# Auto SIMPLE demo

This demo shows a proof-of-concept video processing-based data acquisition workflow for (i)SIMPLE experiments.

As of now, data acquisition for this project was performed 'manually', i.e. by direct timing annotation from video footage. However, this is not only very time-consuming, but also makes collaboration and verification difficult. Moreover, as (i)SIMPLE chip designs become more intricate, it becomes impossible to include all of the required channel markings on the chip.

Compared to other paper-based microfluidics research, even our 'simple' (i)SIMPLE chip designs are rather complex, and our video capturing setup is rather low-tech. This is how we used to justify our 'manual' approach; it would take too much effort to actually design a image processing pipeline which could support our raw data. To tackle this issue, this demo leverages the fact that we can use our design files to align the raw video footage with the chip, and precisely mask off the parts we're interested in.

### Basic functionality

The demo program handles a single video (as far as I know, almost any video file type should work) and a corresponding design (as an .svg file). The program produces a real-time plot (which can be saved) and .csv output which can be used further.

A Python virtual environment is provided with this demo. As far as I know, this should work out-of-the-box on a windows system, but if that is not the case I can help out. if you want to run this on a different OS you would need to install Python yourself. Please contact me with any questions.

### Installation (Windows)

* Activate local admin
* Extract the .zip file somewhere
* Install Python with python-3.7.1.exe

**Please make sure you select the option to “add Python to PATH”!**

* Open start.bat. This will open a Command Prompt window and navigate it to the demo folder.

Type the following instructions in the command line.

* Create and launch a virtual environment

py –m venv \_py

ve

* Install the required packages

pip install –r requirements.txt

* Move cairo.dll to C:\Windows\System32

### Usage

* Open start.bat. This will open a Command Prompt window and navigate it to the demo folder.

Type the following instructions in the command line.

* Launch the Python virtual environment.

ve

* Launch the demo script.

demo.py shuttle\_video.mp4 shuttle\_design.svg

You can also provide the time interval (in s, default is 5 s) and the height of the channel (in mm, default is 0.153 mm):

demo.py shuttle\_video.mp4 shuttle\_design.svg -dt 10 -hc 0.127

A series of windows will pop up to setup your measurement. Press Esc to move to the next one once you're ready.

* Select the chip from your video by dragging a rectangle around it. On the right you'll see your selected region overlaid with the design. You can reposition the corners of the selection individually to improve the fit. A good fit is important if you want consistent results! You can clear the selection by pressing Ctrl+Z.
* For every masking layer in your design file, a new window will open with the masked off region on the left and the detected liquid on the right. To make sure that you're measuring the right liquid you have to assign the correct colour to filter out by on clicking the left image. If the liquid is not present on the frame right away you can scan through the video with the scrollbar on the bottom until you find it. The default colour is blue.
* Once you've selected all of the filtering colours, the measurement will start. It can take a while to finish depending on the time interval you choose (and of course on your computer). The results of your measurement will be written into shuttle\_video.csv. If you start a new measurement from the same video, this file will be overwritten, so beware!

Of course, you can run the demo with any other video and/or design files. Just replace the names of the example files with those of your own files in the command above in step 4. If you place your files in the demo folder it's enough to just write the file names, otherwise you should fill in the full path to your files.

### Preparing design files

If you want to check out this script with your own data, you will need to make your own design file. This file should contain:

* An overlay layer (named **overlay**, no caps).

This layer will be used to align the video footage to the design, so it's probably easiest to include the full PSA cutting design as solid lines without any colour.

* All of the regions you want to measure in separate layers.

These layers should consist of a single continuous shape representing the channel (or portion of the channel). The name of this layer will be used in the legend of the output graph and in the output file. If you want to ensure a specific order for these layers, you can format the layer names as "1 - ..." (without the quotes).

* A solid white background layer **(\_background**).

This layer should be included to make sure that there is no transparency in the final .png render, as this could mess with the masking.

To make sure that this background layer covers the entire image, it is best to draw it with respect to the page instead of the design itself:

* 1. Select the overlay layer
  2. Resize the page to the selected layer by clicking File > Document Properties > Resize page to content… > Resize page to drawing or selection
  3. Turn on “Show page border”
  4. Enable “Snap to page border” in the snap controls bar
  5. In the “\_background” layer, draw a rectangle over the page border (solid white fill, no stroke**, no transparency**)

You can go by the example overlay file if anything is unclear.