Google Spatial Audio

Rig HW3.0 User Guide

Release version: 0.0.3

Date: 09/April/2024

Author:	Dobromir Denchev
Status:	Approved
Approver:	Iva Echeva

Contents

1 Spatial Audio Latency Measurements

1.1 Motivation

With the variety of different vendors, phones and earbuds on the market and together with the newly added Android Spatial Audio (**SA**) features with Head Tracking functionality, there is a need to measure and compare end-to-end Head Tracking Latency (**HTL**) of the different implementations.

The test setup has to:

- Allow end-to-end measurement of commercial samples, without need of disassembly or additional test wires/signals.
- Emulate human head wearing in-ear earbuds and option to rotate between predefined positions in horizontal plane, with selectable speed.
- Be easily reproducible on different Android based devices.
- Give an option for measure automation – to minimize human's subjective factor in the measurements.
- Give an option for measure automation – to average results for long sequences with different settings and thus to minimize measurement errors.

2 How To Use

2.1 Test setup components description:

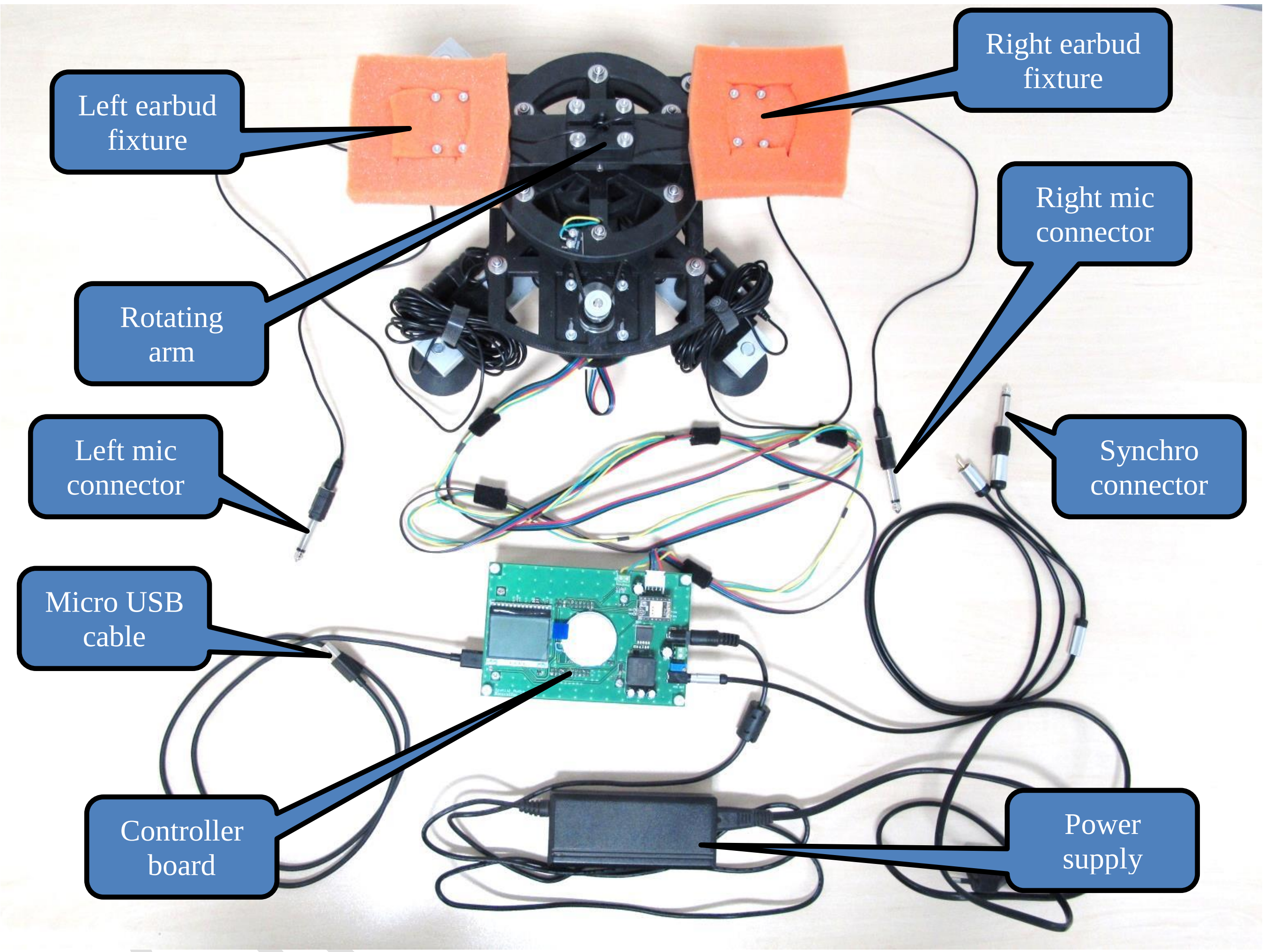


Figure 1

This test rig (*Figure 1*) is equipped with highly reliable, 40W desktop-style single output power supply with 3-pin (with FG) standard IEC320-C14 replaceable power plug. Please use power cord with the proper plug for your power outlet. The input voltage must be in the range 80-264 VAC. This power supply is equipped with green led indicator light that lights-on while it is plugged in the power network.

The test rig is built over solid aluminium base that firmly secures all mechanical parts and gives enough weight and stability to the whole test platform. There are adjustable rubber legs at each corner of the base. All legs have to be properly adjusted for the particular work place to stabilize the test platform in horizontal plane.

Controller board is connected to the stepper motor with long cables that allows flexible arrangement of the test setup. The intention is to keep the rotating parts and microphones far away from noise sources on the desk as for example, PC Power supply fan.

On the right side of the controller board there is a 3.5mm stereo audio connector that exports the motion synchronization signal to the recording interface. Please note that by default only the Left channel (the tip of the jack) is routed to output the motion signal. This is intentionally done - to avoid signal shortage in case using mono jacks at the other side of the connecting cable. Next to the audio connector there is a small trimmer potentiometer that can adjust the motion synchronization signal amplitude (only if needed).

Tested earbuds have to be positioned in the Left and Right Fixtures. Attention have to be put to not mistake left and right earbuds as this may impact the tests.

The test rig is equipped also with two BOYA-M1 microphones with preamplifiers that are transferring the earbuds audio to the recording audio interface. To avoid additional noises generated by the current pulses through the stepper motor, the pre-amplifiers are powered by single LR44 cell-battery (one in each preamplifier). Please note that there are Power ON/OFF switches at each preamplifier, which have to be set to ON (Cameras) position prior testing and switched to OFF after that – to preserve battery power. Both preamplifiers are attached to the base with velcro stripes (*Figure 2*). This allows easy battery replacement and option to store the microphone cables there as well. For battery replacement instructions, please refer to the microphones user manual documentation (https://www.boya-mic.com/product/by-m1).

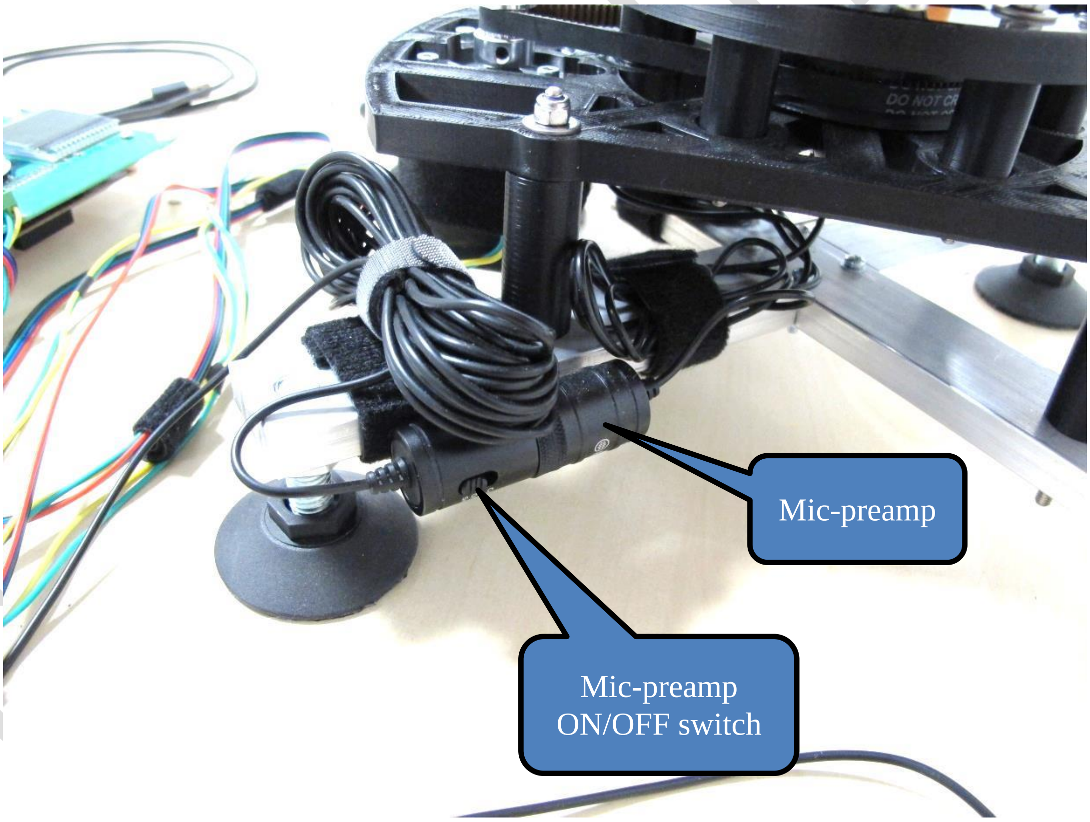


Figure 2

The test rig is controlled through simple user interface with LCD display and big rotary knob (*Figure 3*). By rotating the knob ring can move up/down through menu. Pressing the centre button will enter mode of changing the value for the current menu (again by rotating the ring). To exit the change mode – just press the knob button again.

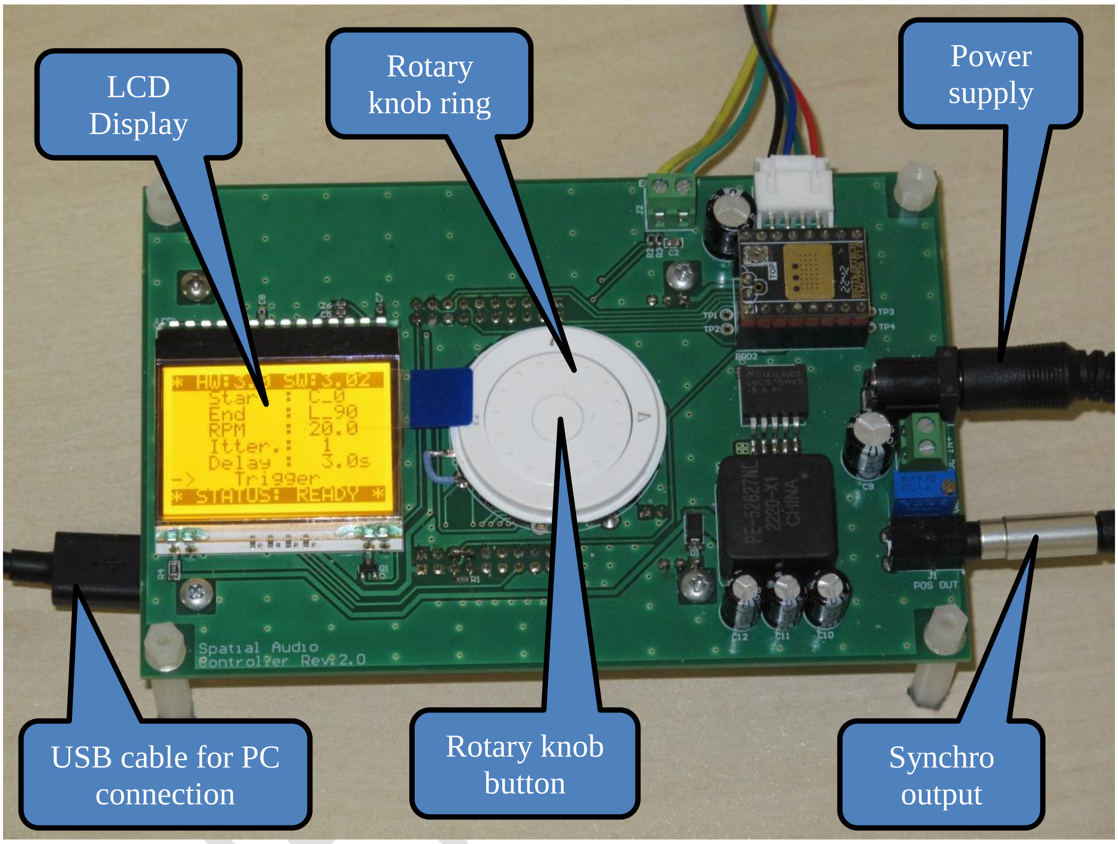


Figure 3

2.2 Earbuds fixture

Prior to latency measurements, the tested earbuds have to be attached to the earbuds fixtures and microphones. For earbuds with rubber seal, this is done by gently push of the rubber earbud seal to envelop inside the sound absorbing foam (*Figure 4*). Please adjust the silicone pipe with the microphone in position that does not touching the earbud. This will minimize motion noises injected directly to the microphone (*Figure 5*). The microphone itself is attached to the other side of the same silicone pipe within the acoustic foam holder and usually do not need any additional adjustments:

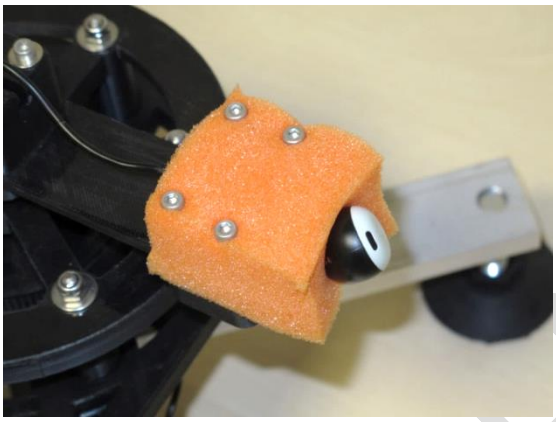


Figure 4

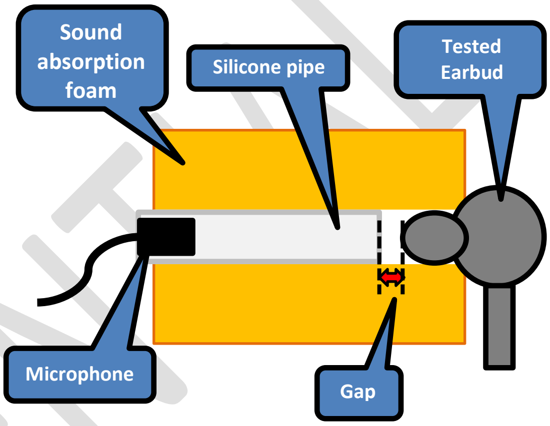


Figure 5

For earbuds without rubber seal that fit in the sound absorbing foam, there is wider silicone tube piece provided, that to act as a junction between the tested earbud and the channel in the sound absorption foam. Again, once the tested earbud is firmly attached to the wider pipe, please push it together with the added silicone pipe to envelop inside the sound absorbing foam and keep in mind the needed gap with the other silicone pipe with the microphone (*Figure 6*).

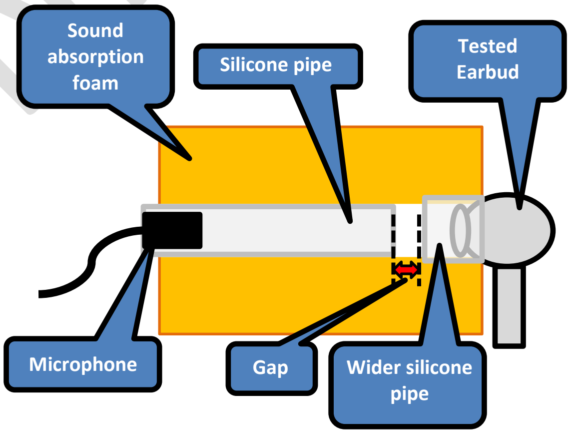


Figure 6

To keep the earbud in position and to avoid vibrations during the motion tests, please put on top of the mounted earbud the additional fixture from audio absorption foam (*Figure 7*). Please mind the space for microphone and cabling in the additional fixture (*Figure 8*).

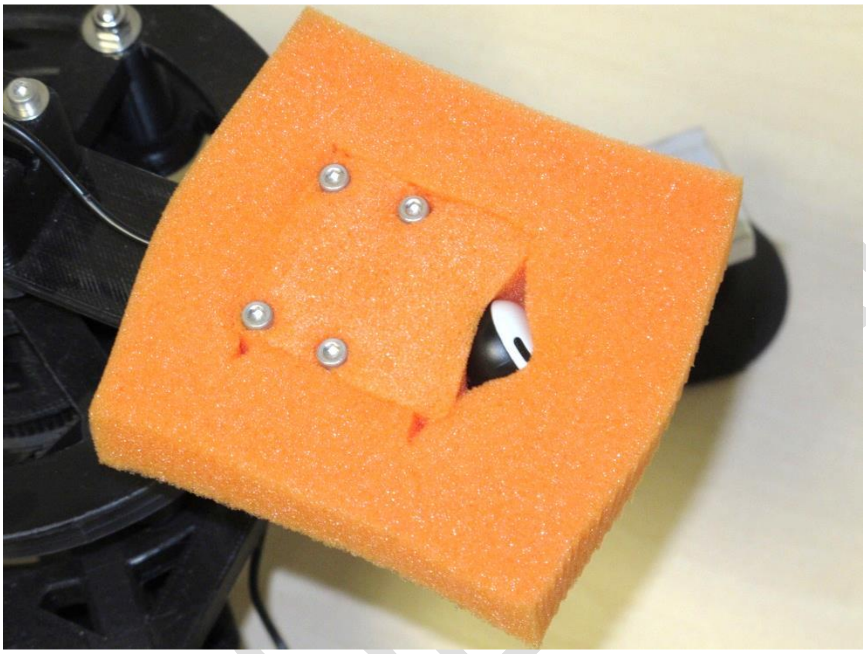


Figure 7

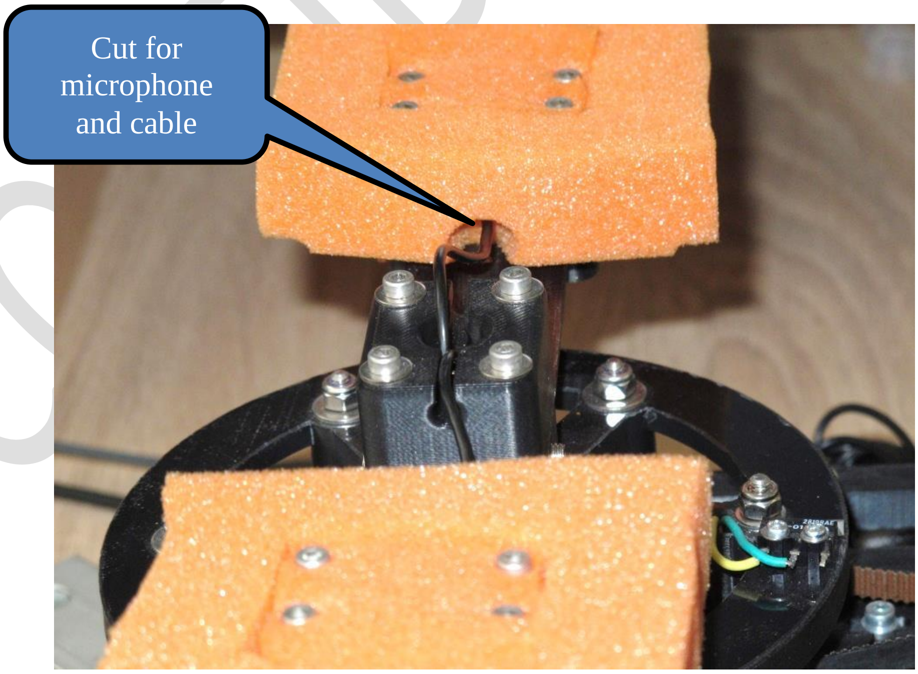


Figure 8

2.3 User Interface

2.3.1 User Menu

After power up, the microcontroller will initialize the rotating arm to initial position and will enter the operational mode of the test rig.

The user interface allows manual execution of various tests scenarios. On top of this there is also PC interface that allows automatic execution of test sequences.

The "User Menu" of the user interface is composed of header, footer and six attribute lines – see *Figure 9*.

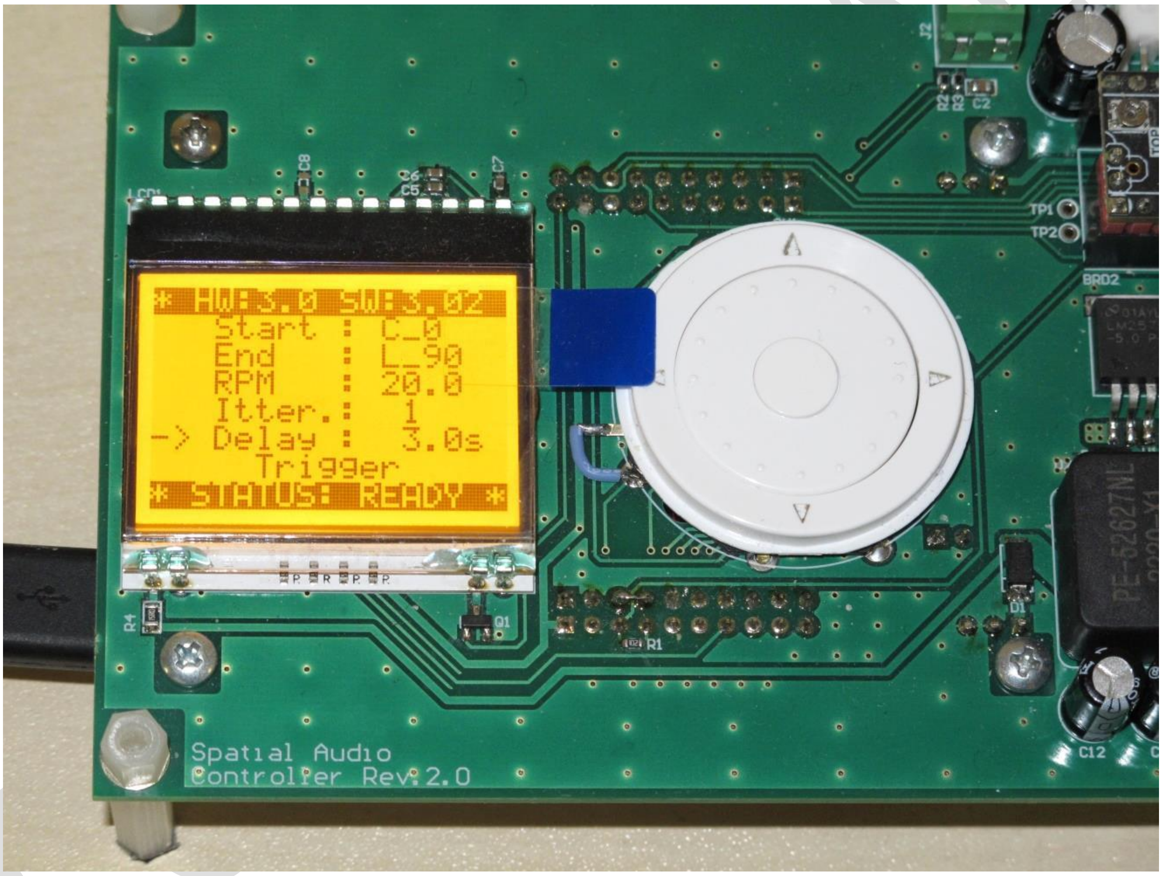


Figure 9

The header line (on the top of the LCD screen) displays the current version of the HW and SW.

The footer line (on the bottom of the LCD screen) displays the current rig status:

- **INIT** is displayed during system boot and initial positioning of rotating arm.
- **READY** means that the rig is ready to accept new commands.
- **RUNING** means that there is motion in progress and no new commands can be accepted.

The remaining 6 lines on the LCD screen can be selected by rotation of the rotary knob ring. Selected line is marked with "**->**" arrow symbol in the beginning (left side) of the line. Pressing the rotary knob button will "activate" selected item and will allow modification of the current value by rotation of the rotary knob. To exit this edit mode, just press the rotary knob button again.

- **Start** Defines the start motion position of rotating ARM.
  - There is selectable list of predefined Horizontal positions with step of 15 degrees:
    - L90 – 90 degrees to Left
    - L75 – 75 degrees to Left
    - L60 – 60 degrees to Left
    - L45 – 45 degrees to Left
    - L30 – 30 degrees to Left
    - L15 – 15 degrees to Left
    - C_0 – Central position
    - R15 – 15 degrees to Right
    - R30 – 30 degrees to Right
    - R45 – 45 degrees to Right
    - R60 – 60 degrees to Right
    - R75 – 75 degrees to Right
    - R90 – 90 degrees to Right
  - By default after power up, the Start is set to the central position C_0.
- **End** Defines the end motion position of rotating ARM.
  - There is selectable list of predefined Horizontal positions with step of 15 degrees:
    - L90 – 90 degrees to Left
    - L75 – 75 degrees to Left
    - L60 – 60 degrees to Left
    - L45 – 45 degrees to Left
    - L30 – 30 degrees to Left
    - L15 – 15 degrees to Left
    - C_0 – Central position
    - R15 – 15 degrees to Right

  - R30 – 30 degrees to Right
  - R45 – 45 degrees to Right
  - R60 – 60 degrees to Right
  - R75 – 75 degrees to Right
  - R90 – 90 degrees to Right
- By default after power up, the End is set to the most left position L90.

- **RPM** Defines the revolutions per minute of rotating ARM.
  - The speed of rotation can be adjusted in the interval [0.1-50] rpm, with the step of 0.1 rpm.
  - By default after power up, the RPM is set to 25 rpm.
- **Itter.** Defines the number of iterations tat to be repeated with single trigger.
  - The number of iterations can be adjusted in the interval [1-10], with the step of 1 iteration.
  - By default after power up, the Itter. is set to 1.
- **Delay** Defines the delay interval between every two iterations defined by **Itter.**
  - The delay time can be adjusted in the interval [0.5-10] seconds, with the step of 0.1s.
  - By default after power up, the Delay is set to 3.0s.
- **Trigger** There are no modifiable parameters here.
  - Activating this menu will trigger the last displayed on the screen motion parameters.
  - At the end of the motion, the Start and End positions will be automatically swapped on the screen. This allows multiple sequential runs between selected two positions, without any additional interaction with the user interface.
  - This is the default selected menu after power up.

2.4 Audio Recording Setup

For recording a multitrack audio data, you have to use audio interface device capable of recording of multiple track at once at the chosen PC platform.

It is highly recommended to use audio recording interface that supports real 32bit Float recording capabilities. This will remove the need to setup the audio levels and will minimize the possible problems because of incorrect audio levels setup.

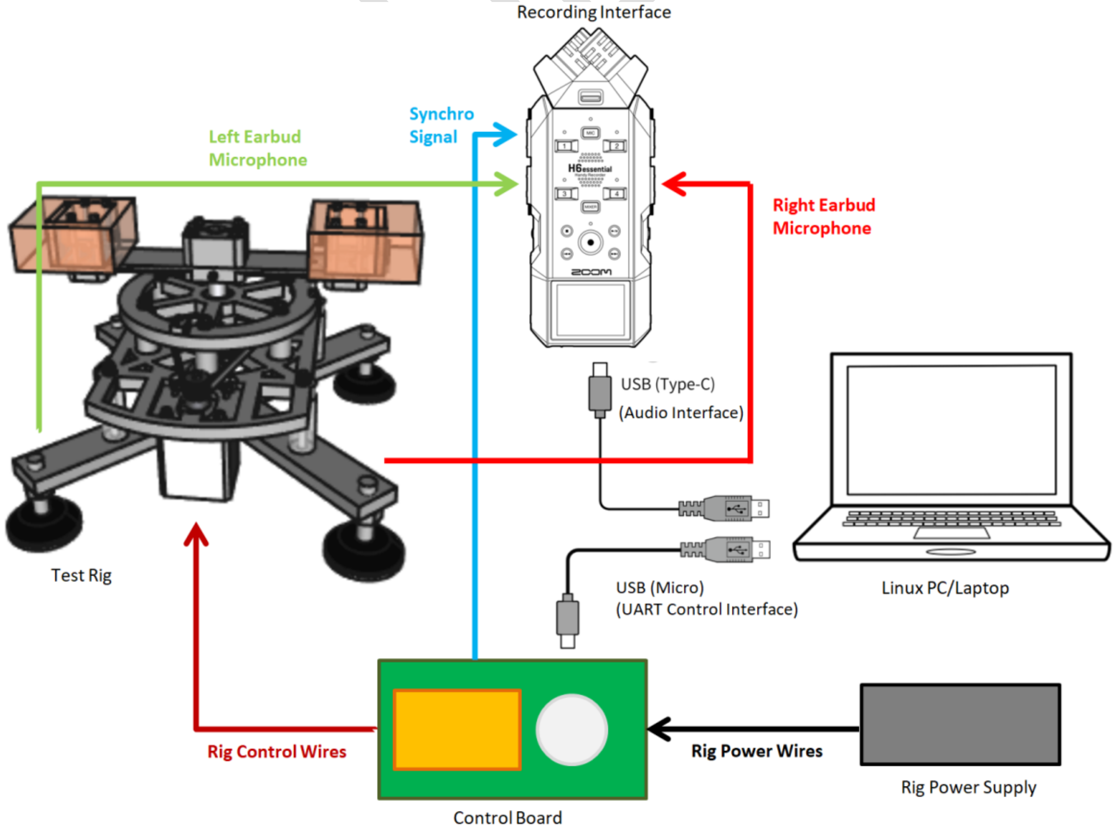
A proven solution is to use Zoom H6essentials device (https://zoomcorp.com/en/us/handheld-recorders/handheld-recorders/h6essential/) as multichannel PC audio interface that supports Float32 recording.

For manual testing at PC side, it was proven to use Audacity application as multichannel recording and measurement program (https://www.audacityteam.org/). Please note that trying to select less than 6 channels for recording leads to unstable behavior, so **for manual tests we were always using 6 channels recording with only 3 useful tracks in the records**.

NOTE: It seems the Audacity program scans for attached devices only at startup, so Zoom H6essential device must be plugged to the PC/Laptop and operational, before starting Audacity program.

Please do not forget to turn ON the microphones preamplifiers before doing latency tests and to switch it OFF once measurements are done – to not deplete the batteries.

Please put attention to keep the audio playback in the tested earbuds in non-saturated amplitude range. As a rule of thumb, **please set audio playback to 75% of the maximum phone volume**.



2.4.1 Audio Setup HW requirements

- HTL measurement rig HW version 3.0 and FW version 3.02 (or newer)
  - 3.5mm to 6.3mm TRS audio cable (Synchro audio signal; included in rig kit)
  - 2x BOYA M1 mics with 6.3mm TRS audio cables (L/R earbud audio; included in rig kit)
  - Power Supply Unit (included in rig kit)
- PC or Laptop with at least 3 USB ports
  - Micro USB cable for connection between PC/Laptop and test rig (rig control interface)
  - USB-C cable between PC/Laptop and ZOOM H6essential (Audio recording interface)
  - USB-C cable between PC/Laptop and tested phone (ADB interface)
- ZOOM H6essential recorder.

2.4.2 Audio Setup SW requirements

- PC or Laptop with Linux Ubuntu 20.04, 64bit
- Automation Latency Measurement Tool release v1.2 (or newer)
- CTS Verifier APP that support "**Immersive Audio Test**"

2.4.3 PC SW environment setup

- The first step is to configure access rights - adding the user to the 'dialout' user group:
  - This have to be done once for each user that will use provided Automation scripts
  - In Linux Ubuntu Terminal please run:

    ```
    ./setup.sh
    ```
  - Log out and log in again, for group adding to take effect
  - You can confirm that user has been added to the 'dialout' group by running *setup.sh* script again and check that it prints:

    ```
    "Found that the user is a member of the "dialout" group.
    OK."
    ```
- Connect all cables and start the tool, using the run.sh script that also collects full logs. For example:

  ```
  ./run.sh Pixel7
  ```
  - In this example we are specifying text name of the device under test - **Pixel7.** This name is used in forming the file names in *results/folder*.

2.5 Latency Measurements

This test rig is designed to produce set of audio signals that allows measurement of the head tracking latencies of 3D spatial audio stream, when there is rotation of the tested earbuds between pre-defined angles in horizontal plane.

For all these latency measurements, the produced audio signals have to be recorded simultaneously to multitrack uncompressed (WAVE) file. Recorded files can be used later for off-line manual or automated analysis of the latencies.

There are 3 audio signals produced by this test rig:

- Motion synchronization signal (**Synchro**)
- Left Earbud microphone signal (**Left**)
- Right Earbud microphone signal (**Right**)

The sound from the tested earbuds is captured by two dedicated microphones with embedded battery powered preamplifiers. Please refer to the *Test setup components description* chapter for more details about battery replacement and on/off switching of the microphones.

Motion signal is common for the both microphones motion synchronization and have to be recorded simultaneously with Left and Right signals.

The “motion signal” is generated locally on the test rig and in this firmware version is set to be 2kHz square wave during no-motion and 10kHz during motion.

- ➢ The moment when the Synchro signal switches **2kHz->10kHz** marks the **start** of the selected earbuds motion
- ➢ The moment when the motion signal switches **10kHz->2kHz** marks the **end** of the selected earbuds motion

The tested earbuds and phone have to be paired and “Spatial Audio” option has to be enabled at the phone side, together with enabled option for “Head Tracking”. Tested earbuds have to be properly installed on the rotation arm holders. Please refer to *Earbuds fixture* chapter for more installation details.

During the Head Tracking Latency measurements, the paired mobile phone have to play predefined 5.1 multichannel audio content – **sine wave with constant amplitude and frequency, only at one channel**. Such reference test audio file will be provided together with the rig documentation release.

2.5.1 Earbuds proximity sensor

Tested earbuds are usually equipped with proximity sensors – to understand when are put in the ears and when are outside the ears.

For these measurements it is crucial to make the tested earbuds to detect that they are inside the ears!

Just disabling in-ear-detection feature in the phone setup is not enough, as head tracking is not working on the earbuds when they are not detecting to be inserted in the ears, so prior HTL measurements need to cover proximity sensors of the tested earbuds with patches of non-transparent, self-adhesive aluminum foil.

Different earbud models are using different types, position and number of proximity sensor(s) and there is no common rule for position and size of the foil patches that have to be applied. **This is model specific and has to be aligned with earbud manufacturer.**

Example for proximity sensors coverage for ***Pixel Buds Pro*** *earbuds*:

2.5.2 Automatic HTL Measurements

To minimize the human factor for manual data parsing, there is an option to use automation scripts and to control the test rig and audio interface remotely from computer/laptop.

Automation tool is running on Linux Ubuntu PC/Laptop and controls the tested Android phone (over ADB), the test rig device (over USB-UART) and managing the processes for recording of test audio data in different conditions and analysis of recorded data – to extract measured HTL numbers.

The **run.sh** bash script is used to calls the HTL test application with or without command line option "--certification", easily chosen interactively by asking question:

```
Are you testing Android with adb connected to the phone device? [y/n]
```

If the user answers "y" (for Yes), the option --certification is added and ADB features are enabled:

The ADB features include:

- Controlling the playback and CtsVerifier app on DUT phone device:
    - upload of test wav file containing 2kHz signal in center channel of 5.1 wav file;
    - start/stop of test signal
- Calling the CtsVerifier app with intent to send the results of the measurement process.
    - NOTE: Correct version of CtsVerifier application must be already installed prior to running the test with "*run.sh*".
- NOTE: Once prompted, please allow ADB connection for USB debugging on the tested phone screen UI.

After execution of the **run.sh** bash script, there will be interactive set-up process for setting USB Audio and Control interfaces and some test parameters. The typical configuration file is automatically generated with some default values that can be changes later in the ***config.txt*** (if needed):

```
cat config/config.txt
-------------------------------
channels 6
samplerate 44100
channel_sync 3
channel_left 5
channel_right 6
port_name /dev/ttyACM1
hw_rev 3
passing_threshold_ms 300
passing_reliability_percent 80
passing_range_ms 40
-------------------------------
```

The following three parameters define the acceptance criteria for the HTL measurement tests:

```
passing_threshold_ms 300
passing_reliability_percent 80
passing_range_ms 40
```

These values configure the requirements for the test to be "Pass" or "Fail" or Retest".

After the test, the measurement results will be processed to calculate "**latency**" in milliseconds and "**reliability**" in percents.

The "**latency**" is the average value of the majority of measurement results that fall within the acceptable variation range specified "passing_range_ms", typically 40ms.

The reliability is the percent of measurement data that reside within this range "passing_range_ms".

- "**Pass**" result means that the "**latency**" is less or equal to "**passing_threshold_ms**" AND "**reliability**" value is greater or equal to "**passing_reliability_percent**" value.
- "**Fail**" result means that the "**latency**" is higher than "**passing_threshold_ms**" value.
- "**Retest**" result means that the "**latency**" is less or equal to "**passing_threshold_ms**", BUT "**reliability**" is less than desired "**passing_reliability**" value.

The automated test procedure sequence is:

- Make sure that the test rig is powered and also the external sound interface of the PC (for example: Zoom H6essential device) is powered and connected to PC in multi-channel recorder mode.
- Connect Micro-USB cable for serial communication
- Configure access rights - add the user to the '*dialout*' user group.
    - For Ubuntu (Linux) please run:

      ./setup.sh
    - and log-out and log-in again, for group adding to take effect.
- Start the tool, using the **run.sh** script, that also collects full logs:

  ./run.sh Phone1
    - In this example we are specifying text name of the device under test - the “Phone1” phone with its earbuds.
    - This same name is used in forming the file names in *results/* folder.
- Initially there is no file *config/config.txt*. It will be generated if it is not present, by entering a simple interactive setup asking to choose IDs of Audio interface and Serial port.
- If Audio and Serial ports need to be configured again, simply delete *config/config.txt* and run again. The questions of setup should appear again and results will be saved in *config/config.txt*

- If everything is fine, results will be saved in results/ subfolder folder that will be created.
  - *results/wav/test_Phone1/* will contain wav files
  - *results/csv/* will contain a .csv files with latency measurement results.
  - *results/xlsx/* will contain a .xls files with latency measurement results with auto generated charts.
- To start new measurement with clean results, please rename or remove the "*results/*" folder.
- In case you need to reconfigure the device for new setup with new Audio interface or audio cables connected to different channels on audio interface, setup can be run again by deleting *config/config.txt* and running the tool from the beginning (e.g. *./run.sh Phone2*).

For more details and configuration example, please refer to Automation scripts release note document.

2.5.3 Manual HTL Measurements

For manual HTL measurement, the test rig and phone are not connected to PC. The user has to manually set desired start and end motion positions and desired rotation speed (through the Control Board user interface), to start the phone playback of predefined 5.1 multichannel audio content provided together with the rig SW release, to enable recording of the 3 streams and to trigger motion through user interface and rotary knob. Please refer *User interface* chapter for more details.

Having all 3 streams recorded simultaneously giving visible interpretation of the 3D Spatial Audio signal processing, according to the position of the head. This test setups allows to do different latency measurements – for example: how fast the phone algorithm reacts to start of the motion, how fast it reacts to the end of the motion, how long it take to re-calibrate itself to other base position, etc.

For noise suppression, additional High Pass Filtering (HPF) has to be applied over recorded Left/Right microphone data – to remove unnecessary noises and to improve readability of the recorded data. Experimental results with Audacity filtering are good for HPF with cut frequency of 1500Hz and slope 48dB.

Example of recorded and HPF filtered audio streams for HTL latency measurement is given at *Figure 10.*

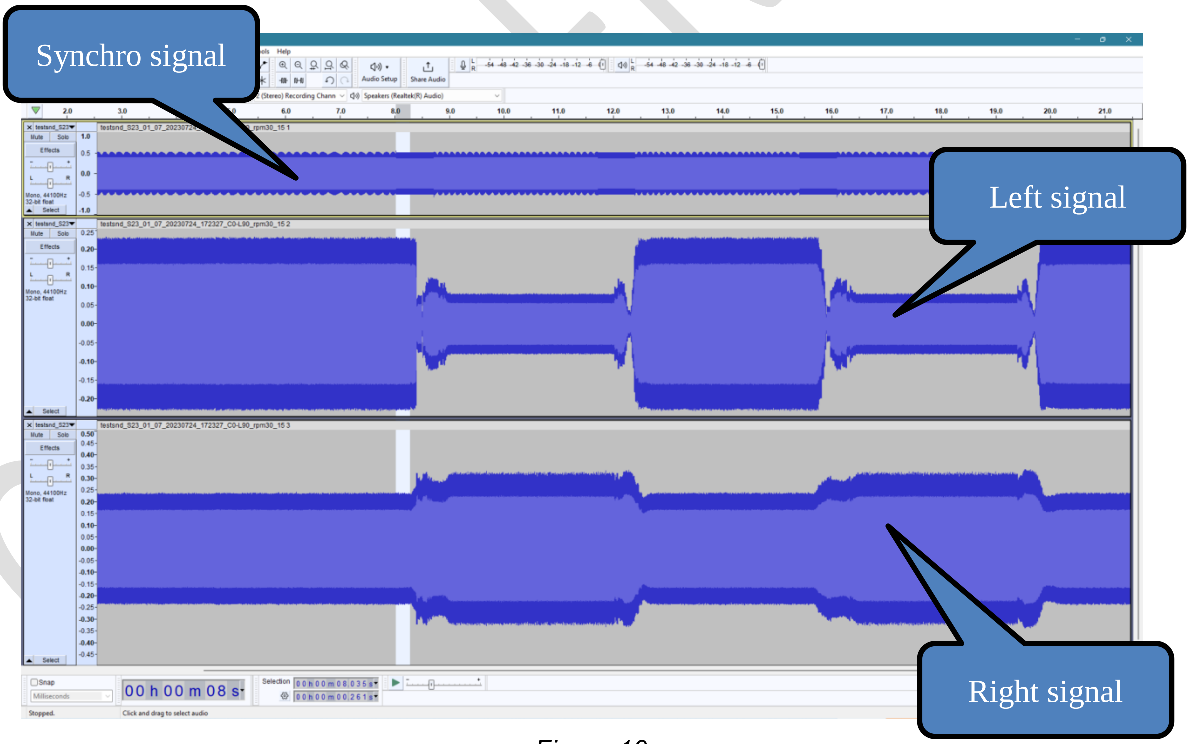


Figure 10

For manual latency measurements according to the start of the motion in given direction, please zoom the signals to level where you can see amplitude change in the Left and/or Right microphone data and just select with the mouse the time interval between the start of Motion event (2kHz -> 10kHz change of Synchro signal) and the first visible change in the amplitude of the Left/Right signals (*Figure 11*).

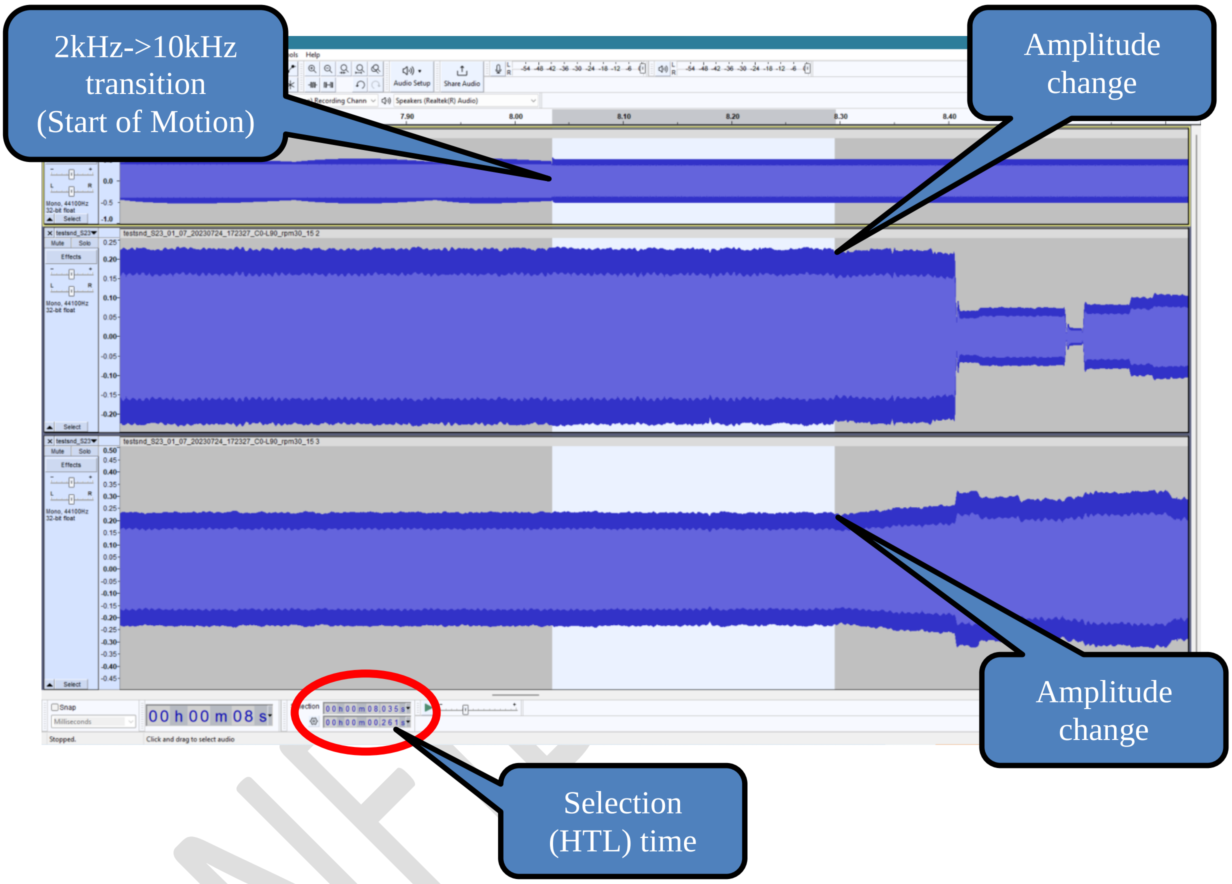


Figure 11