

Borduhr «8 DAYS» Chronograph

Author: Simon Widmer

Preface

I am coding this clock just out of fun. The clock face design is fictitious as I do not plan to copy an existing clock. However, I was well inspired by some famous aircraft clock models mostly from the (pre-) Second World War. The following list does not claim to be exhaustive, but is intended to provide interested readers with information on historical aircraft dashboard clocks.

- Allon a Versailles (model unknown)
- Borletti 2354
- Breguet Type 11
- Breitling Wakmann 618 12-24, 640-24-10, 118-12-24 and the later 651-12, A-13
- Carley & Clemence MK IIIA
- Dodane Type 11
- Elgin AC-34898, AN-5743-1
- Hamilton-Elgin E-37500, CDIA and A-13A
- Jaeger LeCoultre A-10 Chronoflite
- Junghans FL. 22600, FL. 23885
- Kienzle FL 23886
- Leonidas 429 and 1-665
- Lip Typ 14
- Longines Wittnauer AN5743-L2
- Mathey Tissot Type 12
- Molnija AChS1 and 81322
- Nero Lemanian 7518
- Omega (Askania) FL.22601
- Revue Thommen Type A-11 and B-13
- Smith and Jaeger Lecoultre Mk II and Mk IIIA
- Seikosa Type 92
- Sonja 1-1947
- Waltham XA, AN 5743 and Type A 13A MIL-C-6499
- Zenith EFAP and Type 20
- and many more

This clocks are masterpieces in engineering and design on its own. The readability of the clocks with the deep black dial is unsurpassed. All of them are labeled *8 DAYS*, indicating that they could run for 8 days after winding it fully up. Please don't get me wrong: I'm not a fan of war at all – I just like the *aircraft clocks* from that time.

Description

Measured on several platforms, the CPU-Load of Borduhr should remain below 2% – but mostly even below 1%. Another goal is to keep it scalable (resizing the window). This gave me a big headache but the result was worth it in my opinion. Why did I titled this clock «Borduhr – 8 DAYS»? Because I just could not think of a better, short name yet. *Borduhr* is the german term for *aircraft dashboard clock*.

Note that the label «*SWISS MADE*» of my «Borduhr» is correct indeed – I am living in the «clock-country» Switzerland and try to continue the art of designing clocks. It is just that this one is made out of pure software instead of gear wheels.



Borduhr «8 DAYS»

Installation

Pre-Requirements

Kindly note that there is no installation as such needed of running “Borduhr”. As it is coded in Tcl/Tk, of course one has to install the interpreter first.

- Linux: For most distributions, it is a matter of: `sudo apt install tcl && sudo apt install tk`
- Windows: go to www.ironetcl.com and follow the installation instructions.
- Sourcecode: Or get the sourcecode of [Tcl/Tk](#) and compile on your own. This is no big deal and should be straight forward under Linux.

Unzip Borduhr

Then, unzip borduhr into your chosen destination.

- Linux:
 - `chmod +x ./borduhr.tcl` (if the file is not yet executable)
 - run `./borduhr.tcl` in a terminal shell or make a desktop launcher
- Windows: double-click on borduhr and chose `wish.exe` as the program, that could run this code.

Functions



The red pushbutton (red pusher) on bottom right is used for the chronograph and works as follows:

- first click: start the chronograph
- second click: stop the chronograph and show the estimated time
- third click: clear the chronograph and set back the hands to zero



The black pushbutton button on bottom left is used as follows:

- single click to open the popup menu:
 - Night mode: This mode switches the clock to fluorescent colors. This was realized with a UV lamp (black-light headlamp), that illuminates the cockpit instruments from behind. This caused a reaction within the radium dials and as a result the dials and gauges lit up. This would not to give away the position of the fighters in night operations.
 - Day mode: This mode switches the clock to daylight condition again.
 - Help: That's the one you are currently reading
 - About: self explaining dialog
 - Quit: Close this clock. You do not need it :)

Features

- Platform independant. I test it regularly with:
 - Linux (*Debian Bookworm*) on x86
 - Windows 11 on x86
 - Linux Wine emulator (*emulating Windows 10*)
 - others platforms (such as MACOSX) may be supported in future.
- works at least with TCL/TK 8.5.6 and 9.0
- Library independent. No other dependencies to e.g. tcllib or other (binary) libraries (*.dll, *.so) are required.
- High readability with red second hand and green minute- and hours hand.
- integrated 12-hours chronometer [complication](#)
 - hours subdial: So called *complication* located on top below number **12**. Can time periods of up to 12 hours.
 - minutes subdial: Complication located next to number **9**. Can time periods of up to 60 minutes.
 - seconds subdial: Complication located next to number **3**. Can time periods of up to 60 seconds.
- Day name indicator: Weekday complication located above number **7**.
- Calendar subdial: integrated analog day/date complication located on bottom above number **6**.
 - long orange arrow-hand is for the day
 - short blue arrow-hand is for the month
- AM / PM complication located below number **1**
 - AM: from Latin *ante meridiem*, translating to “before midday”

- PM: from Latin post meridiem, translating to “after midday”
- Annual complication: Full year display located above number **5**
- Colors and fonts are highly configurable over `settings.conf`. However, the design itself cannot be configured unless the sourcecode is altered.
- Window-position of the clock is configurable:
 - either leave that task to the windowmanager itself or
 - set a fix window-position
- Resizable! No fixe window sizes – this makes it even more flexible.
- Day/night mode. After switching, the clock face is glowing as if lighted with a ultraviolet lamp.

Specialties

I have never liked the fact that the big second hand is part of the chronograph function instead of the smaller one (on my wristwatch) – so please forgive me but I’ve exchanged those functions. In my opinion, this design makes it easy to distinguish the main watch from the stopwatch.

I thought that it would be a cool function to add not only a long day-hand but also a smaller month hand. You will get used to it soon. Still, I don’t know how to display the year in a cooler *analog* way. Maybe I can find a solution while having a look at the perpetual watches that are around.

Note that the numbers containing a **1** on the clockface is in fact a (sans-serif) **I** as lots of clock of that time. Due to the fact that I could not find such a font and fonts in general cannot be loaded into Tcl/Tk without installing it, I decided to handle it in this way. Hope that this is pleasing you anyways.

Precision

Please note that this application is as precise as your system is. Under Linux, as long as NTP synchronisation is activated, the precision is in other words as accurate as it could be with this technology. See also: [timesyncd service](#)

Limitations and trade-offs

- The drawings are built on the native and powerful canvas widget. Unfortunately, it does not do antialiasing to smoothen the drawings (except for fonts). On my 4K display, this is not a big issue but on displays with lower resolutions, it can be disturbing. Hopefully, the Tcl/Tk coredevelopers are going to change that soon.
- No measuring of fractional seconds for the chronometer. As far as I tested, this would be noticeable in a much higher CPU load.
- Just one tick per second, no smooth movement of the second hand. (compare to the mechanical masterpieces that move the second hand eight or even ten times a second!)
- Bold fonts are not supported yet (do we need this?)
- Themes with custom clockfaces are not supported.
- Round window without borders and titlebar is not supported as Tcl/Tk does not support transparent windows yet.

Thanks

My thanks goes to:

- John Ousterhout et all for the invention and the development of [Tcl/Tk](#)
- John Gruber, the inventor of [markdown](#). His syntax definitions made this document possible.
- John MacFarlane, the developer of [Pandoc](#) to convert the README into *html*.
- the developers of [wkhtml2pdf](#), so the *Html*-file can be converted to a *PDF*

Feature requests?

For any kind of feature you would like to see or bug reporting or thank you's:

[Email me](#)