Gostim2 User Guide

Christophe Pallier

Contents

Gostim2 User Guide

Version: 0.2.0 **Author**: Christophe Pallier **License**: GNU General Public License v3

What is Gostim2?

Gostim2 is a multimedia stimulus delivery system designed for experimental psychology and cognitive neuroscience. It presents images, sounds, text, and video according to a **fixed schedule defined in a plain-text table** — no programming required.

Gostim2 is best suited for experiments where:

- Stimuli are presented in a predetermined order and at known times.
- Precise timing is critical (fMRI, MEG, EEG studies).
- You want a simple, portable setup that runs on Linux, Windows, and macOS.

> **Important limitation**: Gostim2 does **not** support real-time branching or feedback. The presentation schedule is fully determined before the experiment starts. All keypress events are recorded, but they cannot alter stimulus delivery. If you need adaptive or interactive paradigms, consider goxpyriment (Go) or expyriment (Python).

GUI vs. CLI: which should I use?

Gostim2 ships as **two separate programs** that share the same engine:

	`gostim2-gui` (graphical)	`gostim2` (command-line)
How to start	Double-click or run from terminal	Run from a terminal with flags
Configuration	Interactive form — point-and-click	Flags and/or a TOML config file
Settings persistence	Saved automatically to `gostim2_config.toml`	Must be re-specified each time (or use `-config`)
Best for	First-time users, occasional experimenters	Scripting, automation, HPC clusters, TTY-only servers
Requires display	Yes	No (can run headless in TTY mode on Linux)
Timing	Identical to CLI	Identical to GUI

Both programs use **the same timing engine** and produce **identical output logs**. Choosing one over the other is purely a matter of workflow preference.

Installation

Pre-built binaries (recommended)

Download the appropriate package for your platform from the GitHub Releases page:

Platform	Package
Windows	`gostim2-setup.exe` (installer) or `.zip`
macOS Apple Silicon	`.app.zip` or `macos-arm64.tar.gz`
macOS Intel	`macos-x86_64.tar.gz`
Linux x86_64	`.AppImage` or `.deb` or `.tar.gz`

> **macOS note**: The first time you open the app, macOS may show a security warning because the binary is not signed. See macOS installation and security for step-by-step instructions to bypass this.

Building from source

Requires Go 1.25+ and the SDL3 libraries (`sdl3`, `sdl3_image`, `sdl3_ttf`):

```
git clone https://github.com/chrplr/gostim2
cd gostim2
make build          # builds gostim2 and gostim2-gui for the current platform
make install        # copies binaries to $GOPATH/bin
```

The input file: TSV/CSV format

The entire experiment is described in a single **tab-separated values (TSV)** or **comma-separated values (CSV)** file. You can create and edit this file in any spreadsheet program (LibreOffice Calc, Excel) or a plain-text editor.

The delimiter is detected automatically (tab, comma, semicolon, or pipe).

Required columns

Your file **must** contain these four column headers (case-insensitive):

Column	Type	Description
`onset_time`	integer (ms)	When the stimulus should appear, in milliseconds from experiment start
`duration`	integer (ms)	How long to display/play the stimulus (see per-type notes below)

Column	Type	Description
`type`	string	Stimulus type — one of the values described in the next section
`stimuli`	string	The content: a filename, a text string, or a stream specification

Extra columns

Any additional columns (e.g. `condition`, `trial_id`, `expected_response`) are ignored during playback but are **preserved verbatim in the output log**, making it easy to merge logs with your design matrix.

Minimal example

```
onset_time  duration    type    stimuli
1000    500 IMAGE   face01.png
2000    300 TEXT    Hello, world!
4000    2000    BOX Please press\nany key
7000    1   SOUND   beep.wav
```

Important notes on timing

- All times are in **milliseconds**.
- `onset_time` is measured from the moment the participant presses the key to start the experiment.
- Stimuli must not overlap in time. The validator will reject a file with overlapping onsets.
- For `SOUND` stimuli, set `duration` to `1` (or any positive integer) — the sound plays to its natural end regardless of this value.

Asset directories

By default, Gostim2 looks for media files (images, sounds) in a folder named `stimuli/` or `assets/` located **next to the CSV file**. You can override this with the `-stimuli-dir` / `-assets` flag (CLI) or the corresponding field in the GUI.

A recommended project layout:

```
my_experiment/
  run1.csv
  stimuli/
    face01.png
    face02.png
    beep.wav
  gostim2-results/     (created automatically)
```

Stimulus types

IMAGE

Displays a single image file for the specified duration.

```
type      stimuli
IMAGE     face01.png
```

Supported formats: PNG, JPEG, BMP, GIF (first frame only), and other formats supported by SDL3_image.

SOUND

Plays a single audio file. The sound plays to its **natural end** regardless of the `duration` value; set `duration` to `1` as a placeholder.

```
onset_time   duration   type    stimuli
7000         1          SOUND   beep.wav
```

Supported formats: **WAV**, **FLAC**, **MP3**, **OGG**. Audio is automatically resampled if the sample rate does not match the device output rate.

TEXT

Renders a single line of text centered on the screen.

```
type   stimuli
TEXT   Fixation cross coming up...
```

Font, size, text colour, and background colour are all configurable (see Display options).

BOX

Renders **multiline text** centered on the screen. Use the two-character sequence `\n` within the `stimuli` string to insert a line break.

```
type   stimuli
BOX    Please press\nthe SPACE bar\nto continue
```

Renders as:

```
Please press
the SPACE bar
to continue
```

IMAGE_STREAM

Plays a rapid sequence of images, one after another, within a single row of the CSV. Images are separated by `~` in the `stimuli` column.

Simple form — all frames use the same duration from the `duration` column, with no inter-frame gap:

```
onset_time   duration   type           stimuli
2000         200        IMAGE_STREAM   face01.png~face02.png~face03.png
```

Timed form — each frame specifies its own `duration` and `gap` (blank screen between frames), using the format `filename:duration:gap`:

```
onset_time   duration   type           stimuli
2000         1          IMAGE_STREAM   face01.png:200:50~face02.png:200:50~face03.png:200:0
```

Here each face is shown for 200 ms followed by a 50 ms blank screen, except the last which has no trailing gap.

TEXT_STREAM

Rapid sequence of text strings. Uses the same `~` separator and optional `:duration:gap` timing as `IMAGE_STREAM`.

```
onset_time   duration   type          stimuli
5000         300        TEXT_STREAM   The~cat~sat~on~the~mat
```

Each word is displayed for 300 ms (from the `duration` column) with no gap.

With explicit timing:

```
onset_time   duration   type          stimuli
5000         1          TEXT_STREAM   The:300:50~cat:300:50~sat:300:50
```

SOUND_STREAM

Plays a sequence of audio files. Uses the same `~` separator and `:duration:gap` format. For audio streams, `duration` acts as the **SOA (Stimulus Onset Asynchrony)** — the time between the start of one sound and the start of the next.

```
onset_time   duration   type           stimuli
3000         500        SOUND_STREAM   word1.wav~word2.wav~word3.wav
```

Each sound starts 500 ms after the previous one (regardless of the sound's actual length).

With explicit per-item timing:

```
onset_time   duration   type           stimuli
3000         1          SOUND_STREAM   word1.wav:500:0~word2.wav:700:0~word3.wav:500:0
```

VIDEO

Plays a video file. If `duration` is set to `1`, the video plays to its natural end; set a value in ms to cut it off early.

```
onset_time   duration   type    stimuli
10000        1          VIDEO   clip.mp4
```

Running the GUI (`gostim2-gui`)

Launch `gostim2-gui` (double-click the icon, or run it from the terminal). A setup window appears:

1. **Experiment CSV** — click **"..."** to browse for your `.csv` or `.tsv` file.
2. **Stimuli directory** — usually auto-detected as the `stimuli/` or `assets/` folder next to the CSV; override if needed.
3. **Subject ID** — enter a participant identifier (always starts empty, never pre-filled).
4. **Results directory** — where output logs are saved (default: `gostim2-results/` next to the CSV).
5. **Resolution** — pick a fixed resolution or check **Autodetect resolution** to use your monitor's native resolution.
6. **Window mode**:
   - *Windowed*: standard window.
   - *Fullscreen Desktop*: borderless window at desktop resolution; OS compositor remains active. Safest option.
   - *Fullscreen Exclusive*: takes exclusive control of the display, bypasses the compositor. Lowest latency; recommended for EEG/MEG/fMRI.
7. **Font / font size**, **colours** (background, text, fixation cross).
8. **VSync** — enabled by default; disable only for VRR/G-Sync setups.
9. Click the green **START** button.

When the experiment window opens, press any key to begin (unless **Skip wait** is checked). Press **Escape** at any time to abort.

Settings are automatically saved to `gostim2_config.toml` in the same directory, and reloaded at the next launch.

Running the CLI (`gostim2`)

```
gostim2 -csv experiment.csv [options]
```

Required argument

Flag	Description
`-csv <path>`	Path to the stimulus CSV/TSV file. `-tsv` is an alias.

File and directory options

Flag	Default	Description
`-stimuli-dir <path>`	auto-detected	Directory containing image/sound assets. `-assets` is an alias.
`-results-dir <path>`	`gostim2-results`	Directory for output log files (resolved relative to the CSV file).
`-subject <id>`	`unknown`	Participant identifier, included in the output filename and log.
`-config <path>`	—	Load parameters from a TOML config file (same format as `gostim2_config.toml`).
`-start-splash <path>`	—	Image to display before the experiment (waits for key press).
`-end-splash <path>`	—	Image to display after the experiment (waits for key press).

Display options

Flag	Default	Description
`-res <WxH | Autodetect>`	—	Screen resolution, e.g. `1920x1080`. Use `Autodetect` for native resolution.
`-width <int>`	`1440`	Screen width (ignored if `-res` is set).
`-height <int>`	`900`	Screen height (ignored if `-res` is set).
`-display <int>`	`0`	Index of the monitor to use.
`-scale <float>`	`1.0`	Logical scale factor applied to all stimuli.
`-fullscreen`	off	Exclusive fullscreen (bypasses compositor).
`-fullscreen-desktop`	off	Borderless fullscreen (compositor active).
`-bg-color <R,G,B,A>`	`0,0,0,255`	Background colour as RGBA values 0–255.
`-text-color <R,G,B,A>`	`255,255,255,255`	Text colour as RGBA values 0–255.
`-fixation-color <R,G,B,A>`	`255,255,255,255`	Fixation cross colour.

Font options

Flag	Default	Description
`-font <path>`	built-in Inconsolata	Path to a TTF font file.
`-font-size <int>`	50	Font size in points.

Fixation cross options

Flag	Default	Description
(default)		Fixation cross shown only on blank screens.
`-no-fixation`		Never display the fixation cross.
`-fixation-always`		Always display the fixation cross (superimposed on stimuli).

Timing options

Flag	Default	Description
`-vsync`	on	Synchronize rendering to the display refresh rate.
`-no-vsync`		Disable VSync.
`-vrr`	off	Variable Refresh Rate mode: disables VSync and uses busy-wait for ~2 ms precision. For G-Sync/FreeSync monitors.

Other options

Flag	Description
`-skip-wait`	Skip the “Press any key to start” prompt.
`-dlp <device>`	Serial device path for DLP-IO8-G hardware triggers (e.g. `/dev/ttyUSB0`).
`-version`	Print version information and exit.

Example invocations

```
# Basic run
gostim2 -csv study1/run1.csv -subject sub-01

# Fullscreen exclusive at 1920×1080, custom font
gostim2 -csv run1.csv -fullscreen -res 1920x1080 -font ~/fonts/DejaVuSans.ttf -subject 42
```

```
# Load most settings from a config file, override subject only
gostim2 -config my_experiment.toml -subject sub-07

# Run from a TTY console (Linux) for minimal latency
gostim2 -csv run1.csv -fullscreen -subject sub-01
```

Output logs

After each run, Gostim2 writes a timestamped TSV file to the results directory. The filename follows the pattern:

`{csv_basename}_sub-{subject_id}_{YYYY-MM-DDTHH-MM-SS}.tsv`

Log structure

The file starts with a **metadata header** (lines beginning with `#`) recording system information: SDL version, audio/video drivers, platform, display mode, and the exact flags used.

This is followed by a data table with these columns:

Column	Description
`subject_id`	The participant identifier from `-subject` or the GUI field.
`intended_ms`	The onset time as specified in the CSV.
`actual_ms`	The real onset time measured at runtime.
`type`	Event type: `IMAGE_ONSET`, `SOUND_ONSET`, `TEXT_ONSET`, etc., or `RESPONSE`.
`label`	The stimulus filename or text string.
extra columns	All extra columns from the original CSV are reproduced here.

Results directory resolution

The results path is always resolved **relative to the CSV file**, not to the working directory:

Command	Results saved in
`gostim2 -csv study1/run1.csv`	`study1/gostim2-results/`
`gostim2 -csv study1/run1.csv -results-dir data`	`study1/data/`
`gostim2 -csv study1/run1.csv -results-dir ""`	`study1/`

Advanced topics

Linux: maximum timing precision

On Linux, you can achieve the lowest possible latency by running Gostim2 from a bare TTY console (Ctrl+Alt+F3), after stopping the display manager:

```
sudo systemctl stop gdm   # or lightdm, sddm, etc.
gostim2 -csv run1.csv -fullscreen -subject sub-01
```

This allows SDL3 to use the **Direct Rendering Manager (DRM)** directly, bypassing Wayland/X11 entirely.

Hardware triggers (DLP-IO8-G)

For EEG/MEG setups requiring TTL triggers, connect a DLP-IO8-G USB device and pass its serial port:

```
gostim2 -csv run1.csv -dlp /dev/ttyUSB0 -subject sub-01
```

On Windows the port is typically `COM3` or similar.

Config file (TOML)

The GUI automatically writes `gostim2_config.toml` after each session. You can hand-edit this file or create one from scratch and pass it with `-config`:

```
CsvFile        = "run1.csv"
StimuliDir     = "stimuli"
ResultsDir     = "gostim2-results"
FullscreenMode = "exclusive"
Resolution     = "1920x1080"
FontSize       = 60
BgColor        = [0, 0, 0, 255]
TextColor      = [255, 255, 255, 255]
```

Quick reference: stimulus type summary

Type	Content of `stimuli` column	`duration`
`IMAGE`	Single image filename	Display time (ms)
`SOUND`	Single audio filename	Use `1` (plays to end)
`TEXT`	Single line of text	Display time (ms)
`BOX`	Multiline text; `\n` for line breaks	Display time (ms)
`IMAGE_STREAM`	`img1.png~img2.png` or `img1.png:dur:gap~...`	Default frame duration if no per-item timing
`TEXT_STREAM`	`word1~word2` or `word1:dur:gap~...`	Default frame duration if no per-item timing
`SOUND_STREAM`	`snd1.wav~snd2.wav` or `snd1.wav:soa:gap~...`	Default SOA if no per-item timing
`VIDEO`	Single video filename	`1` to play to end, or cut-off time (ms)

License and credits

Gostim2 is a port of audiovis to Go, using go-sdl3 bindings.

Author: Christophe Pallier christophe@pallier.org

Source code: https://github.com/chrplr/gostim2