Tester for Otoplastics

With USB interface

Software Manual

**Additional Manual for Android platforms
as of Android version 6.0 (API 23+)**

2025

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Purpose

This additional manual walks you through the process of using a USB leakage tester with an Android platform. The paper describes how to acquire, install and use the *app* (OtoUSB). Also the physical USB connection is briefly discussed It is generally assumed that the user is sufficient familiar with the Android Operating System. The *app* can be downloaded and installed via the Google Play Store by searching for OtoUSB or directly via link:

- https://play.google.com/store/apps/details?id=com.paad.otousbtest&hl=en

Via our site you will find next to the link to the Google Play Store the possibility to download the *app* directly as an OtoUSB.apk file, so without visiting the Play Store. The *app* resides in a .zip file: unzip the file and place OtoUSB.apk on your device in (e.g.) the .../downloads directory and, after tapping the *app* icon, follow the prompts to install the *app*. The *zip*-file is called OtoUSB2.xx and is located on:

- https://www.cursorengineering.nl/en/app-android-usb-en/

Please read the **General Manual** on how to use the USB Leakage Tester itself. The English version is called **OTOmanUSBalpha_EN.pdf** (based on the *alpha app*), or **OTOmanUSBclassic_EN.pdf** (*classic app*), URL:

- https://www.cursorengineering.nl/en/documentation-usb/

1.2 System requirements

To be able to use the *app* for USB leakage testers on smartphones and tablets, at least Android 6.0 (Marshmallow, API level 23) is required. To determine your Android version, e.g. in a tablet looks like: select "Settings > About tablet > Android-version". In addition to the minimum Android version, your smartphone or tablet also must have an OTG function. With the USB OTG (USB On-The-Go) function, with a suitable cable, external USB devices can be connected (and powered) changing the tablet or smartphone into a main device (*a host*). Nowadays almost all Android platforms have an OTG function. The mere presence of a physical USB connector provides no guarantee that the ability to function as host is present. You need to consult your supplier or read the manual of the relevant device; another possibility is to download a so-called USB checker, e.g. "USB Host Diagnostics" from the *Android Play Store*.

1.3 Connection

You can find everything about the cable connection of your USB leakage tester with an Android platform in the General Manual (the link has already been mentioned in §1.1). The leakage tester has a USB-C (or micro-B) connector. The tester can be connected with a suitable USB cable, e.g. USB-A to USB-micro-B, USB-C to USB-micro-B, or with USB-C to USB-C. By the way, a USB-A to USB-C is supplied as standard cable.

Photo 1.3.1: USB-C to USB micro-B   Photo 1.3.2: USB-A to USB-C   Photo 1.3.2: USB-A to USB-C

2 USE

2.1 First use

Before connecting the USB leakage tester to your Android platform install the OtoUSB *app*.
After the insertion of the USB connector, the Android OS will ask you to allow the *app* to access the USB device, tap **OK**. Tapping the checkmark 'Use by default for the USB device' prevents asking this question every time.
If, for one reason or another, the *app* is already running without a tester attached, than you have to terminate the *app* and restart it yourself (see §2.2).
In many cases things go as described above, but with some manufacturers and/or Android versions you have to detect a USB device (in the present case your USB leakage tester) first. With Sony Xperia's (e.g.) proceed as follows: attach the tester to your phone or tablet, nothing will happen for now; tap ⊕ (*apps*) then find and tap *Settings* > *Device connection* > *USB Connectivity* and finally, after tapping *Detect USB device*, the controller *app* and the tester will come to life.

2.2 Normal use

Normally, by connecting the USB tester to the Android platform, the *app* will automatically start and set up the USB leakage tester with the correct data. After a successful connection a beep will sound from the tester and the LED light will turn green. The app will look like the image below. All measurements are performed from this screen.
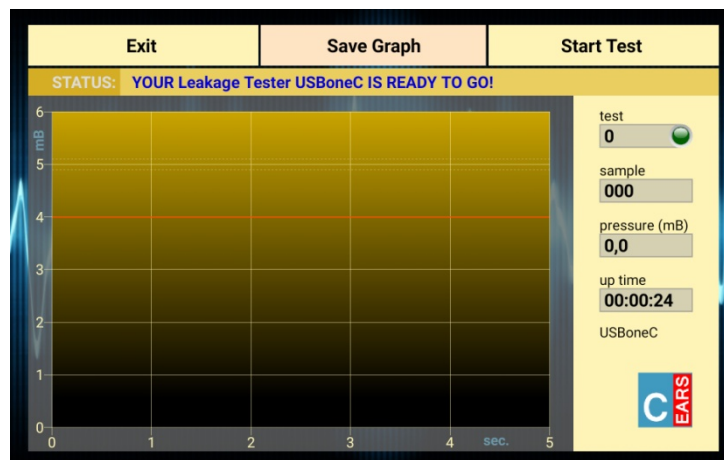
Picture 2.2.1: measurement screen with device name

The *app* will remain active until you terminate it, either by tapping **EXIT**, by unplugging the USB cable, or due to lack of resources by the Android OS itself.
Was the *app* terminated by means of tapping the **EXIT** button, then restart the *app* preferably by tapping on the *app* icon, rather not by unplugging (and plugging in again) the USB cable.

2.3 *Standby* mode

Every Android screen has a timeout after which the screen turns off and goes into *standby* mode.
If the user-set timeout is reached while a connected USB tester and *app* is running, then the physical connection will of course remain established, a beep will sound and the LED light on the tester will light up orange. The *app* also remains active and the up-time clock continues to run. This situation is similar to moving the *app* to the background when using another Android feature. When the *app* is brought back to the foreground it will reconnect to the attached tester and the LED light lights up green again. The only limitation in this situation, for security reasons, is that an ongoing save procedure cannot be completed.

If the screen timeout is annoyingly short, the user can block entering the *standby* mode in the app or adjust it in the Android OS. To temporarily block the screen timeout in the app, please refer to §3.2.5.
The behaviour in *standby* mode with regard to connected USB devices differs per manufacturer and Android version. Example: sometimes in *standby* mode, a USB port is switched off completely causing a connected device to lose power at all. Know that the Android OS independently determines whether to stop an active *app* during the standby mode when system resources become limited.

2.4 Shutdown

When the cable between tester and platform is being disconnected (detaching the tester) the running *app* will automatically close down. Termination can also be done by tapping the **EXIT** button; de tester enters *shutdown* mode (orange led and very low power consumption).

3 MEASUREMENT

3.1 Standard measurement

To start a test touch the **Start Test** button, the air pump in the tester will raise the pressure in the tester within moments to 5mB. The actual leakage measurement will take exactly five seconds.
In this period, ideally, the pressure must remain at 5mB, but an arbitrary margin of 4mB is adhered to indicate that a lower value is to be interpreted as insufficient fitting of the earpiece. The final assessment is, of course, with the hearing care professional. The picture below shows a possible result of a measurement.
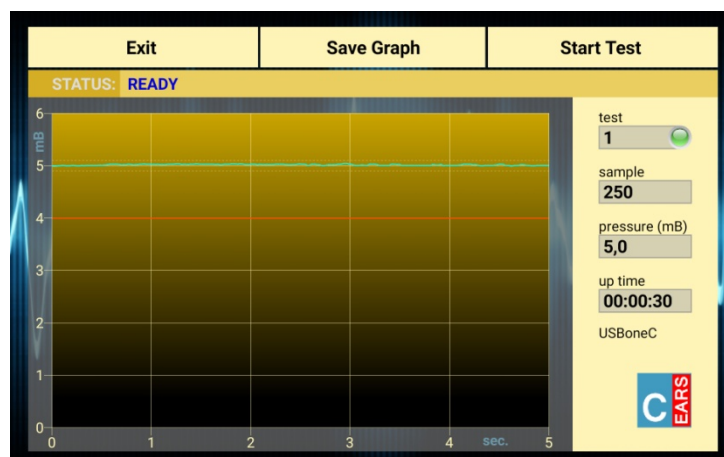
Picture 3.1.2: a measurement result

3.2 Screen legend

3.2.1 control

Three buttons are visible at the top of the screen:

- **EXIT**: the *app* terminates and the leakage tester will enter low power *standby* mode
- **Save Graph**: the graph will be saved to: .../OtoUSB/*filename*.png, the maximum *filename* length is forty-eight (48) characters
- **Start Test**: a new session will start, the button name changes to **Wait...** and for the **EXIT** button temporarily to **Stop Test**: the running test will be terminated

3.2.2 graph

The graph shows a unit on each axis:

- **mB**: the y-axis shows the relative pressure in millibar [mB] with respect to the ambient air pressure, the measurement will be made at a pressure of 5mB; the orange line indicates the (4mB) lower limit that meets the specifications. SI unit for pressure: Pascal [Pa] (5mB ≡ 500Pa).
- **sec.**: the x-axis shows the total measurement time in seconds; the measurement time can only start when the initial pressure of 5mB has been reached

3.2.3 information panel

Details are displayed at the right of the graph and provide information about the progress of a session:

- **test no.**   **:** [number]   : the number of sessions done since the *app* started
                        [*LED colour*]: *example*
                        [d. green]   : session running
                        [l. green]   : session completed, result within specs
                        [orange]   : session completed, not within specs or session terminated
                        [red]   : session started, but unexpectedly halted or ended erroneously
- **sample no.** : the number of measurement moments (= samples) during the session
- **pressure**   : the instantaneous measured pressure, or the last measured pressure (**milliBar**)
- **up time**   : the time (hh:mm:ss) passed since *app* has an active connection to a USB tester
- **USBxxxx**   : the name of the attached USB leakage tester

3.2.4 status

The status of the leakage tester is shown to the right of '**STATUS**:' on the status line (which is displayed above the graph). The leak tester has a number of *states*; the most important of these are discussed in the next chapter. If only the user's attention is required, the background will turn orange, however, in the event of an error message, the background will turn red.

3.2.5 logo (key function)

Tapping the *Cursor* logo will open a popup window for five seconds, with information about the tester: software, firmware, etc.; by tapping the screen again it disappears immediately. If the finger is held on the logo for a little longer, the lock screen timeout (as set by the user) is lifted: the screen will remain **on** continuously until the action is repeated or the *app* has been terminated. The screen mode is always visible in the popup window.

3.3 Archiving

3.3.1 general

After each test the user has the opportunity to save the resulting graph (colour and b/w) to a file for later handling by touching the **Save Graph** button, a pop-up window appears allowing the user to enter a filename:
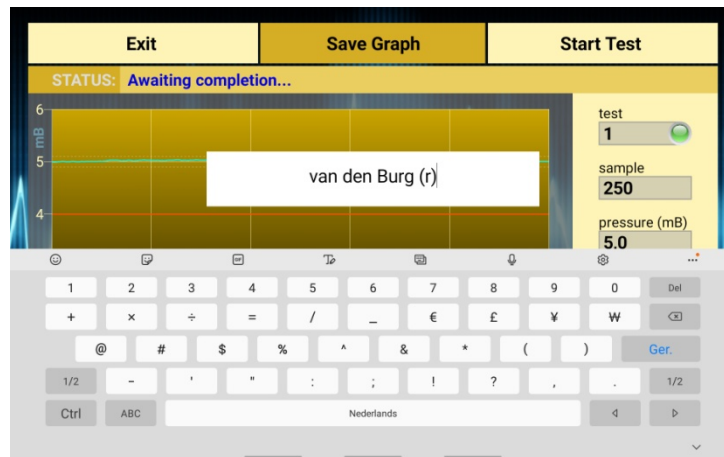
Picture 3.3.1: saving a graph

3.3.2 procedure

Click in the *pop-up* window on the so-called *placeholder* (filename[.png]) and enter the name; a maximum of 48 characters can be used, conclude with [**Done**]. Touching the screen outside this pop-up window terminates the saving procedure, but can be restarted or continued by tapping **Save Graph** again until the saving procedure has been completed or until the next measurement starts. The graphic result of an ear measurement can only be saved once. Two files will be saved: one in colour (avg. size 38kB) and one in b&w (avg. size 21kB) with a transparent background. A date/time stamp is always added at the bottom *inside* the graph.
The graph will be saved to the so-called 'Portable Network Graphics' file format, therefore the *extension* name reads **.png**. One should not enter this extension name: the extension will be automatically added.
For status messages during the saving procedure, see section 4.4.

3.3.3 map location

The graphs will be captured and stored in the directory .../Internal storage/Pictures(Images)/*OtoUSB*. Use an *app* e.g. "My Files" from the *Google Play Store* and from the aforementioned map you can edit the graphs, email them etc. or possibly transfer them to another platform where the report will be processed.
For certain Android versions, storing data in the internal memory (still) requires the user's permission read about this in the next section (3.3.4).

3.3.4 permission

The *app* will ask for permission via a so-called pop-up window at times where permission is required.
Once permission has been given, the question will not be asked again until the permission is withdrawn.
Without this permission the *app* will work fine, but without graph storage.
All Android versions from 6 (API 23) up to and including version 10 (API 29) request permission to store data.
From Android version 11 (API 30) permission is no longer required and thus no longer asked for.

4 STATUS MESSAGES

4.1 Permissions

Starting with Android 6.0 (Marshmallow, API 23), the app must ask the user for permission to save files. However, as of Android version 11 (Red Velvet Cake, API 30), this permission is no longer required.

So on older Android versions the OtoUSB *app* still needs permission for:

- **Storage**: this permission is needed to obtain access to your memory to save test results (graphs); without permission the *app* will work fine, but it is impossible to save results

On occurrences in the *app* where permission is necessary, but not (yet) granted, the *app* will ask for permission in a pup-up window. A once given permission remains granted until consent is withdrawn.

4.2 Device connection

After attachment or activation, the *app* tries to establish a USB connection with the USB leakage tester. Messages during this phase are shown on the status line.

An overview of the most important messages during connection setup:

- **NO TESTER ATTACHED**: not a single USB leakage tester available on this platform, resulting in:
- **ATTACH YOUR (USB) LEAKAGE TESTER PLEASE**: the *app* waits approx. 30s for a tester to be attached
- **CONNECTING TO 'Leakage Tester USBxxxx'**: the attached tester has been recognized and information is being exchanged
- **IMPROPER RESPONSE FROM 'Leakage Tester USBxxxx'**: the leakage tester did not respond at all or erroneously*
- **CHECKSUM ERROR in 'Leakage Tester USBxxxx'**: the operating settings in the tester have not been received or processed correctly; some repetition attempts will be performed
- **YOUR 'Leakage Tester USBxxxx' IS READY TO GO!**: the connection has been established, the leakage tester is ready for the first test
- **DEVICE 'Leakage Tester USBxxxx' DETACHED**: de cable connection has been broken*

* leads to **Exit (only)**: terminate *app* and/or dis- and reconnect leakage tester

4.3 Measurement

Normal messages while a test is running:

- **SESSION START**: just a short notice
- **AIR IN**: the air pump will try to pressurise the system to just above 5mB
- **STABILIZING PRESSURE**: the pump halts, the target pressure has been reached and the system will check for a short time whether the pressure is stable enough; after this phase, *when the leakage is relatively high*, the pressure curve might enter the graph below 5mB
- **MEASURING PRESSURE**: the actual measurement session runs, the graph will be drawn
- **SESSION FINISHED**: the last sample has been delivered, the sessions has been completed and the system will be depressurized, while the system is depressurizing one can already start filing the graph
- **READY**: the test has been completed, the leak tester is ready to take a new measurement

Possible messages on errors or warnings during or after a session:

- **SESSION TERMINATED**: tapping **Stop Test** causes this message
- **TARGET PRESSURE (5mB) NOT REACHED**: the system looses so much air that the pressure cannot be increased sufficiently: the otoplastic may have a very poor fit or placed incorrectly; before drawing any conclusions, the system should always be checked for leaks without an otoplastic, also check whether the air pump operates properly
- **PRESSURE UNSTABLE**: a stable test-pressure cannot be acquired, if it persists service is necessary
- **PRESSURE DISCHARGE TOO SLOW**: after a session, pressure should leave the system within a few moments; when air cannot leave - or only with difficulty - the system, this indicates internal contamination or valve malfunction, if this message remains service is required
- **READY (BELOW REJECTION LEVEL)**: the session has been completed, but at some point during that session a pressure of 4mB or less was measured
- **CONNECTION WITH 'Leakage Tester USBxxxx' LOST!**: there *was* a connection, but platform or *app* lost the connection with the leak tester, terminate *app* and dis- and reconnect the tester*

* leads to **Exit (only)**: terminate *app* and/or dis- and reconnect leakage tester

4.4 Archiving

Attention: please read §4.1 about privacy rules concerning archiving.

Common messages during archiving:

- **file '*filename*.png' already exists**: the *filename* already exists, make up and (re-)enter a different name
- **Filename cannot be empty**: the *filename* may not be empty
- **Character (x) not allowed in a filename**: although strictly allowed, characters are banned from the range **x** = ["\", "/", "*", "?", ":", "<", ">", "\\", "|"] for compatibility reasons; these tokens are automatically deleted as you type
- **Press 'Save Graph' again to resume**: because the screen has been touched outside the text box, the text box disappeared: tap **Save Graph** again, already filled characters will be preserved
- **Graph has been saved**: the file '*filename*.png' has been saved in the directory .../OtoUSB