

EyeLink Python Example Scripts

Last Updated: June 19, 2026 | Support: support@sr-research.com

The examples bundled in this directory demonstrate how to use the PyLink library to interface EyeLink eye trackers natively within Python environments, including PsychoPy, Pygame, and OpenSesame. These scripts handle tracker connection, parameter configuration, recording, and real-time data retrieval.

Installation, Setup, & Core API Documentation

To use these example scripts, the **pylink** library must be installed. The installation steps for PsychoPy and OpenSesame include the **pylink** dependency within their installation.

(Note: Offline installation wheel files (.whl) for PyLink and **pygame_eyelink_coregraphics** are provided in the **Offline_Install** directory for various Python builds (from Python 2.7 **cp27** to Python 3.14 **cp314**) and operating systems. For further installation guidance, please refer to the SR Support Forum at <https://www.sr-research.com/support/thread-13.html>).

- **Pylink API User Guide:** The core library documentation is available as **Pylink api userguide.pdf** and **pylink.chm** in the root of this directory.

Example Scripts by Environment

1. PsychoPy (**Psychopy_examples**)

For general instructions on installing and configuring PsychoPy for use with EyeLink systems, please see the SR Support Forum: [Getting Started with Psychopy](#).

A. Builder - Code Components (**Builder_CodeComponent**)

These visual templates use PsychoPy Custom Code Components to manually script the PyLink integration within the Builder interface. (For a detailed walkthrough, refer to the **UsingPyLinkFromPsychoPyBuilder.pdf** guide located in this folder).

- **EyeLinkPicture_Builder:** Initializes the tracker, controls recording, populates the Host PC backdrop, and sends standard Data Viewer integration messages.
- **EyeLinkPursuit_Builder:** Records dynamic target positions for a smooth pursuit task, sending dynamic interest area data to the **.edf** file.
- **EyeLinkSaccade_Builder:** A pro-saccade and anti-saccade task demonstrating real-time eye event retrieval.
- **EyeLinkFixationWindowFastSamples_Builder:** A gaze-based trigger where a trial advances only when the participant directs their gaze to a central fixation cross.
- **EyeLinkGCWindow_Builder:** A gaze-contingent display that uses real-time gaze data to mask an image or reveal a moving window over an image.
- **EyeLinkMRIdemo_Builder:** Demonstrates continuous recording across a block of trials, synchronized with an external MRI pulse, using trial onset/offset markers for segmentation.

- **EyeLinkVideo_Builder**: Records gaze over video stimuli while logging specific frame information to allow video playback in Data Viewer.

B. Builder - EyeLink Plugins (**Builder_EyeLinkPlugin**)

These visual templates achieve the same paradigms as above but utilize the official EyeLink PsychoPy Plugin components for a more streamlined, GUI-driven integration.

- **EyeLinkPicture_Plugin**: Static image presentation and standard Data Viewer integration using the plugin.
- **EyeLinkPursuit_Plugin**: Smooth pursuit task logging dynamic interest area data using the plugin.
- **EyeLinkSaccade_Plugin**: Pro-saccade and anti-saccade task using the plugin.
- **EyeLinkFixationWindowFastSamples_Plugin**: Gaze-contingent fixation trigger using the plugin.
- **EyeLinkGCWindow_Plugin**: Real-time gaze-contingent window/masking using the plugin.
- **EyeLinkMRIdemo_Plugin**: Continuous recording and external MRI synchronization using the plugin.
- **EyeLinkVideo_Plugin**: Video playback with frame logging using the plugin.

C. Coder (**Coder**)

Hand-coded scripts utilizing the PsychoPy library for stimulus presentation.

- **picture**: A static image presentation example demonstrating how to open/close data files, start/stop recording, and format standard integration messages.
- **pursuit**: A dynamic smooth pursuit example that logs moving target and dynamic interest area information to the **.edf**.
- **saccade**: Saccadic event retrieval example for pro-saccade and anti-saccade paradigms.
- **fixationWindow_fastSamples**: Gaze-contingent fixation trigger utilizing real-time sample retrieval.
- **GC_window**: Continuous gaze-contingent visual masking using real-time spatial updates.
- **MRI_demo**: Continuous long-form recording setup synchronized with external MRI triggers.
- **video**: Presents video files and logs frame data for accurate overlay in Data Viewer.

2. Python and Pygame (**Python_Pygame**)

For instructions on using PyLink in standard Python or Pygame environments, see the SR Support Forum [Getting Started with Python and PyLink](#) or the [Getting started with Python and PyLink.pdf](#) located in this folder.

A. Pure Link Examples (**Link_Examples**)

These scripts do not require graphical libraries like Pygame or PsychoPy to run.

- **link_event**: Demonstrates connecting to the tracker, configuring parameters, and retrieving eye events (e.g., Fixation Start/End, Saccade Start/End) in real-time.
- **link_sample**: Demonstrates how to retrieve real-time samples (timestamped gaze position, pupil size, etc.) during active data recording.

B. Pygame Examples (**Pygame**)

Hand-coded scripts utilizing the Pygame library for stimulus presentation.

- **picture**: A static image presentation example demonstrating tracker connection, recording control, and Data Viewer integration messaging.
 - **pursuit**: A dynamic smooth pursuit example that logs moving target and dynamic interest area information.
 - **saccade**: Saccadic event retrieval example for pro-saccade and anti-saccade paradigms.
 - **fixationWindow_fastSamples**: Gaze-contingent fixation trigger utilizing real-time sample retrieval.
 - **GC_window**: Continuous gaze-contingent visual masking using real-time spatial updates.
 - **MRI_demo**: Continuous long-form recording setup synchronized with external MRI triggers.
-

3. OpenSesame ([OpenSesame](#))

- Refer to [README_OpenSesame.txt](#) located within the directory for specific setup instructions. Current plugins are built for OpenSesame 4.
 - For further instructions on installing and configuring OpenSesame for use with EyeLink systems, please see the SR Support Forum: [Getting Started with OpenSesame](#).
-

EyeLink Data Viewer Integration

Beyond marking event times, these examples demonstrate how to send specially formatted messages to your EyeLink Data File (.edf) to include integration with Data Viewer. These messages allow Data Viewer to automatically parse your experiment's structure, including trial start/end times, trial variables, interest areas, and other stimulus information.