**Guide to Arena Automation Software & Hardware**

**[WORK IN PROGRESS!]**

|  |  |
| --- | --- |
| **Author:** | Abel Corver |
| **For questions email:** | abel.corver@gmail.com |
| **Last updated:** | 18 December 2016 |

|  |
| --- |
| This guide summarizes how to use the suite of hardware and software tools that collectively facilitate the automatic, unsupervised recording and analysis of dragonfly activity during whole-day recording sessions --- henceforth referred to as the *Arena Automation System*. |

Table of Contents

[Where to find the project files 2](#_Toc469673817)

[A note on the technology stack used 2](#_Toc469673818)

[Running the data acquisition & automatic triggering software 2](#_Toc469673819)

# Where to find the project files

All project files are maintained in a single directory. The current location of this directory is:

|  |
| --- |
| Z:/people/Abel/arena-automation/ |

A complete version history of the project is maintained at:

|  |
| --- |
| https://github.com/acorver/arena-automation |

Currently, accessing the project requires a GitHub account with access privileges --- email Abel Corver for access. Ideally, future contributing developers to the project are encouraged to create a GitHub account and commit their changes to the online repository, thus maintaining a full version history.

# A note on the technology stack used

All real-time processing and interfacing with other systems is done in C/C++. Post-processing is done using Python scripts. Initial, interactive exploration of data analysis approaches is often done in R.

# Running the data acquisition & automatic triggering software

Go through the following steps:

**Step 1.** Turn on the power. The power can be controlled remotely using electronically switched relays, which are connected via USB cable to a host computer --- currently the PC running Photron. There are several options for controlling the power:

**Option 1**: Unplug the microcontrollers that control the relays. It’s best to unplug the end of the USB cable that goes into the computer, rather than the end that connects to the microcontroller, as the former is easier to remove.

**Option 2**: Start the Arduino software. In the *Tools* menu, click *Port*, and select the COM port controlling each relay --- see below for the current port mappings. Once a port is selected, open the Serial Monitor (Ctrl-Shift-M), and type “+” (without the quotes) to turn on all relays. Type “-“ to turn all relays off.

|  |
| --- |
| COM 12: Big relay board with five relays, currently controlling all cameras.  COM 14: Smaller relay board with two relays, currently controlling motor and telemetry power |

**Option 3:** Use the Arena Automation Web Interface to turn all power on and off. This is currently a work-in-progress and too unstable for use.

**Step 2.** Calibrate the room and start Cortex. Cortex needs to be running before we can start the Arena Automation Software. In rare cases where Cortex crashes during a recording session, both Cortex and the Arena Automation Software have to be restarted.

**Step 3.** Make sure the *FlySim Arduino* as well as the *Teensy TTL Trigger* are connected (via USB cable) to the computer that will run the Arena Automation Software --- currently the computer running Photron.

**Step 4.** Confirm the settings are correct. The settings file is loaded once, when the Arena Automation software starts. For changes in settings to take effect, the software (See Step 4) must be restarted. The settings file can be found here:

|  |
| --- |
| <arena automation root>/deploy/settings/settings.json |

The software will always look for a file with that name. Therefore, if one wants to make temporary changes to settings, one can save a copy of the original *settings.json* file in the same directory and change the original.

The current most stable motion tracking algorithm is activated in the settings file as follows:

|  |
| --- |
| "method": "velocity" |

To change takeoff threshold, change the following settings:

|  |
| --- |
| "takeoff\_detection\_window": 10,  "takeoff\_speed\_threshold": 40 |

However, it is unlikely these settings should be changed in most circumstances. The current settings balance false positives and consistent trigger delays of ~200 milliseconds.

**Step 5.** Start the Photron PFV viewer. Currently this program is run on a different PC than that running Cortex. The Photron PC should be the one connected to the TTL Trigger USB interface. Photron should be configured to respond to TTLs and automatically continue recording afterwards.

[TODO: Add more details on this configuration here.]

**Step 6.** Now execute the following file on the Photron PC:

|  |
| --- |
| <arena automation root>/deploy/ArenaAutomation\_161215.exe |

**Step 7.** Open the Windows task manager, right click the *ArenaAutomation.exe* process, and change the process priority to “Realtime”. This is important for consistent performance!

**Step 8.** Now run the following file on the Photron PC:

|  |
| --- |
| <arena automation root>/web-interface/Arena Interface.bat |

After about 10-30 seconds, the following webpage will become available:

|  |
| --- |
| http://10.101.30.47/ |

On the Photron computer on which “Arena Interface.bat” was run, this page is also available as:

|  |
| --- |
| http://localhost/ |

This page will display a log of ongoing activity.

**Step 9.** Make sure that the log contains the following information:

* Successful connection with FlySim interface
* Successful connection with Teensy TTL Trigger interface
* Successful connection with Cortex

If there were problems connecting to any of these interfaces, restart or reconnect the relevant interface. Currently, connections to FlySim and the TTL Trigger interface can be re-established without restarting the program. However, re-establishing a connection with Cortex requires restarting the *ArenaAutomation.exe* process (Step 6).

**Step 10.** The main output file has the following location and filename:

|  |
| --- |
| <arena automation root>/deploy/data/<date time>.msgpack |

To confirm that the program is working and frames are being recorded, it is helpful to confirm that this file is growing in size. It is normal for this file to grow to anywhere from hundreds of megabytes to a few gigabytes over the course of a day’s recording.