.



Port 0 and 1 are **Source** Ports: they source data to the SLIMbus data channels. Port 2 and Port 3 are **Sink** ports: they sink data from the SLIMbus data channels. It is important to know if a port is a Source or a Sink port because it will decide on the kind of message we must transmit to the port to configure it.

We can also see in this example that all 4 data ports are attached to the Generic Device. In such a case, we will need to send messages to that Generic device to activate data streaming. The Interface Device will not play any role in setting up the data stream.

The SLIMbus Audio Bridge has 8 data ports attached to its Generic devices. The data ports have a particular implementation: they can be programmed as Source port (transmitter) or as Sink ports (receiver).



If we want to use the SPDIF input, we see that we must use port 0 and port 1 (SDIN1), set as source. The SPDIF output can actually use any of the 8 ports as the signal selection is done outside the SLIMbus domain.

Once we know which data ports has to be used and to which device it belongs, we are ready to proceed with the bus configuration.

3.3. Bus Configuration

The SLIMbus must be configured in such a way that it can host the desired data channels. This is a big topic and most of the complexity of the procedure will reside in this step.

The bus configuration is given by three parameters:

- The Root Frequency (**RF**)
- The Clock Gear (**CG**)
- The Subframe Mode (**SM**)

The Root Frequency and the Clock Gear parameters will actually define the bus clock and therefore the available bandwidth. They will also dictate the Transport Protocol to be used in the data channel.

The Subframe Mode parameter will specify the amount of bandwidth available to carry audio streams. Always make sure that the Subframe Mode frees enough space and structure the bus to optimally host the desired data channels.

The Root Frequency, the Clock Gear, the Subframe Mode, the channel rate and the sampling rate are all interacting. To properly set a data stream, the five parameters must be considered all together.

The bus configuration can be adapted to our needs by using the proper reconfiguration messages.

3.3.1. Bus Frequency

The bus frequency is given by the following formula: $F = RF / 2^{(10-CG)}$

The Root Frequency (RF) parameter points to a list of defined frequencies.

RF value	1	2	3	4	5	6	7	8	9
Frequency (MHz)	24.576	22.5792	15.36	16.8	19.2	24	25	26	27

Root Frequency 1 and 2 are the cardinal frequencies commonly used in digital audio.

24.576 MHz is associated to 48 kHz, 96 kHz and 192 kHz audio streaming.

22.5792 MHz is associated to 22.05 kHz, 44.1 kHz, 88.2 kHz audio streaming.

19.2 MHz and 26 MHz are common clocks use in the GSM RF base bands.

For instance, if the Root Frequency in use is 1 (24.576 MHz) and the Clock Gear in use is 9, the actual bus clock will be equal to $24.576 / 2^{(10-9)} = 12.288$ MHz. Consequently, the maximum bandwidth of the SLIMbus will be 12.288 Mbps.

A decision tree can help the user to define the optimal bus and channel parameters.

If the bus frequency divided by the sampling frequency (or sampling rate) is equal to either 2×2^N or 3×2^N with N being an integer, then the channel rate will equal the sampling rate and the transport protocol to be used is “Isochronous”. This is the simplest configuration.

Example 1: The Root Frequency is equal to 1 (24.576 MHz) and the Clock Gear is equal to 9. Then, the actual bus clock is 12.288 MHz. The sampling rate of interest is 48 kHz. $12.288 \text{ MHz} / 48 \text{ kHz} = 256 = 2 \times 2^7$.

Example 2: The Root Frequency is equal to 1 (24.576 MHz) and the Clock Gear is equal to 8. Then, the actual bus clock is 6.144 MHz. The sampling rate of interest is 32 kHz. $6.144 \text{ MHz} / 32 \text{ kHz} = 192 = 3 \times 2^6$.

If the bus frequency divided by the sampling frequency (or sampling rate) is **NOT equal to either 2×2^N or 3×2^N with N being an integer, then the channel rate will never equal the sampling rate. In this case, select a channel rate higher than the sampling rate and choose either the “Pulled” or the “Pushed” protocols.**

Example 3: The Root Frequency is equal to 8 (26 MHz) and the Clock Gear is equal to 9. Then, the actual bus clock is 13 MHz. The sampling rate of interest is 48 kHz. $13 \text{ MHz} / 48 \text{ kHz} = 270.833333$. The closest channel rate values are either:

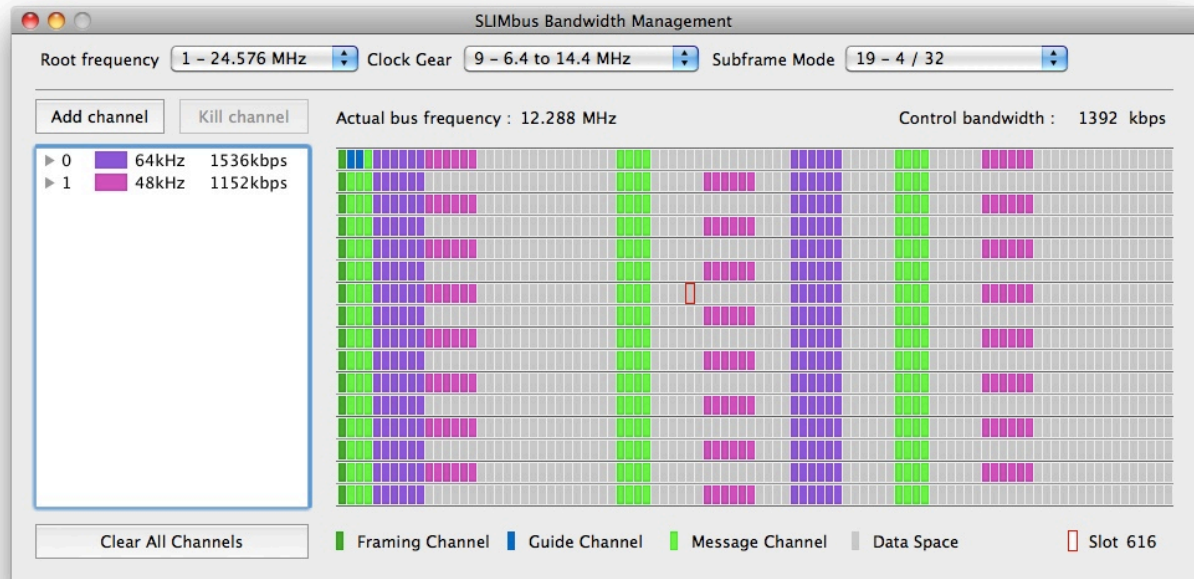
- $13 \text{ MHz} / 256 = 50.781250 \text{ kHz}$
- $13 \text{ MHz} / 192 = 67.750833 \text{ kHz}$

Both values are greater than 48 kHz and can be used with the Pushed or the Pulled protocols.

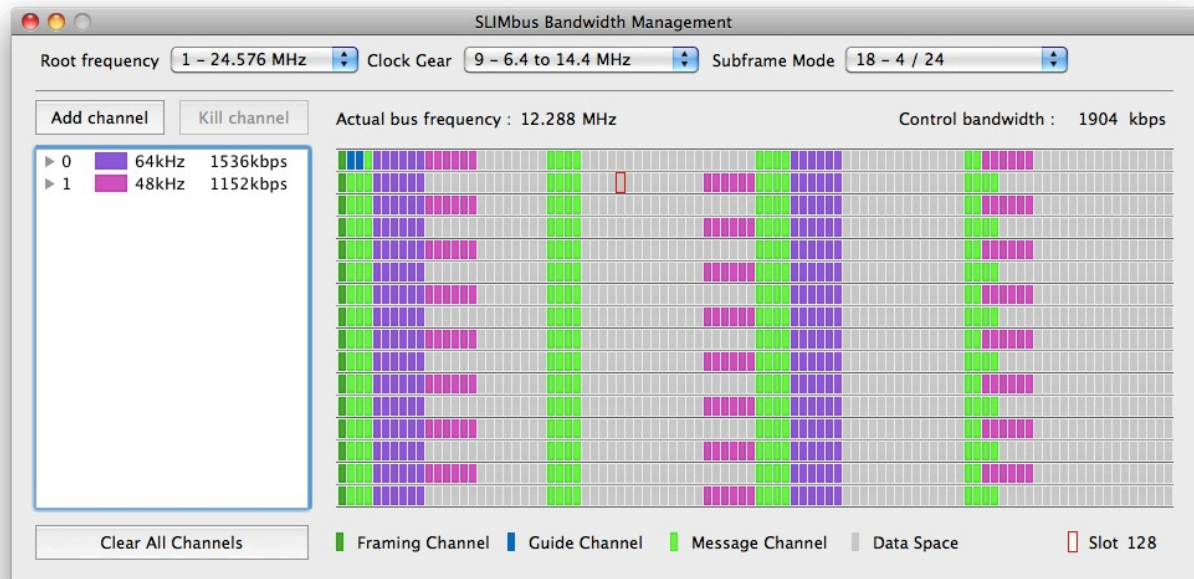
3.3.2. Subframe Mode

The subframe mode and the sampling rate are also interdependent. The subframe lengths equal to **6** or **24** slots are more suited to the channel rates that are equal to the bus clock divided by 3×2^N (32 kHz family) while the subframe lengths equal to **8** or **32** slots are more suited to the channel rates that are equal to the bus clock divided by 2×2^N (48 kHz / 44.1kHz families).

The following picture shows two channels; one at 48 kHz (pink) and the other one at 64 kHz (purple). We obviously see that the 64 kHz channel has more segments in a superframe than the 48 kHz channel. We also see that the segment spread in the superframe is very different. However, both channels fit with this subframe length of 32 slots.



The next figure shows the same channel configuration but with a subframe length of 24 slots.



This time, we clearly see (on the right side) that the 48 kHz channel collides with the message channel (green slots). This is obviously forbidden. This proves that the 24 slot long subframes are not suited for the 48 kHz (and associated) sampling rates while the 32 slots long subframes can host with better luck both 32 kHz and 48 kHz sampling rates families.

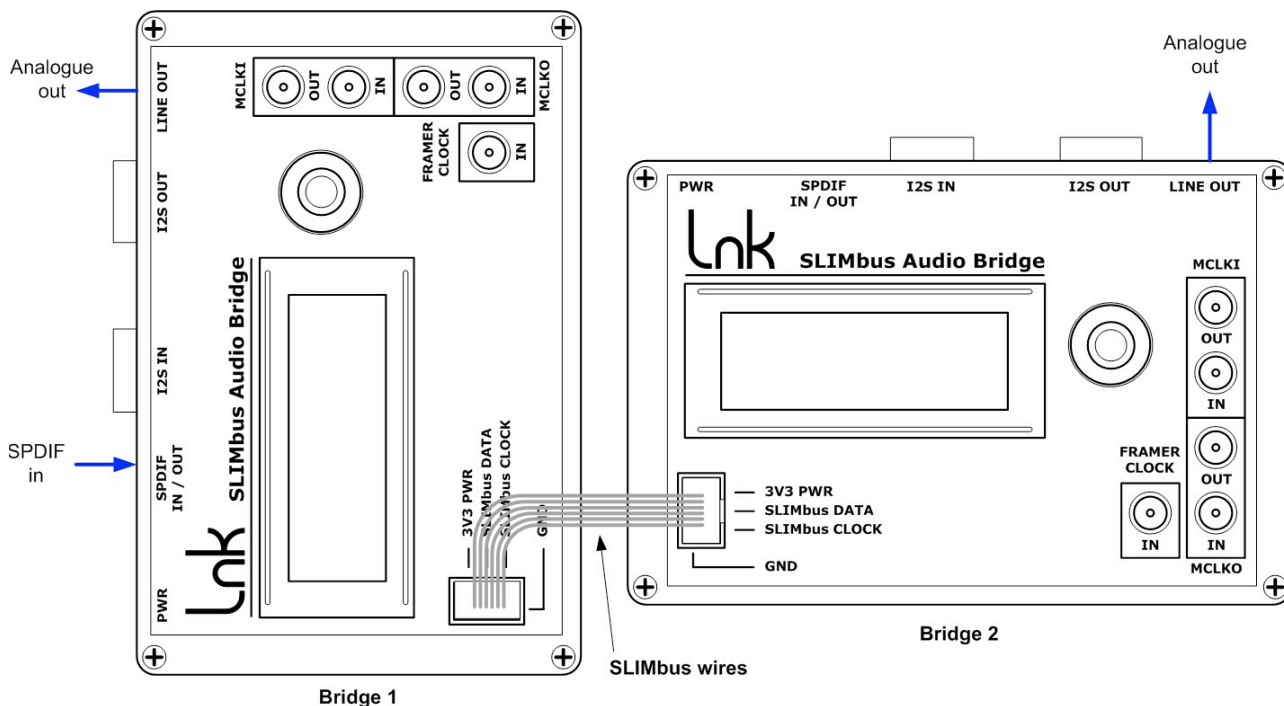
Note: Both the ScriptBuilder and AudioBridge softwares offer tools to easily define the channel configuration parameters.

4. Practical Experimentation

To clarify the theoretical explanations that were just given, we will describe in details how to setup two audio channels between two SLIMbus Audio Bridges. We will follow the steps described earlier to collect or define all the necessary information to setup a channel.

4.1. What do we want to do?

We want to input a SPDIF stream in the first SLIMbus Audio Bridge and output the audio stream on the analogue output of both the first SLIMbus Audio Bridge and the second SLIMbus Audio Bridge.



4.2. Finding the Enumeration Addresses

Bridge 1 will have the address (or Instance Value IV) 0 and Bridge 2 will receive address (or Instance Value IV) 1. Therefore, the Enumeration addresses of all the devices appearing on the SLIMbus will be:

	Device	MID	PID	DI	IV	EA
Bridge 1	Interface	0x01C1	0x0001	0x00	0x00	0x01C100010000
	Framer			0x01		0x01C100010100
	Generic			0x02		0x01C100010200
Bridge 2	Interface	0x01C1	0x0001	0x00	0x01	0x01C100010001
	Framer			0x01		0x01C100010101
	Generic			0x02		0x01C100010201

We can verify that by booting the SLIMbus and looking at the messages that are transmitted by all the devices. The message capture and other SLIMbus screenshots are coming from the AudioBridge software.

Live View

ID	R	Src	Dst	Message
0	P	0x01C100010201	0xFF	REPORT_PRESENT (DC=0x00, DCV=0x01)
1	P	0x01C100010200	0xFF	REPORT_PRESENT (DC=0x00, DCV=0x01)
2	P	0x01C100010101	0xFF	REPORT_PRESENT (DC=0xFE, DCV=0x01)
3	P	0x01C100010100	0xFF	REPORT_PRESENT (DC=0xFE, DCV=0x01)
4	P	0x01C100010001	0xFF	REPORT_PRESENT (DC=0xFD, DCV=0x01)
5	P	0x01C100010000	0xFF	REPORT_PRESENT (DC=0xFD, DCV=0x01)

We can see 6 logical devices sending a REPORT_PRESENT message. If we open the Component Map window, we will see 2 similar components with distinct Instance Values.

Component Map

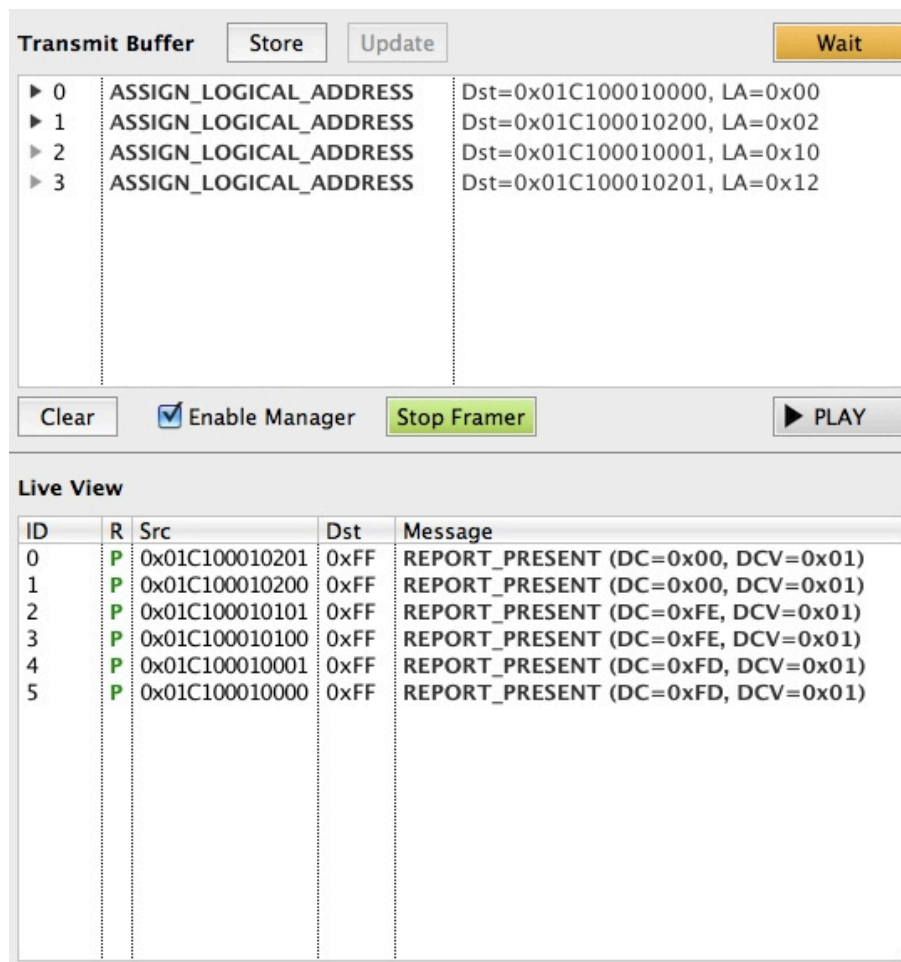
▼ 0	Component 0	MID=0x01C1	PID=0x0001	IV=0x01	
	Generic	LA=	DI=0x02	DC=0x00	EA=0x01C100010201
	Framer	LA=	DI=0x01	DC=0xFE	EA=0x01C100010101
	Interface	LA=	DI=0x00	DC=0xFD	EA=0x01C100010001
▼ 1	Component 1	MID=0x01C1	PID=0x0001	IV=0x00	
	Generic	LA=	DI=0x02	DC=0x00	EA=0x01C100010200
	Framer	LA=	DI=0x01	DC=0xFE	EA=0x01C100010100
	Interface	LA=	DI=0x00	DC=0xFD	EA=0x01C100010000

We also see that these logical devices are waiting to receive a Logical Address. The Logical Address must be seen as a nickname used to communicate with the devices. Using 48 bits addressing in every message would not have been efficient. Therefore SLIMbus uses 8 bits Logical Addresses to communicate with the Logical Devices. The Logical Addresses must be assigned by sending the ASSIGN_LOGICAL_ADDRESS message.

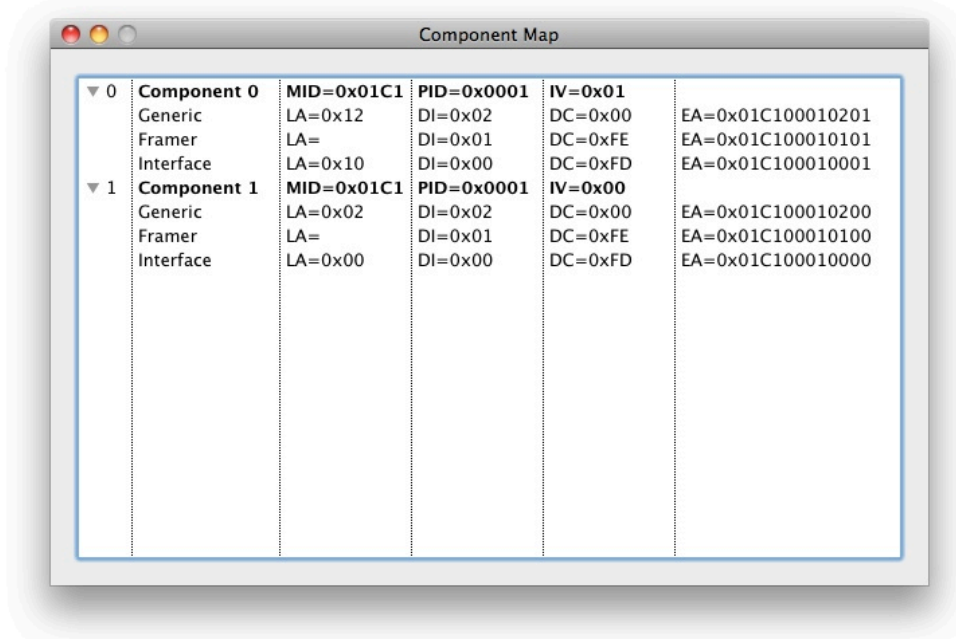
Note that it is not mandatory to give a Logical Address (**LA**) to all the device present on the SLIMbus. It is good practice to assign a **LA** to the Interface Device as this device is the one that will tell you if an error condition has been encountered.

The devices of interest for our exercise are the Generic Devices. They will participate to the audio streams.

Therefore, in order to proceed with our channel setup, we must first assign Logical Addresses to the two Generic Devices and to the two Interface Devices.



Once played, we can verify the effect of the messages on the Component Map window. See the **LA** values that got now assigned a value.

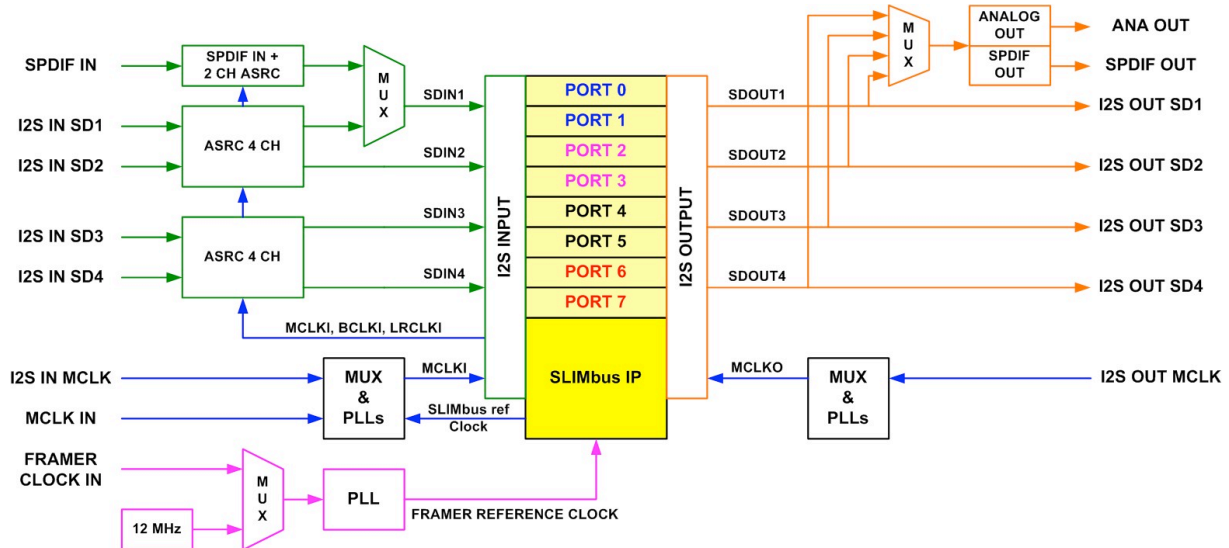


Bridge 1 correspond to **Component 1** and Bridge 2 corresponds to **Component 0**. We can see that we will now use Logical Address 0x02 to talk to the Generic Device of the Bridge 1 and Logical Address 0x12 to talk to the Generic Device of the Bridge 2.

Note: the value of the Logical Address is totally arbitrary. It was our own choice to select the value 0x02 and 0x12. Any other value could be used as long as this value uniquely identifies a logical device.

4.3. Finding the right data ports

Let's review again our SLIMbus Audio Bridge synoptic.



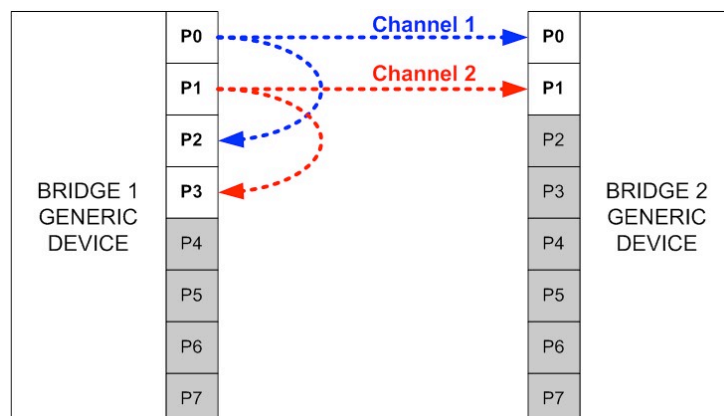
We see that, in order to use the SPDIF input, we need to use Port 0 and Port 1. For the analogue output, all ports are usable, as long as the analogue output is configured to use the selected ports. However, on the Bridge 1, as port 0 and port 1 will already be used for the SPDIF input, we are left with port 2 to port 7 for the analogue output.

The port selection will be as follow:

- Bridge 1, Port 0 as Source to feed the data channel with SPDIF channel B.
- Bridge 1, Port 1 as Source to feed the data channel with SPDIF channel A.
- Bridge 1, Port 2 as Sink to get from the data channel the content of SPDIF channel B.
- Bridge 1, Port 3 as Sink to get from the data channel the content of SPDIF channel A.
- Bridge 2, Port 0 as Sink to get from the data channel the content of SPDIF channel B.
- Bridge 2, Port 1 as Sink to get from the data channel the content of SPDIF channel A.

The selection of the port function is achieved by the following messages:

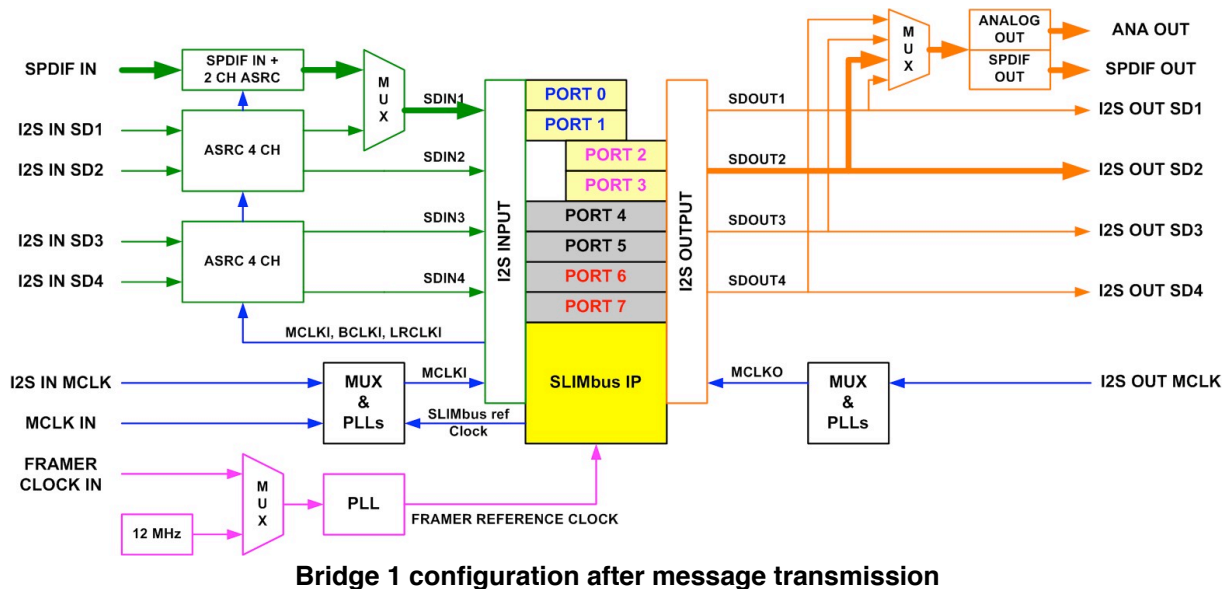
- CONNECT_SOURCE (PN, CN) to tell a port PN that it will be Source of channel CN
- CONNECT_SINK (PN, CN) to tell a port PN that it will be Sink of channel CN



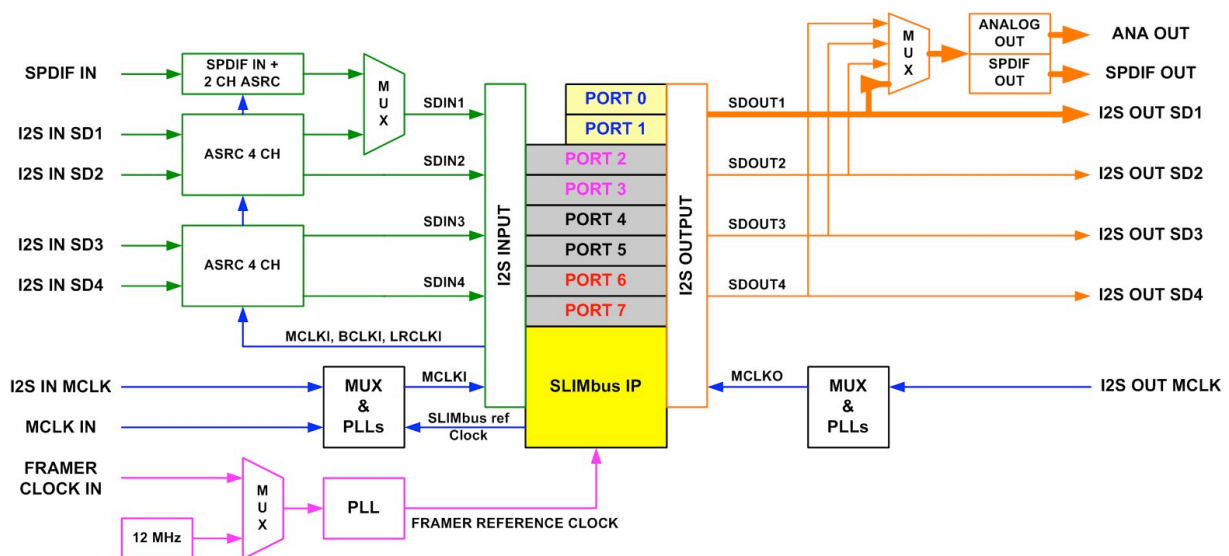
We will connect Bridge 1 / Port 0, Bridge 1 / Port 2 and Bridge 2 / Port 0 to Channel 1.
We will connect Bridge 1 / Port 1, Bridge 1 / Port 3 and Bridge 2 / Port 1 to Channel 2.

Transmit Buffer			Store	Update	Wait
► 0	CONNECT_SOURCE	Src=0xFF, Dst=0x02, CN=1, PN=0			
► 1	CONNECT_SOURCE	Src=0xFF, Dst=0x02, CN=2, PN=1			
► 2	CONNECT_SINK	Src=0xFF, Dst=0x02, CN=1, PN=2			
► 3	CONNECT_SINK	Src=0xFF, Dst=0x02, CN=2, PN=3			
► 4	CONNECT_SINK	Src=0xFF, Dst=0x12, CN=1, PN=0			
► 5	CONNECT_SINK	Src=0xFF, Dst=0x12, CN=2, PN=1			

In the message parameters, the Logical Address of the device is specified by “Dst”. “CN” is the channel number and “PN” is the port number. Note that the channel number can take any value from 0 to 255. It was our own (logical?) choice to select 1 and 2 as channel identifiers.



Bridge 1 configuration after message transmission



Bridge 2 configuration after message transmission

4.4. Configuring the bus

In order to make it simple, we will work with the Root Frequency 1 (24.576 MHz). As we only have two 48 kHz audio channels to fit in the bandwidth, we can use any Clock Gear between 7 and 10. Let's use Clock Gear 8.

The SLIMbus Audio Bridge 1 is configured by default to boot the bus at Root Frequency 1 (24.576 MHz). But it also uses by default Clock Gear 9 and Subframe Mode 0.

Subframe Mode 0 means that 100% of the bandwidth is allocated to the control space. There is absolutely no space for a data channel. We need to free some space by changing the subframe mode.

We saw that 48 kHz channels like subframe mode length of 32 slots. So we will select that value. We want to have some bandwidth for the control, still. So we will choose Subframe Mode 19 that corresponds to 32 slots long subframes with 4 slots allocated to the control space.

We will transmit a reconfiguration sequence to change at the same time the Clock Gear and the Subframe Mode.

The screenshot shows a 'Transmit Buffer' window with three buttons: 'Store', 'Update', and 'Wait'. The buffer contains a sequence of four messages:

Index	Message	Parameters
0	BEGIN_RECONFIGURATION	
1	- NEXT_CLOCK_GEAR	CG=8
2	- NEXT_SUBFRAME_MODE	SM=19
3	RECONFIGURE_NOW	

After playing these messages on the bus, it will run at 6.144 MHz and use 1/8 of the bandwidth for control (a bit less than 768 kbps). We have put the bus in the desired configuration. It is now time to create and activate the data channels.

4.5. Creating the Data Channels

The data channel setup always follows the same scheme. It is a reconfiguration sequence with the following messages:

```

BEGIN_RECONFIGURATION
NEXT_DEFINE_CHANNEL
NEXT_DEFINE_CONTENT
NEXT_ACTIVATE_CHANNEL
RECONFIGURE_NOW

```

The parameters of these messages will actually make a channel unique and according to our specifications.

4.5.1. NEXT_DEFINE_CHANNEL

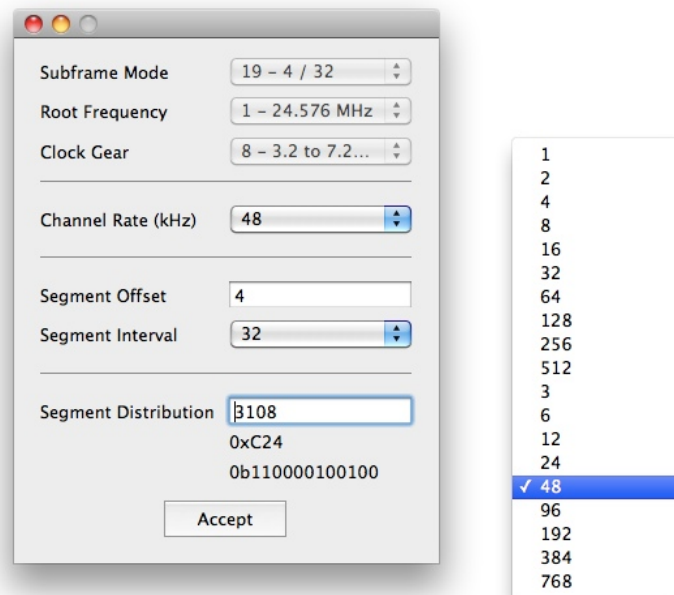
This message defines the structure of the channel. It specifies how the segments are organized in the superframe, the size of the segments and the transport protocol to be used.

For the transport protocol, it is pretty simple. We will use a 48 kHz channel to carry a 48 kHz sampling rate. Therefore, we can use the Iscochronous protocol: **TP=0**.

The size of the segment will directly define the sample size. If we want to carry 16 bits audio samples, we will also use a segment of 16 bits (4 slots). So **SL=4**.

Now remains the Segment Distribution (**SD**) that defines the segment organization in the superframe. This value has a special, non intuitive coding that is explained in detail in the SLIMbus Specification document.

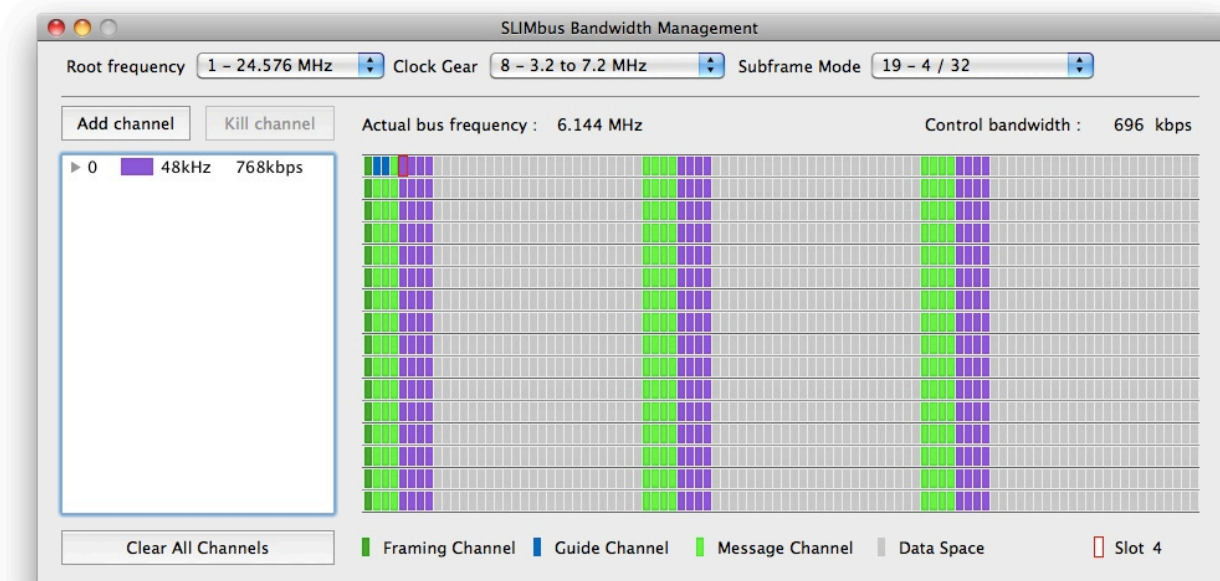
As we want to make it easy and error free, we will use the tools provided with the AudioBridge software to compute the value of the Segment Distribution.



We just need to select the desired channel rate in the drop box and the Segment Distribution will be computed automatically for us. The message will look like:

NEXT_DEFINE_CHANNEL (CN=1, SD=3108, TP=0, SL=4)

The channel will have the following segment organization.



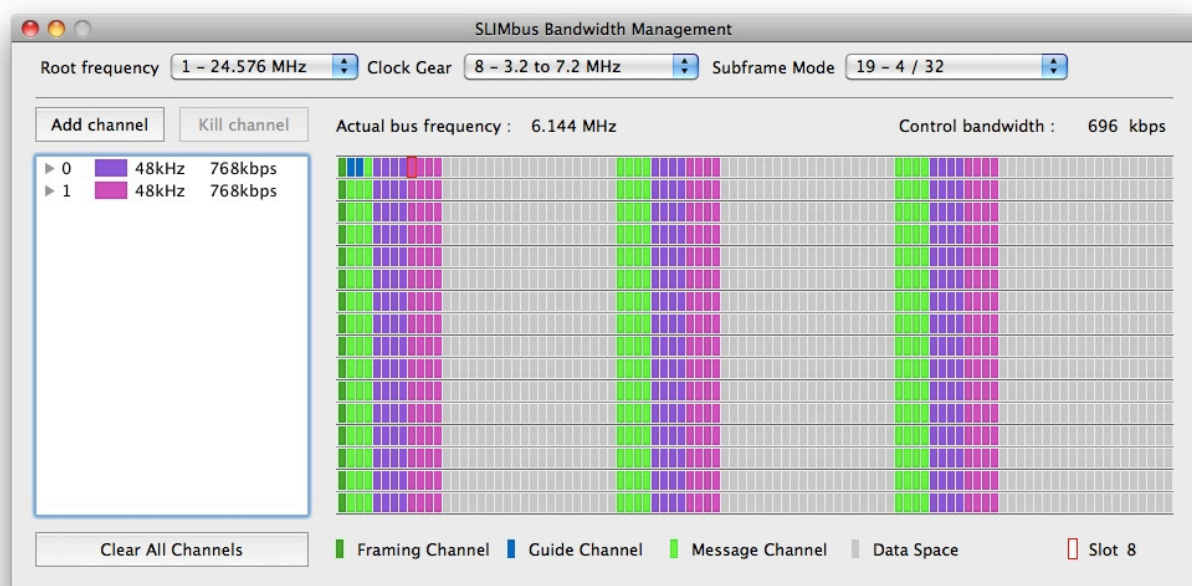
The second channel we need to create must not conflict with the first one. The Segment Length and Transport Protocol values will be the same as for the first channel but the Segment Distribution Value will be a bit different.

The values 3108 was obtained by using a Segment Offset equal to 4. That means that the first segment of the channel in a superframe will appear in slot 4. We can verify it on the previous figure.

In order to avoid any overlap, the second channel must have an offset greater or equal to 8 (offset of the first channel + segment length of the first channel). There is no benefit in letting empty slots between channels. It would just be a waste of bandwidth. Therefore, we will use an offset equal to 8 for the second channel.

NEXT_DEFINE_CHANNEL (CN=2, SD=3112, TP=0, SL=4)

The channels will have the following segment organization.



4.5.2. NEXT_DEFINE_CONTENT

The parameters of that message are not playing any role on the channel management. However, devices may use these values to figure out what is transported by the channel. That's the case of the SLIMbus Audio Bridge. So we would better fill these parameters with the right value.

- **Frequency Locked (FL)** indicates if the sampling rate and the channel rate are phased locked. When using the Isochronous Protocol, it is always the case, so let's make **FL=1**.
- **Presence Rate (PR)** is the sampling rate. SLIMbus codes the various existing sampling rates in a look-up table (see Specification). 48 kHz corresponds to **PR=3**. When defining the parameter values with the AudioBridge software, use contextual menu to select the desired value. It is the simplest way to proceed.
- **AUX Format (AF)** will be left to 0 by default. **AF=0**.
- **Data Type (DT)** indicates what kind of data is transported. We will use **DT=1**, corresponding to L-PCM format.
- **Channel Link (CL)** will be left to 0 by default. **CL=0**.
- **Data Length (DL)** is the sample size in slots. We use 16 bits so we make **DL=4**.

The message will look like:

NEXT_DEFINE_CONTENT (CN=1, FL=1, PR=3, AF=0, DT=1, CL=0, DL=4)

Note that this content definition is valid for both channels so we can simply do

NEXT_DEFINE_CONTENT (CN=2, FL=1, PR=3, AF=0, DT=1, CL=0, DL=4)

for the second channel.

4.5.3. NEXT_ACTIVATE_CHANNEL

This message activates a channel once its structure and its content have been defined. There are no particular parameters other than the channel identifier.

NEXT_ACTIVATE_CHANNEL (CN=1)

NEXT_ACTIVATE_CHANNEL (CN=2)

4.5.4. The complete sequence

Now, we can put all the messages together to have our reconfiguration sequence for channel 1.

BEGIN_RECONFIGURATION

NEXT_DEFINE_CHANNEL (CN=1, SD=3108, TP=0, SL=4)

NEXT_DEFINE_CONTENT (CN=1, FL=1, PR=3, AF=0, DT=1, CL=0, DL=4)

NEXT_ACTIVATE_CHANNEL (CN=1)

RECONFIGURE_NOW

The sequence for channel 2 will be very similar.

BEGIN_RECONFIGURATION

NEXT_DEFINE_CHANNEL (CN=2, SD=3112, TP=0, SL=4)

NEXT_DEFINE_CONTENT (CN=2, FL=1, PR=3, AF=0, DT=1, CL=0, DL=4)

NEXT_ACTIVATE_CHANNEL (CN=2)

RECONFIGURE_NOW

Note that SLIMbus allows you to merge the two sequences into one. It has the benefits of creating the 2 channels at the exact same moment while with 2 sequences, the channels are created sequentially.

BEGIN_RECONFIGURATION

NEXT_DEFINE_CHANNEL (CN=1, SD=3108, TP=0, SL=4)

NEXT_DEFINE_CONTENT (CN=1, FL=1, PR=3, AF=0, DT=1, CL=0, DL=4)

NEXT_ACTIVATE_CHANNEL (CN=1)

NEXT_DEFINE_CHANNEL (CN=2, SD=3112, TP=0, SL=4)

NEXT_DEFINE_CONTENT (CN=2, FL=1, PR=3, AF=0, DT=1, CL=0, DL=4)

NEXT_ACTIVATE_CHANNEL (CN=2)

RECONFIGURE_NOW

4.6. Script review

Let's come back on the complete list of messages involved in setting up the two data channels. The following figure shows all the message traffic from the moment we booted the SLIMbus till the activation of the 2 audio channels.

	Live View				
	ID	R	Src	Dst	Message
Messages transmitted by the devices when boot was completed	0	P	0x01C100010201	0xFF	REPORT_PRESENT (DC=0x00, DCV=0x01)
	1	P	0x01C100010200	0xFF	REPORT_PRESENT (DC=0x00, DCV=0x01)
	2	P	0x01C100010101	0xFF	REPORT_PRESENT (DC=0xFE, DCV=0x01)
	3	P	0x01C100010100	0xFF	REPORT_PRESENT (DC=0xFE, DCV=0x01)
	4	P	0x01C100010001	0xFF	REPORT_PRESENT (DC=0xFD, DCV=0x01)
	5	P	0x01C100010000	0xFF	REPORT_PRESENT (DC=0xFD, DCV=0x01)
Logical Address assignment	6	P	0xFF	0x01C100010000	ASSIGN_LOGICAL_ADDRESS (LA=0x00)
	7	P	0xFF	0x01C100010200	ASSIGN_LOGICAL_ADDRESS (LA=0x02)
	8	P	0xFF	0x01C100010001	ASSIGN_LOGICAL_ADDRESS (LA=0x10)
	9	P	0xFF	0x01C100010201	ASSIGN_LOGICAL_ADDRESS (LA=0x12)
Data Port configuration	10	P	0xFF	0x02	CONNECT_SOURCE (CN=1, PN=0)
	11	P	0xFF	0x02	CONNECT_SOURCE (CN=2, PN=1)
	12	P	0xFF	0x02	CONNECT_SINK (CN=1, PN=2)
	13	P	0xFF	0x02	CONNECT_SINK (CN=2, PN=3)
	14	P	0xFF	0x12	CONNECT_SINK (CN=1, PN=0)
	15	P	0xFF	0x12	CONNECT_SINK (CN=2, PN=1)
Bus configuration	16	P	0xFF	All	BEGIN_RECONFIGURATION ()
	17	P	0xFF	All	- NEXT_CLOCK_GEAR (CG=8)
	18	P	0xFF	All	- NEXT_SUBFRAME_MODE (SM=19)
	19	P	0xFF	All	RECONFIGURE_NOW ()
Channel 1 setup	20	P	0xFF	All	BEGIN_RECONFIGURATION ()
	21	P	0xFF	All	- NEXT_DEFINE_CHANNEL (CN=1, SD=3108, TP=0, SL=4)
	22	P	0xFF	All	- NEXT_DEFINE_CONTENT (CN=1, FL=1, PR=3, AF=0, DT=1, CL=0, DL=4)
	23	P	0xFF	All	- NEXT_ACTIVATE_CHANNEL (CN=1)
	24	P	0xFF	All	RECONFIGURE_NOW ()
Channel 2 setup	25	P	0xFF	All	BEGIN_RECONFIGURATION ()
	26	P	0xFF	All	- NEXT_DEFINE_CHANNEL (CN=2, SD=3112, TP=0, SL=4)
	27	P	0xFF	All	- NEXT_DEFINE_CONTENT (CN=2, FL=1, PR=3, AF=0, DT=1, CL=0, DL=4)
	28	P	0xFF	All	- NEXT_ACTIVATE_CHANNEL (CN=2)
	29	P	0xFF	All	RECONFIGURE_NOW ()

Note: To get the audio streams routed to the analogue output of the SLIMbus Audio Bridges, we shall manually set the DAC Input Stream to **SDO2** for the first Bridge and to **SDO1** for the second Bridge.