# DispLagBox: Simple and Replicable High-Precision Measurements of Display Latency

Patrick Stadler
Patrick.Stadler@stud.uni-regensburg.de
University of Regensburg
Regensburg, Germany

Andreas Schmid
Andreas.Schmid@ur.de
University of Regensburg
Regensburg, Germany

Raphael Wimmer
Raphael.Wimmer@ur.de
University of Regensburg
Regensburg, Germany

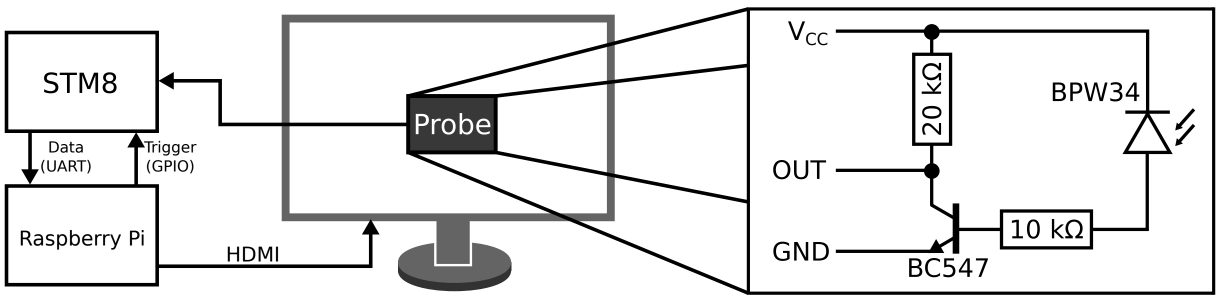


**Figure 1: Schematic illustration of the *DispLagBox* measuring device. Software running on a Raspberry Pi writes an image into the internal framebuffer and triggers a high-precision timer running on the STM8 microcontroller. Once the attached display changes brightness, a photodiode stops the timer again. Latency is reported back to the Raspberry Pi via UART. This process is automatically repeated multiple times in a row.**

## ABSTRACT

The latency of a computing system affects users' performance. One important component of end-to-end latency is *display lag* - the time required to turn framebuffer contents into photons emitted by a computer screen. However, there is no well-documented and widely available method for measuring display lag. Thus, the effect of display lag is rarely considered in scientific studies and system development. We developed DispLagBox, a simple open-source device for measuring display lag. It supports the *International Display Measurements Standard* but also offers additional metrics for characterizing display lag with a resolution of 0.1 ms. The device, based on a Raspberry Pi computer, measures the time between VSYNC and a change in brightness on the connected display. Repeated measurements can be conducted automatically, so that not only average latency but also latency distributions for each device can be reported. For most displays we tested, DispLagBox reports latencies that are close to those reported by a commercial black-box measurement device. Typically, the difference is 1 – 3 ms.

## CCS CONCEPTS

• **Hardware** → Hardware validation.


*MuC'20, September 6–9, 2020, Magdeburg, Germany*

ACM ISBN 978-1-4503-7540-5/20/09…$15.00
https://doi.org/10.1145/3404983.3410015

## KEYWORDS

latency, display, human-computer interaction

**ACM Reference Format:**
Patrick Stadler, Andreas Schmid, and Raphael Wimmer. 2020. DispLagBox: Simple and Replicable High-Precision Measurements of Display Latency. In *Mensch und Computer 2020 (MuC'20), September 6–9, 2020, Magdeburg, Germany.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3404983.3410015

## 1 INTRODUCTION AND MOTIVATION

The latency (also known as lag) of a interactive computing system has a measurable effect on the user's performance with it. This has been confirmed in several studies. MacKenzie and Ware [7] have shown that higher latency increases the difficulty of pointing tasks significantly. Jota et al. [5] have found that the minimum latency users are able to notice when operating touch-based systems can be as low as 10 ms. Attig et al. [1] reviewed empirical studies on the influence of latency on user performance. They conclude that even latencies starting from 16 to 60 ms can negatively affect the performance of users for a large number of tasks. Consistently low latency is especially important when performing critical tasks such as remote-controlling robots or vehicles - but also the video game community has a high demand for fast and reliable systems [6]. Thus, input devices and displays for gamers are optimized for low latency. For example, current gaming mice and gamepads support polling rates of up to 1000 Hz and computer screens offer refresh rates of up to 240 Hz. The overall latency of a system is called *end-to-end latency.* It is defined as *"the time difference between a user input to a system and the display of the system's response to that input"* [3]. End-to-end latency is comprised of multiple *partial latencies* such as *input device latency*, *processing time* of the computer, *network latency*, and finally *display lag* - the latency added by the display. It is important to measure and understand those partial latencies to find bottlenecks influencing the end-to-end latency of a system instead of regarding the whole process from user input to system output as a black box[12]. Wimmer et al. have developed a cheap and easily replicable device for measuring the latency of input devices, such as computer mice, keyboards and game pads[2, 12]. We propose a similar approach for measuring the latency of computer displays in this paper. This allows experts and consumers to automatically measure a display's latency hundreds of times in a row in order to precisely characterize it.

## 2 RELATED WORK

Even though many reviews of computer displays, e.g. by hardware magazines or websites, specify display latency, the results of different reviewers can not be simply compared among each other as there is rarely a documentation of the used measurement process and hardware available to the public. Furthermore, there is only information on displays that have actually been tested by those hardware reviewers. Tests of obscure or old devices might not exist or have become unobtainable.

To conduct their own measurements of display latency, consumers have invented several procedures to approach this problem. One common method is displaying a stop watch with a computer and splitting the video signal between a CRT monitor and the display that should be tested. Both displays are then photographed and the time difference between the two stop watches is considered to be the difference in display lag between the devices [9]. As criticized by Thieman in a blog post on the German website *prad.de*, this method assumes that both signals are being sent at the same time by the video card. Furthermore, this method does not measure absolute latencies but only a difference in latency between the two displays [10].

A different approach to measuring display latency is to measure the time from an image being rendered on a computer until a photo sensor detects a change in brightness on the screen. This method has been for example be used by Seo et al. to measure the latency of head-mounted displays [8]. A commercially available device using this approach to measure display latency is the *Video Signal Input Lag Tester* by *LeoBodnar*[1], which uses an FPGA to generate a HDMI signal and a photo sensor to measure the output on the display. However, this device does not use a standardized measuring process and its circuitry and source code are not available to the public. Thus, this device's measurements are not comparable to the results of different measuring approaches.

Measuring display latency with the setup described by the *International Display Measurements Standard* (IDMS) [4] requires laboratory equipment such as an oscilloscope and a photo detector, limiting its practical use to professionals who can afford and operate such devices.

With the measuring device presented in this paper, we implement the measurement process for display latency proposed by the IDMS using inexpensive off-the-shelf components. Due to the free and publicly available schematics and source code, the device can be built by anyone with basic knowledge of electronics and computers, allowing for comprehensible and replicable measurements following an industry standard.

## 3 OUR APPROACH TO MEASURING DISPLAY LATENCY

### 3.1 Definition of Display Latency

In chapter 10.3 of the *Information Display Measurement Standard* [4], a standardized procedure to measure display latency is proposed. A video source connected to the display under test changes the displayed image from black to white and sends a trigger signal to an oscilloscope. A photodetector pointed at the center of the display detects the change in brightness and also sends its detected voltage to the oscilloscope. The time between the trigger signal and the display's brightness reaching 50% of its maximum luminosity is defined as the display's latency (Fig. 2).

We use this definition for DispLagBox and implemented the proposed setup using inexpensive off-the-shelf components instead of professional grade lab equipment.

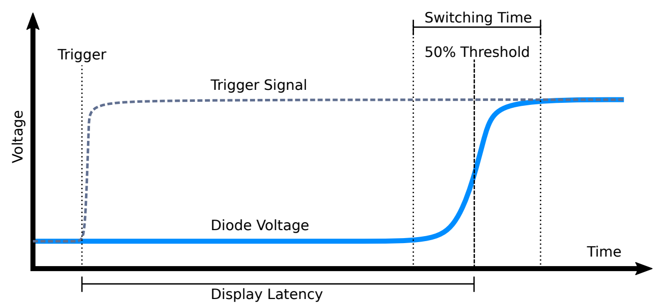


**Figure 2: Visualization of our definition of display latency. The voltage for the 50% threshold is determined during a calibration process by displaying a 50% grey image.**

### 3.2 Components

To ensure easy replicability of DispLagBox, we focused on using inexpensive and broadly available components. A Raspberry Pi 4 is used to display the user interface of the measuring software and the black and white images used for the measurements. We decided to use the Raspberry Pi because of its low acquisition cost, the broad availablity, and the comparatively well documented and open source graphics stack. The software on the Raspberry Pi is written in C and renders images directly to the framebuffer to decrease possible influences on the measurements caused by graphics frameworks.

The Raspberry Pi is connected to a STM8 microcontroller [2] via UART. The STM8 performs the actual measurement by reading the value of a *Osram BPW 34* photo diode through its built-in 10 bit ADC.

The photo diode is enclosed in a separate 3d printed case so it can be easily placed on the screen and fixed in position with a

[1]http://www.leobodnar.com/shop/index.php?main_page=product_info&cPath=89&products_id=212

[2]https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm8-8-bit-mcus/stm8s-series.html

velcro belt. Additionally, the case shields the sensor from ambient light to reduce disturbances caused by external influences.

The latencies introduced by the photo diode and the ADC are in the range of microseconds, thus they do not confound the measurements in a significant way.

The complete source code, circuitry and PCB layouts for the device are available under a open source license[3] so the device can be replicated easily. The total cost of the device is around 65 Euros (price includes Raspberry Pi 4), making it affordable for non-professionals.

## 3.3 Measuring Process

The measurement process implements the standard for measuring video latency proposed by the IDMS [4].

Before starting a measurement, the device is connected to the display under test via HDMI. The test probe is then placed at the center of the display with the photo diode facing the screen and fastened with a velcro belt. The exact position of the probe is important as the displayed image is refreshed from top to bottom, so a slight displacement in height influences the measured result.

The screen's brightness has to be set to the maximum level (100%) to avoid measurement errors that could occur if the display's backlight brightness is controlled by PWM.

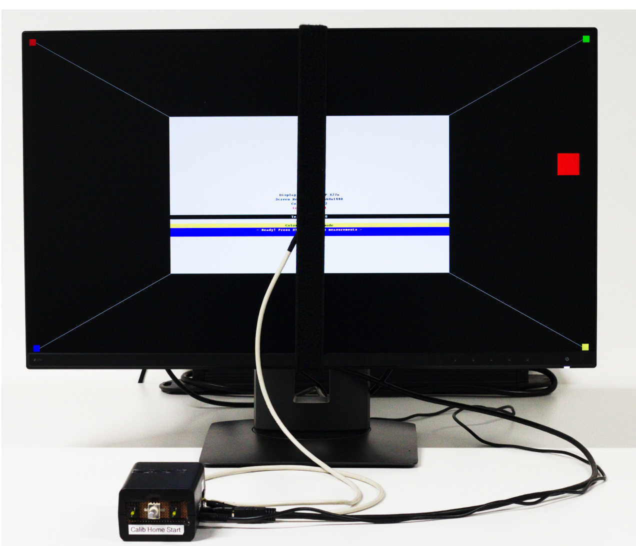


**Figure 3: The measuring setup. The probe is attached to the monitor with a velcro belt. The UI of DispLagBox is shown on the display.**

To compensate for the difference in brightness between individual models and makes, the device needs to be calibrated before testing a new display. To calibrate for the IDMS approach, the Raspberry Pi generates a full screen 50% gray image which is recorded via the attached photodiode and stored as a threshold value by the microcontroller. The process of calibration ensures even conditions among all tested monitors.

The test procedure itself is composed of following steps:

[3]https://hci.ur.de/projects/display_latency

(1) DispLagBox waits for the display's VSYNC callback (provided through the *Dispmanx* interface), then changes display color from black to white by writing to the VideoCores's framebuffer and starts STM8's measurement routine with a trigger signal sent via GPIO.
 The time between writing to the framebuffer and the HMDI signal being sent to the display is negligible according to a Raspberry Pi engineer's post in the official forums [11]: *"(t2 - t1) is effectively zero because the PV starts pulling pixels from HVS during the blanking period."*
(2) At the 50% brightness threshold (determined during the calibration process) the current timestamp is stored.
(3) The STM8 stops its measurement timer and sends all measurements (composed of timestamp and raw 10 bit ADC value) to the Raspberry Pi via UART.
(4) The Raspberry Pi creates a CSV file with the measured values for each test series, automatically includes the name of the display provided by EDID and saves this file to a USB memory stick.

The device measures one brightness value per 100 µs, providing a sufficient resolution for display latency which is normally in the range of single digit to low two digit milliseconds. This allows for visualizing the change in brightness over time, highlighting differences in how the pixels are driven on different displays (Fig. 4). Each measurement series consists of 20 individual measurements so possible variances in latency become visible. Each measurement series takes under one minute to finish, so one monitor can be tested in around 10 minutes (including connecting the monitor, calibration and transferring the results to a PC).

## 4 EXAMPLE MEASUREMENTS

To validate the reliability of our measurements, we tested several monitors and compared the latencies to values produced by *LeoBodnar*'s *Video Signal Input Lag Tester* (VSILT) at the center position.

In addition, we replicated the test setup described in chapter 10.3 of the IDMS standard [4] using the same components as we used for DispLagBox but replacing the STM8 with a *RIGOL DS1102E* oscilloscope to measure the time between trigger signal and 50% brightness reached. The 50% brightness threshold was determined by showing a 50% grey image and measuring the voltage of the photo sensor.

Because DispLagBox performs multiple measurements per test series, we can not only report a single value for display latency, but a distribution of values (Fig. 5) as well as a curve of measured brightness values with a 100 µs resolution (Fig. 4).

As an example, we conducted a test series of 190 measurements with seven different monitors that were available in our lab. All image enhancement features as well as overdrive have been turned off for the measurements.

The results of our measurements can be seen in figure 5 and table 1. While many displays of different manufacturers have similar latencies, we could find outliers in either direction, with the *Dell U2713H* being remarkably slow. Measuring the same device with the *LeoBodnar VSILT* and an IDMS-setup confirms this outlier.

The latencies measured with DispLagBox are comparable to the ones measured with an actual IDMS-setup. However, our results do

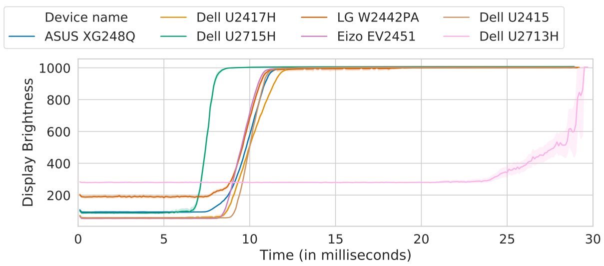


**Figure 4: Measured transition from black to white on several monitors. Higher values represent a brighter screen.**

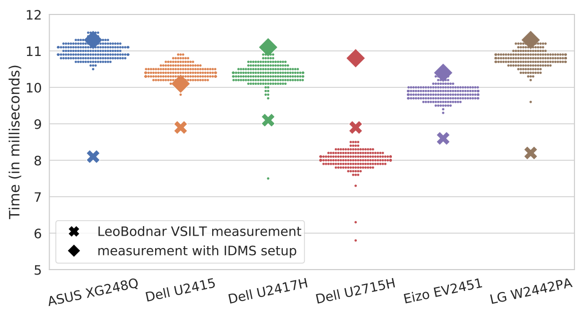


**Figure 5: Latency distributions of several displays. The *Dell U2713H* is excluded here because showing its latency would require a significantly longer y axis.**

**Table 1: Measurement results. All values in milliseconds.**

| Device | min | med (std) | max | VSILT | IDMS |
|---|---|---|---|---|---|
| Asus XG248Q | 10.5 | 11.0 (0.21) | 11.5 | 8.1 | 11.3 |
| Dell U2415 | 9.8 | 10.4 (0.19) | 10.9 | 8.9 | 10.1 |
| Dell U2417H | 7.5 | 10.4 (0.31) | 10.9 | 9.1 | 11.1 |
| Dell U2715H | 5.8 | 8.0 (0.27) | 8.5 | 8.9 | 10.8 |
| Eizo EV2451 | 9.3 | 9.8 (0.19) | 10.4 | 8.6 | 10.4 |
| LG W2442PA | 9.6 | 10.8 (0.68) | 19.6 | 8.2 | 11.3 |
| Dell U2713H | 21.5 | 24.8 (1.51) | 28.4 | 38.3 | 32.8 |

not match the values measured with the *LeoBodnar VSILT* exactly. A reason for this could be the different measurement standard used by VSILT, as well as the sensor's exact position on the screen. We contacted *LeoBodnar* in regards to this issue but they refused to give us any information on their device.

## 5 LIMITATIONS

While using inexpensive and widely available components ensures replicability of results, it also effects limitations. Currently, monitors have to be connected to DispLagBox via HDMI as this is the only display interface offered by the Raspberry Pi 4. Furthermore, the Raspberry Pi 4 offers a maximum display refresh rate of 60 Hz and does not support adaptive synchronization techniques which reduce latency such as FreeSync and G-Sync. Especially gamers might be interested in measurements of displays with refresh rates of 144 or 240 Hz and support of those technologies. These limitations could be avoided by switching to a more powerful device than the Raspberry Pi. However this change would increase hardware cost and make it harder to compare measurements conducted by different people.

Even though the results of DispLagBox match the latencies measured with an IDMS-setup within a 1 millisecond range, they differ from latencies measured with the *LeoBodnar VSILT* by 1 - 3 milliseconds. In future work, the reasons for those differing results, as well as some outliers, have to be investigated.

## 6 CONCLUSION AND OUTLOOK

We have created a cheap and easily replicable device for measuring display latency based on inexpensive off-the-shelf components. The measurement process implements the IDMS standard and can replicate the results of an IDMS-setup using professional grade lab equipment within a one millisecond range. In contrast to similar devices, DispLagBox can measure distributions of latency instead of only reporting a single average value.

By measuring more monitors with DispLagBox, a database for latencies of different devices could be created. We plan to provide a platform for others who replicated it to share the results of their measurements. This platform, measurement results, as well as a documentation and assembly guide for the device can be found at the project website[4].

## REFERENCES

[1] Christiane Attig, Nadine Rauh, Thomas Franke, and Josef F Krems. 2017. System Latency Guidelines Then and Now–Is Zero Latency Really Considered Necessary?. In *International Conference on Engineering Psychology and Cognitive Ergonomics*. Springer, 3–14. https://doi.org/10.1145/2470654.2481317

[2] Florian Bockes, Raphael Wimmer, and Andreas Schmid. 2018. LagBox–Measuring the Latency of USB-Connected Input Devices. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–6.

[3] Ding He, Fuhu Liu, Dave Pape, Greg Dawe, and Dan Sandin. 2000. Video-based measurement of system latency. In *International Immersive Projection Technology Workshop*. 111.

[4] International Committee for Display Metrology. 2012. Information display measurements standard. *Society for Information Display (SID)* 135 (2012).

[5] Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. How fast is fast enough?: a study of the effects of latency in direct-touch pointing tasks. ACM Press, 2291. https://doi.org/10.1145/2470654.2481317

[6] Ulrich Lampe, Qiong Wu, Hans Ronny, André Miede, and Ralf Steinmetz. 2013. To Frag or to Be Fragged - An Empirical Assessment of Latency in Cloud Gaming. SciTePress - Science and and Technology Publications, 5–12. https://doi.org/10.5220/0004345900050012

[7] I Scott MacKenzie and Colin Ware. 1993. Lag as a determinant of human performance in interactive systems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. 488–493. https://doi.org/10.1145/169059.169431

[8] Min-Woo Seo, Song-Woo Choi, Sang-Lyn Lee, Eui-Yeol Oh, Jong-Sang Baek, and Suk-Ju Kang. 2017. Photosensor-Based Latency Measurement System for Head-Mounted Displays. *Sensors* 17, 5 (May 2017), 1112. https://doi.org/10.3390/s17051112 Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[9] Vincent Teoh. 2016. TV Input Lag Test Database. http://www.hdtvtest.co.uk/news/input-lag Library Catalog: www.hdtvtest.co.uk.

[10] Thomas Thiemann. 2009. Untersuchung des Testverfahrens einer Input-Lag-Messung. https://www.prad.de/untersuchung-des-testverfahrens-einer-input-lag-messung/ Library Catalog: www.prad.de.

[11] timg236 (Raspberry Pi Engineer). 2018. Possible to measure time of HDMI output generation based on framebuffer copy? - Raspberry Pi Forums. https://www.raspberrypi.org/forums/viewtopic.php?f=67&t=257280

[12] Raphael Wimmer, Andreas Schmid, and Florian Bockes. 2019. On the Latency of USB-Connected Input Devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. ACM Press, Glasgow, Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300650

[4] https://hci.ur.de/projects/display_latency