

# plc - PlasmaLabControl

**Version:** 2013-03-14  
**Author:** Daniel Mohr  
**Email:** [mohr@mpe.mpg.de](mailto:mohr@mpe.mpg.de)

## Contents

<b>1</b>	<b>Manual of plc - PlasmaLabControl</b>	<b>2</b>
1.1	Device Names	3
1.2	Config File	3
1.2.1	Config File: ini	4
1.2.2	Config File: camera*	4
1.2.3	Config File: controller 'dc' or 'mpc'	5
1.2.4	Config File: interprocess communication	5
1.2.5	Config File: dispenser*	5
1.2.6	Config File: RF-Generator *	6
1.2.7	Config File: "electrode motion controller" and "translation stage controller"	6
1.3	Keyboard Control	6
1.4	Cameras	6
1.4.1	Cameras: shutter	7
1.4.2	Cameras: additionally commands	7
1.4.3	Cameras: create histogram	7
1.4.4	Cameras: brightness	7
1.5	Controller, Port, Channel	7
1.5.1	Digital Controller (red box)	8
1.5.2	Multi Purpose Controller (blue box)	8
1.5.3	Gas System: Mass Flow Controller	8
1.6	RF-Generator	8
1.6.1	RF-Generator: Combined Changes	8
1.6.2	RF-Generator: ignite plasma	8
1.6.3	RF-Generator: pattern	9
1.7	Diagnostics/Particles: Laser	10
1.8	Electrode Motion:	10
1.9	Setpoints:	10
1.10	Logging:	11
<b>2</b>	<b>Consistency of Time</b>	<b>12</b>
<b>3</b>	<b>Other Scripts</b>	<b>15</b>

3.1	<a href="#">plc_viewer.py</a>	15
3.2	<a href="#">rawmovies2recordings.py</a>	16
3.3	<a href="#">rawmovieviewer.py</a>	16
3.4	<a href="#">digital_controller_server.py</a>	17
3.5	<a href="#">digital_controller_client.py</a>	19
3.6	<a href="#">multi_purpose_controller_server.py</a>	19
3.7	<a href="#">multi_purpose_controller_client.py</a>	21
3.8	<a href="#">camera_server.py</a>	21
3.9	<a href="#">camera_client.py</a>	22
3.10	<a href="#">acceleration_sensor_logger.py</a>	23
3.11	<a href="#">acceleration_sensor_server.py</a>	24
3.12	<a href="#">acceleration_sensor_client.py</a>	25
3.13	<a href="#">pressure_mks_651_client.py</a>	26
3.14	<a href="#">pressure_mks_651_server.py</a>	26
3.15	<a href="#">pressure_mks_900_client.py</a>	27
3.16	<a href="#">pressure_mks_900_server.py</a>	28
3.17	<a href="#">check_real_time_difference_server.py</a>	29
3.18	<a href="#">check_real_time_difference_client.py</a>	29
3.19	<a href="#">translation_stage_scan.py</a>	30
3.20	<a href="#">environment_sensor_5_logger.py</a>	31
<b>4</b>	<b>Extern Bugs</b>	<b>32</b>

# 1 Manual of plc - PlasmaLabControl

This package "plc" provides the python module

- `plc_gui`

and the scripts

- `plc.py`
- `plc_viewer.py`
- `debug_controller.py`
- and some other ones in this development status.

Normally it should be enough for you to use `plc.py` for controlling the experiment. On your console you can start:

```
plc.py -h
```

You get the help output:

## plc - PlasmaLabControl

```
usage: plc.py [-h] [-debug debug_level] [-system_config file] [-config file]

plc - PlasmaLabControl. For more help
type "pydoc plc"

optional arguments:
  -h, --help                show this help message and exit
  -debug debug_level        Set debug level. 0 no debug info (default); 1 debug to
                             STDOUT.
  -system_config file       Set system wide config file to use. This will be read
                             first. (default: '/etc/plc.cfg')
  -config file              Set user config file to use. This will be read after
                             the system wide config file. (default: '~/plc.cfg')
```

Author: Daniel Mohr  
Date: 2012-09-10  
License:

## 1.1 Device Names

To get nice device names and every time the same devices you can use udev. For example you can wrote the following rules:

```
# udevadm info --query=all --name=/dev/ttyUSB3
#
# udevadm info --attribute-walk --name=/dev/ttyUSB1
#
# udevadm test -a -p $(udevadm info -q path -n /dev/ttyUSB1)
#
# udevadm trigger --action=add --sysname-match=ttyUSB1

# JoyWarrior24Fl4
SUBSYSTEM=="usb", ATTRS{idVendor}=="07c0", ATTRS{idProduct}=="1116", MODE=="666", GROUP="dialout"

#####
# Zyflex #
#####
# digital controller
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="0001", SYMLINK+="DOCUs{serial}", GROUP="dialout"
# multi purpose controller
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="0006", SYMLINK+="MPC{s{serial}}", GROUP="dialout"
# electrode motion controller
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="ZPCU0001", SYMLINK+="EMCs{serial}", GROUP="dialout"
# translation stage controller
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="6001", ATTRS{idVendor}=="0403", ATTRS{serial}=="ftDXPBD0", SYMLINK+="TSC{s{serial}}", GROUP="dialout"
# RF-Generator
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="RF_GEN_02", SYMLINK+="s{serial}", GROUP="dialout"
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="RF_DC_02", SYMLINK+="s{serial}", GROUP="dialout"

#####
# Dodecahedron #
#####
# digital controller
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="0002", SYMLINK+="DOCUs{serial}", GROUP="dialout"
# multi purpose controller
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="MIO_0001", SYMLINK+="MPC{s{serial}}", GROUP="dialout"

# RF-Generator
# Master
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="RF_GEN_01_Dod_mstr", SYMLINK+="s{serial}", GROUP="dialout"
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="RF_DC_01_Dod_mstr", SYMLINK+="s{serial}", GROUP="dialout"
# Slave 1
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="RF_GEN_02_Dod_slv", SYMLINK+="s{serial}", GROUP="dialout"
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="RF_DC_02_Dod_slv", SYMLINK+="s{serial}", GROUP="dialout"
# Slave 2
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="RF_DC_03_Dod_slv", SYMLINK+="s{serial}", GROUP="dialout"
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="RF_GEN_03_Dod_slv", SYMLINK+="s{serial}", GROUP="dialout"

#####
#Zyflex Optical Table#
#####
# digital controller
ACTION=="add", KERNEL=="ttyUSB*", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="DIO_0004", SYMLINK+="DOCUs{serial}", GROUP="dialout"

# multi purpose controller
ACTION=="add", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="MIO_0005", SYMLINK+="MPC{s{serial}}", GROUP="dialout"
# electrode motion controller
ACTION=="add", SUBSYSTEM=="tty", ATTRS{idProduct}=="ea60", ATTRS{idVendor}=="10c4", ATTRS{serial}=="ZPCU0001", SYMLINK+="EMCs{serial}", GROUP="dialout"
# translation stage controller
ACTION=="add", SUBSYSTEM=="tty", ATTRS{idProduct}=="6001", ATTRS{idVendor}=="0403", ATTRS{serial}=="ftBNKEKX", SYMLINK+="TSC{s{serial}}", GROUP="dialout"

ACTION=="add", SUBSYSTEM=="tty", ATTRS{idVendor}=="0403",ATTRS{idProduct}=="6001",ATTRS{serial}=="ftDX05JW",SYMLINK+="MKS_P900_1",GROUP="dialout"
```

Do not call these rules to early. For example you can put them in `"/etc/udev/rules.d/99-plc.rules"`.

## 1.2 Config File

The config file(s) is/are parsed by <http://docs.python.org/library/configparser.html> . So we have values in sections.

The default config files are "/etc/plc.cfg" and "~/.plc.cfg". This can be modified by command line switches --- see the output from "plc.py -h". First the global config file is parsed and then the user one, which could overwrite the global config. Both overwrite the hard coded default configs --- as usual.

You can get the default config file from the pull down menu of `plc.py`. The section names and the value names should be meaningful.

In the following a few hints to the named sections are given.

### 1.2.1 Config File: ini

The intervals are in milliseconds.

### 1.2.2 Config File: camera\*

You can make a section "cameraI" for every camera I. From the software there is no limit. "I" is here a natural number starting from 1.

It is possible to set some default parameters; e. g.:

```
[camera1]
guid: 2892819639808492
mode: FORMAT7_0
color_coding: Y8
framerate: 30
camera_file_prefix1 = /tmp/cam_$guid_$date
camera_file_prefix2 = /tmp/cam_$guid_$date

[camera2]
guid = -1
mode = FORMAT7_0
color_coding = Y8
framerate = 30
camera_file_prefix1 = /tmp/cam_$guid_$date

[camera3]
guid = -2
camera_file_prefix1 = /tmp/cam_$guid_$date
```

For the default setting look at the default config.

The guid is the guid of the camera. You can set it as an integer "2892819639808492" or as a hex number "0x000a47010f07b1ec". A value of "-1" means no guid preselected; therefore many other settings are not possible. A value of "-2" means this camera (and possible further ones) are not available; this is the same as do not make this section. But keep in mind, you must overwrite the default setting and therefore you should overwrite the sections "camera1" and "camera2". The default section "camera2" is not harmful.

The variables "mode", "color\_coding" and "framerate" can be disabled by a value of "-1".

The other possible variables "brightness", "trigger\_delay", etc. can be disabled by the value "default".

The default camera\_file\_prefix is "/tmp/cam\_\$guid\_\$date". The \$guid variable will be replaced by the guid of the camera and the \$date by the actual date; e. g. you get "/tmp/cam\_2892819639808492\_2012-08-29\_\*.img". An empty string will set nothing.

**ATTENTION:** If you use the button "get camlist" all firewire ports will be used for a short time by the server. If another camera server is running, it will get a problem! If you don't use it and configure the camera in the config file, the options could be wrong in the gui! At the moment you must decide, what you want here.

If you use two or more unspecified camera tabs in plc.py, you should first start both camera servers and connect to them. Then you should "get camlist" in both tabs (for both cameras/servers) before you can make any other settings or use one camera. Otherwise you the later used camera server gets a problem!

If you have a camera with 60 frames per second, you get every 16666 microseconds a new frame. Therefore a shutter speed longer than 16666 microseconds is not reasonable.

### 1.2.3 Config File: controller 'dc' or 'mpc'

value	measurement unit
update_intervall	milliseconds
complete_update_intervall	seconds

  

value	meaning
start_server	If start is requested, try to start the server program.
connect_server	If set to true, try to connect on startup. If start_server is also true, try to start the server before connecting.

### 1.2.4 Config File: interprocess communication

The interprocess communication between plc.py and the server programs (e. g. digital\_controller\_server.py) is done by socket communication. Ther server programs are listening on choosen ports for starting of communication. Look at each server to find the default ports. An overview is given in the table:

server program	default port
digital_controller_server.py	15112
multi_purpose_controller_client.py	15113
camera_server.py	15114
pressure_mks_900_server.py	15121
pressure_mks_651_server.py	15122
acceleration_sensor_server.py	15123
check_real_time_difference_server.py	15124

You can find typical used ports in

<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>

The choosen default ports are all "Unassigned". Unassigned ports are for example:

- 15001-15117
- 15119-15344

### 1.2.5 Config File: dispenser\*

Setting "key\_binding\_dispenser\*" or "controller" to "-1" will disable the corresponding function.

The times "Toff" and "Ton" are in milliseconds.

### 1.2.6 Config File: RF-Generator \*

Setting "power\_controller" to "-1" will disable this generator box.

Iff "RF\_only\_master" is set to "1" in the section "RF-Generator" only the master generator "RF-Generator 1" will get the RF-On/Off command.

### 1.2.7 Config File: "electrode motion controller" and "translation stage controller"

value	measurement unit
readtimeout	seconds
writetimeout	seconds
update_intervall	milliseconds
T_off	milliseconds

Setting "devicename" to "-1" will disable this controller.

## 1.3 Keyboard Control

A few keys for keyboard control can be set by the config file. The default setting is:

Key	Function
<KeyPress-F1>	dispenser 1
<KeyPress-F2>	dispenser 2
<KeyPress-F3>	dispenser 3
<KeyPress-F5>	camera: view all, stop recording
<KeyPress-F6>	camera: start recording
<KeyPress-F7>	previous setpoints
<KeyPress-F8>	next setpoints
<KeyPress-F9>	set selected setpoints

## 1.4 Cameras

We use only cameras compatible with the DCAM specs (see [http://en.wikipedia.org/wiki/FireWire\\_camera#Interface](http://en.wikipedia.org/wiki/FireWire_camera#Interface) or <http://www.1394ta.org/> ).

With the python wrapper pydc1394 (which you can find on <https://launchpad.net/pydc1394we> ) we use in the background the library <http://damien.douxchamps.net/ieee1394/libdc1394/> .

At the moment we have 3 different kind of camera tested:

- AVT Guppy F-080B
- AVT PIKE F-100B
- AVT PIKE F-421

If you have problems with the cameras or with our software using the cameras, you can also try coriander (see <http://damien.douxchamps.net/ieee1394/coriander/> ). This is the most famous software using libdc1394.

See also the "Config File" section for the camera.

### 1.4.1 Cameras: shutter

The meaning of the number of the shutter comes from the camera. So please read the manual of your camera.

For example for a Guppy F-080B the number x of the shutter means an integration time (or exposure time) of:

$$[\text{time base}] * x + [\text{offset}]$$

The default [time base] is 20 microseconds =  $20 \cdot 10^{-6}$  sec, but can be changed somehow. The [offset] for a Guppy F-080B is  $34 \cdot 10^{-6}$  sec. For example a shutter setting of 1000 means an integration time of 0.040034 seconds.

Here are some examples of the integration time for a number of x of the shutter for the default time base:

Camera	exposure time (sec)
PIKE F-421	$20 \cdot 10^{-6} * x + 70 \cdot 10^{-6}$
PIKE F-100B	$20 \cdot 10^{-6} * x + 43 \cdot 10^{-6}$
Guppy F-080B	$20 \cdot 10^{-6} * x + 34 \cdot 10^{-6}$

### 1.4.2 Cameras: additionally commands

In the config file in the section cameras it is possible to give additionally commands, for the global buttons "record" and "view + stop record". For example:

```
[cameras]
additionally_command_for_view_all: ssh plexp2 "killall -USR1 vd_grab_univ"
additionally_command_for_record_all: ssh plexp2 "killall -USR2 vd_grab_univ"
```

### 1.4.3 Cameras: create histogram

By choosing "create" and "histogram" you get a live histogram for every frame. It is always scaled to the highest peak; the highest peak uses the full height.

By choosing "create" and "horizontal sums" all horizontal lines of every frame are summed. These values are displayed like the histogram.

By choosing "create" and "vertical sums" all vertical columns of every frame are summed and displayed like the histogram.

### 1.4.4 Cameras: brightness

In the label "brightness" you get the relative brightness of the complete picture:

$$\left( \sum_{\text{pixel}} \text{brightness} \right) / (255 \cdot \text{width} \cdot \text{height})$$

Here 255 is the maximum value of a pixel of a picture in format "L". For other possible formats this is wrong. But until now all other cameras are producing frames/pictures of format "L".

## 1.5 Controller, Port, Channel

The physical output to the equipment (e. g. pumps, laser, ...) is done with controller boxes. Therefore in the config file you can define at which box at which port and at which channel are the voltage or measuring of special equipments.

Since the server programs `digital_controller_server.py` and `multi_purpose_controller_server.py` are only simple controllers, this leads to different representation of the values on different controller or even different ports. The understanding of theses representations is at this development state hard coded in plc.

**ATTENTION:** Changing ports or channels is only acceptable for identical types!

### 1.5.1 Digital Controller (red box)

The digital controller has following ports with these types:

- bool A[8], B[8], C[8], D[8]

### 1.5.2 Multi Purpose Controller (blue box)

The multi purpose controller has following ports with these types:

- bool DO[4], R[2] U15, U05, U24
- unsigned short int DAC[4]
- bool DI[4]
- short int ADC[8]

### 1.5.3 Gas System: Mass Flow Controller

The mass flow controller accepts voltages between 0 V and 5 V, which corresponds to flows of 0 sccm and 1 sccm.

The input in plc is done in msccm. For output to the controller this is translated to voltage and than to unsigned short int:

```
x msccm
= (x/1000.0) sccm
= (x/1000.0) * 5.0 V
= (unsigned short int) (((x/1000.0) * 5.0 + 10.0) * 65535.0 / 20.0)
```

If you set the input to the controller the input-panel will be written with the output unsigned short int translated to msccm -- e. g. if you set 500 msccm you get 499.98 msccm.

## 1.6 RF-Generator

### 1.6.1 RF-Generator: Combined Changes

Here you can choose some channels and the buttons to the right will only work on these channels. The 'RF On' and 'RF Off' works on all generator boxes which corresponds to the selected channels. Iff in the config file 'RF\_only\_master' in 'RF-Generator' is set to '1' only the first generator box will get the signal.

### 1.6.2 RF-Generator: ignite plasma

This is a procedure to ignite the plasma. Only the following settings will be done; e. g. the power of the generators or channels will not be changed.

1. RF off on all channels/generators



2. set maxcurrent\_tmp on all channels/generators
3. sleep of 0.01 seconds
4. RF on on all channels/generators
5. sleep of 0.1 seconds
6. set maxcurrent on all channels/generators
7. sleep of 1 second
8. set maxcurrent\_tmp on all channels/generators
9. sleep of 0.1 seconds
10. set maxcurrent on all channels/generators
11. step down to the settings before this procedure in 10 steps with sleepings of 0.1 seconds between the steps

### 1.6.3 RF-Generator: pattern

There are 2 different kinds of pattern. The first one is working on the mikrocontroller. The other one is working in the computer. You can choose it easily.

The pattern-input "entry" has 3 sections separated by ";". For example "1;2;01101001" means a pattern in 1 generator with 2 states. The states are given by "01101001" which represent the 2 states "0110" and "1001". "0110" means channel 1 to OFF (0), channel 2 to ON (1) and so on.

The intervall length is given in microseconds ( $10^{-6}$  sec). Shorter intervall length than 50 ms (= 50000 microseconds) is not possible on the computer; these ones can be run in the micro-controller. The micro-controller needs at least a interval length of  $20 * 10^{-6}$  sec.

With the "load" button you can load a file. Example:

```
[pattern]

# the number of generators with 4 channels each
number_of_generators: 1

# the number of states
pattern_length: 2

# interval length between 2 states in microseconds ( $10^{-6}$  sec)
pattern_intervall_length: 10000000

# the pattern is given by
# number_of_generators*4 * pattern_length "ON/OFF-switches".
# Whitespaces (Space, Newline, etc.) will be ignored.
pattern: 0110
        1001

# optional you can define here the controller: ("microcontroller" or "computer")
controller: computer
```

The "write2gen" button will write the pattern to the micro-controller. This is not necessary for running the pattern on the computer.

The "On/Off" switch start or stop the pattern. If the pattern is running on the computer, this could be a heavy load for the computer. Therefore no other changes with the RF-Generators are possible; but you can still try changes in the GUI, which are not given to the RF-Generator.

## 1.7 Diagnostics/Particles: Laser

The voltage of the diode should be between 0 V and 10 V. In the multi purpose controller is the range from -10 V to +10 V represented with 0 to 65535. This means the largest negative voltage the controller can be set to is -0.00015259021896696368 V; and the smallest positive voltage is 0.00015259021896696368. Therefore the laser gets voltages between 0.00015259021896696368 V and 10 V, which is hard coded.

## 1.8 Electrode Motion:

Only the last action is managed. For example you initiate 1000000 steps and then you click for 1 step, only 1 will be done and the other steps will be ignored. This is a feature and no bug.

## 1.9 Setpoints:

It is possible to define same sets of setpoints in a file and load and/or set these setpoints. For this functionality is also a keybinding available.

In the file you can define your sections as you like. Example with only 1 section but with all possible values:

```
[parabola 0 -- initial settings]
# 0 only load; 1 load and set
load_set: 1

# 1 for on and 0 for off
mass_flow_on_off: 1

# set mass flow to 500 msccm
mass_flow: 500

# 1 for on and 0 for off
pwr_channel_1: 1
pwr_channel_2: 0
pwr_channel_3: 0
pwr_channel_4: 1
pwr_channel_5: 0
pwr_channel_6: 0
pwr_channel_7: 0
pwr_channel_8: 0
pwr_channel_9: 0
pwr_channel_10: 0
pwr_channel_11: 0
pwr_channel_12: 0

# the same kind of values as in the gui
current_channel_1: 1000
current_channel_2: 0
current_channel_3: 0
current_channel_4: 1000
current_channel_5: 0
current_channel_6: 0
current_channel_7: 0
current_channel_8: 0
current_channel_9: 0
current_channel_10: 0
```

```

current_channel_11: 0
current_channel_12: 0

# the same kind of values as in the gui
phase_channel_1: 0
phase_channel_2: 0
phase_channel_3: 0
phase_channel_4: 0
phase_channel_5: 0
phase_channel_6: 0
phase_channel_7: 0
phase_channel_8: 0
phase_channel_9: 0
phase_channel_10: 0
phase_channel_11: 0
phase_channel_12: 0

# select channels for combined use
combined_channel_1: 1
combined_channel_2: 1
combined_channel_3: 1
combined_channel_4: 1
combined_channel_5: 1
combined_channel_6: 1
combined_channel_7: 1
combined_channel_8: 1
combined_channel_9: 1
combined_channel_10: 1
combined_channel_11: 1
combined_channel_12: 1

# 1 for on and 0 for off
rf_on_off: 1

# 1 for do it once and 0 for do not do it
ignite_plasma: 1

```

## 1.10 Logging:

Most of the logging is done by the WatchedFileHandler. This means, the logfiles grow indefinitely until an other process (e. g. logrotate or the user itself) move or delete the logfiles. Under Windows moving or deleting of open files is impossible and therefore the logfile grows indefinitely.

For example you can use logrotate with a config like:

```

# see "man logrotate" for details
# rotate log files daily
daily
# keep 14 days worth of backlogs
rotate 14
# compresses the log files
compress
delaycompress

```

```
# creates nice names
dateext
dateformat .%Y%m%d
# Do not mail old log files to any address.
nomail
/logs/digital_controller.log
/logs/digital_controller.data
/logs/multi_purpose_controller.log
/logs/multi_purpose_controller.data
/logs/plc.log
```

To start logrotate repeatedly you can use a cronjob, e. g.:

```
23 42 * * * /usr/sbin/logrotate --state ../logrotate.status ../logrotate.conf
```

## 2 Consistency of Time

Using different computers to handle the experiment setup leads to log data on different computers. The timestamps in the log data are produced by the real time clock of the computers.

So it is essential to have the same real time on every computer. For sure this is impossible. But we can do something to adjust the real times at its best.

The professional method is using NTP:

- [http://en.wikipedia.org/wiki/Network\\_Time\\_Protocol](http://en.wikipedia.org/wiki/Network_Time_Protocol)
- <http://ntp.org/>

Keeping in mind that the real time clock in a computer is a quartz oscillator which must be read out by software in an unknown time delay and is temperature-sensitive, it is clear that it is not simple!

In <http://www.ntp.org/ntpfaq/NTP-s-algo.htm#Q-ACCURATE-CLOCK> you can find typical values of time offsets. In particular it is mentioned that on a good network an offset of less than 1 ms is no problem.

By using NTP you can ask your ntp-daemon, e. g.:

```
plexp1:~$ ntpq -c pe
      remote           refid      st t when poll reach   delay   offset  jitter
=====
-afs-dbl.rzg.mpg 130.183.254.2    3 u  727 1024  377    0.164    1.176   0.973
-afs-db2.aug.ipp 130.183.14.14    4 u   73 1024  377    0.212     0.222   0.081
*afs-db3.bc.rzg. 130.183.254.2    3 u  606 1024  377    0.167   -0.081   0.089
+plracs.mpe.mpg.d 217.69.78.82     3 u  783 1024  377    0.409   -0.058   0.237
+plexp2.mpe.mpg. 130.183.136.102  4 u   19   64  377    0.100   -0.092   0.005

plexp2:~$ ntpq -c pe
      remote           refid      st t when poll reach   delay   offset  jitter
=====
-afs-dbl.rzg.mpg 130.183.254.2    3 u  228 1024  377    0.179    1.575   1.209
-afs-db2.aug.ipp 130.183.14.14    4 u    73 1024  377    0.205     0.240   0.065
+afs-db3.bc.rzg. 130.183.254.2    3 u 1009 1024  377    0.285     0.009   0.116
*plracs.mpe.mpg.d 217.69.78.82     3 u  665 1024  377    0.411   -0.001   0.240
+plexp1.mpe.mpg. 130.183.14.14    4 u   19   64  377    0.094     0.111   0.007
```

The output gives us a time offset between plexp1 and plexp2 of -0.092 ms or 0.111 ms. This means we have here really an offset of less than 1 ms.

But this configuration is really bad! Because both computers choose their time source independently. If these time sources are different and/or bad, it is clear that this is not the best. Therefore you should synchronize both to the same source or one to the other. Because of the high load by recording the cameras the corresponding computers are not a good time source. Therefore you should use another one if possible.

The small scripts `check_real_time_difference_client.py` and `check_real_time_difference_server.py` also gives a small and dirty information of the time offset between 2 computers, e. g.:

```
plexp1:~$ check_real_time_difference_client.py -ip plexp2.mpe.mpg.de
delay: 0.114594 msec; offset: 0.110760 msec
delay: 0.099449 msec; offset: 0.104457 msec
delay: 0.122928 msec; offset: 0.111998 msec
delay: 0.104924 msec; offset: 0.107838 msec
delay: 0.103755 msec; offset: 0.106728 msec
delay: 0.104784 msec; offset: 0.107840 msec
delay: 0.112901 msec; offset: 0.106488 msec
delay: 0.105300 msec; offset: 0.106973 msec
delay: 0.111587 msec; offset: 0.105774 msec
delay: 0.100853 msec; offset: 0.105916 msec

Time was more than 9726 times different and seems to be OK for 274 times.
average expected absolut time delay: 0.107477 msec
arithmetic mean of expected time delay: 0.106205 msec
standard deviation of expected time delay: 0.046266 msec
```

It is obvious that the real time clocks are different, but with only a small offset of less than 1 ms.

In more detail the expected time delay is around 0.1 msec with a standard deviation of around 0.05 msec. But the time delay on the network between the computers is about 0.1 msec. Hence we know without an assumption the time difference only with a precision of 0.1 msec. Anyway we have a real time clock difference of less than 1 ms.

On the institute network between my office and the lab we have another situation:

```
$ ntpq -c pe
      remote           refid      st t when poll reach   delay   offset  jitter
=====
*plrcs.mpe.mpg.d 217.69.78.82      3 u  234  512  377    0.377   -1.628   0.877
+plexp1.mpe.mpg. 130.183.14.14     4 u  173  512  377    0.143   -1.492   0.660
+plexp2.mpe.mpg. 130.183.136.102  4 u   51  512  377    0.165   -1.596   0.641

$ check_real_time_difference_client.py -ip plexp2.mpe.mpg.de
delay: 0.178970 msec; offset: 1.500981 msec
delay: 0.202898 msec; offset: 1.506871 msec
delay: 0.198203 msec; offset: 1.504049 msec
delay: 0.196783 msec; offset: 1.506424 msec
delay: 0.207130 msec; offset: 1.506066 msec
delay: 0.204803 msec; offset: 1.505876 msec
delay: 0.202456 msec; offset: 1.507442 msec
delay: 0.205850 msec; offset: 1.508251 msec
delay: 0.208686 msec; offset: 1.504560 msec
delay: 0.213785 msec; offset: 1.504321 msec
```

```
Time was more than 9994 times different and seems to be OK for 6 times.
average expected absolut time delay: 1.505484 msec
arithmetic mean of expected time delay: 1.505107 msec
standard deviation of expected time delay: 0.051197 msec
```

We can speculate of a real time clock difference of around 1.5 msec, which is also mentioned by NTP. The time delay on the network is around 0.2 msec. So the real time clock difference of around 1.5 msec is realistic. Looking at the algebraic sign of the offset ("arithmetic mean of expected time delay:" is positiv and "offset" from ntpq is negativ), we see plexp2 is around 1.5 msec behind the office computer.

Back to the ntp-daemon: Enable the following in the /etc/ntp.conf if you want statistics to be logged:

```
statsdir /var/log/ntpstats/
```

Now we find statistics in /var/log/ntpstats/peerstats, e. g. from the office statistics about plexp1:

day,	second,	address,	status,	offset,	delay,	dispersion,	skew (variance)
56321	36168.030	130.183.136.172	9424	-0.001496372	0.000158023	0.015597313	0.000926310
56321	37251.030	130.183.136.172	9424	-0.001492317	0.000143160	0.015639676	0.000659513
56321	39389.030	130.183.136.172	9324	-0.001234431	0.000144671	0.019652997	0.000451609

Or from plexp1 statistics about plexp2:

day,	second,	address,	status,	offset,	delay,	dispersion,	skew (variance)
56321	65001.715	130.183.136.120	941d	0.000164792	0.000100111	0.000602937	0.000011063
56321	65005.715	130.183.136.120	941d	0.000170311	0.000102348	0.000219969	0.000008765
56321	65064.715	130.183.136.120	941d	0.000177307	0.000097810	0.000465769	0.000007394

Or from plexp2 statistics about plexp1:

day,	second,	address,	status,	offset,	delay,	dispersion,	skew (variance)
56321	65073.890	130.183.136.172	941d	-0.000154547	0.000100871	0.000614891	0.000005763
56321	65075.890	130.183.136.172	941d	-0.000159778	0.000102325	0.000295086	0.000003462
56321	65077.890	130.183.136.172	941d	-0.000161170	0.000100325	0.000054969	0.000002973

The field names are from: <http://www.ntp.org/ntpfaq/NTP-s-trouble.htm#Q-TRB-MON-STATFIL>

In the file /var/log/ntpstats/loopstats we find information about the server itself, e. g. on my office computer:

day,	second,	offset,	drift compensation,	estimated error,	stability,	polling interval
56321	34992.030	-0.001478076	8.251	0.000295313	0.022122	10
56321	35089.030	-0.001333698	8.251	0.000280917	0.020694	10
56321	37188.030	-0.001561290	8.053	0.000274817	0.072682	9
56321	37724.030	-0.001548305	8.040	0.000257109	0.068128	9

On plexp1:

day,	second,	offset,	drift compensation,	estimated error,	stability,	polling interval
56321	63031.715	0.000145351	-28.121	0.000108917	0.266518	3
56321	63035.715	0.000146621	-28.121	0.000101884	0.249305	6
56321	63297.715	0.000191703	-28.073	0.000096627	0.233817	6
56321	63557.715	0.000175056	-28.030	0.000090578	0.219254	7

On plexp2:

day,	second,	offset,	drift compensation,	estimated error,	stability,	polling interval
56321	63573.890	0.000006417	-42.375	0.000036539	0.026354	5
56321	63761.890	0.000031242	-42.369	0.000035288	0.024731	6
56321	63828.890	0.000029910	-42.367	0.000033013	0.023144	6
56321	64353.890	0.000039841	-42.348	0.000031079	0.022715	6

## 3 Other Scripts

### 3.1 plc\_viewer.py

The `plc_viewer.py` will play zip-archives with png-pictures inside. A file named "timestamps.txt" must also be inside the archive. This file describes the time of the frames. This kind of archive can be created by `rawmovies2recordings.py`.

The help output is:

```
usage: plc_viewer.py [-h] [-f dir [dir ...]] [-scale x] [-absolutscale x]
                  [-timeratefactor x] [-camcolumn c] [-config file]
                  [-index i] [-create_info_graphics c] [-debug debug_level]

plc-viewer

optional arguments:
  -h, --help                show this help message and exit
  -f dir [dir ...]          will play this directory or this file. default: ./
  -scale x                  Set the scale factor x. default: x = 1.0
  -absolutscale x           Set the absolut scale to x pixel width. default: x =
                             -1 (dissabled)
  -timeratefactor x         Set the time rate factor x. default: x = 1.0
  -camcolumn c              Set the number of columns for the cams. default: c = 3
  -config file              Set the config as used by measuring. (default:
                             './plc.cfg')
  -index i                  If set to 1: create only index and exit.
  -create_info_graphics c   0 do not create info graphics (default for viewing a
                             file); 1 create info graphics (default for viewing a
                             directory).
  -debug debug_level        Set debug level. 0 no debug info (default); 1 debug to
                             STDOUT.

Author: Daniel Mohr
Date: 2012-12-10
License:

Examples:
  plc_viewer.py -f /home/mohr/examplecams/
  plc_viewer.py -s 0.2 -ca 3
  plc_viewer.py -absolutscale 300 -camcolumn 4
```

To play only a single file:

```
plc_viewer.py -f 2012-09-10_PF2011-CAM3_rec_030.zip -a 500 -ca 3
```

If you play a directory many movies are synchronized by the time, which is displayed in the upper right corner. Stepping through the frames by +1 or -1 will not change the time. This function goes 1 frame forward or backward in every movie; therefore the time between the movies is not synchronized anymore. If you choose (also after going forward or backward 1 frame) a time (play, yalp or +- 1s or ...) all movies are synchronized to the given time in the upper right corner.

In the top of the window you see 3 lines of graphics. The first line displays vertical lines with the height of the accumulated changes between 2 frames of all movies. The x-axis is scaled from the time of first frame in the given data to time of the last frame in the given data. The color is representing the brightness of a frame: red for the first movie, green for the second movie and blue for the third movie; more movies are ignored. The graphics in the second line represent only the brightness of the corresponding frame (time and movie; for every movie is a graphic) by the height of the white line. The next graphics in the third line represent the changes between the corresponding frame and the next one by the height of the white line. The actual time of the movies will be displayed by a vertical line in white and gray (50 % of white) respectively. Every graphic is scaled on there own to all necessary values.

## 3.2 rawmovies2recordings.py

The `rawmovies2recordings.py` converts img-files (PAM format) to zip-archives with single png-files and in addition a "timestamps.txt"-file.

The help output is:

```
usage: rawmovies2recordings.py [-h] [-debug debug_level] -f file [file ...]
                               [-outdir dir] [-prefix dir]

rawmovies2recordings.py

optional arguments:
  -h, --help                show this help message and exit
  -debug debug_level        Set debug level. 0 no debug info (default); 1 debug to
                             STDOUT.
  -f file [file ...]        will convert this/these file(s)
  -outdir dir               Set the output directory. Default: ./
  -prefix dir               Set the prefix for the archiv names. Default: rec

Author: Daniel Mohr
Date: 2012-09-24
License:

Example: rawmovies2recordings.py -d 1 -f cam_0_Pf2011-CAM3_2012-09-11_006.img -o t -p "2012-09-11_Pf2011-CAM3_rec_"
```

## 3.3 rawmovieviewer.py

The `rawmovieviewer.py` play directly the img-files (PAM format).

The help output is:

```
usage: rawmovieviewer.py [-h] -f file [file ...] [-scale x]
                        [-timeratefactor x] [-istep i] [-debug debug_level]

RawmovieViewer

optional arguments:
  -h, --help                show this help message and exit
  -f file [file ...]        will play this file(s)
  -scale x                  Set the scale factor x. default: x = 1.0
  -timeratefactor x         Set the time rate factor x. default: x = 1.0
  -istep i                  Only every ith frame will be shown. default: i = 1
  -debug debug_level        Set debug level. 0 no debug info (default); 1 debug to
                             STDOUT.
```



Author: Daniel Mohr  
 Date: 2012-09-20  
 License:

Examples:

```
rawmovieviewer.py -f movie1.img movie2.img movie3.img
rawmovieviewer.py -f movie.img -d 1 -t 0.01 -s 2
rawmovieviewer.py -f movie.img -i 10 -t 10
```

### 3.4 digital\_controller\_server.py

The help output is:

```
usage: digital_controller_server.py [-h] [-device dev] [-logfile f]
                                   [-datalogfile f] [-runfile f] [-ip n]
                                   [-port p] [-timedelay t] [-choosenextport]
                                   [-A d] [-B d] [-C d] [-D d]
                                   [-debug debug_level] [-simulate]
```

digital\_controller\_server is a socket server to control the digital controller on an serial interface. On start every settings are assumed to 0 or the given values and set to the device. A friendly kill (SIGTERM) should be possible.

optional arguments:

-h, --help	show this help message and exit
-device dev	Set the external device dev to communicate with the box.
-logfile f	Set the logfile to f. The WatchedFileHandler is used. This means, the logfile grows indefinitely until an other process (e. g. logrotate or the user itself) move or delete the logfile. Under Windows moving or deleting of open files is impossible and therefore the logfile grows indefinitely. default: /tmp/digital_controller.log
-datalogfile f	Set the datalogfile to f. Only the measurements will be logged here. The WatchedFileHandler is used. This means, the logfile grows indefinitely until an other process (e. g. logrotate or the user itself) move or delete the logfile. Under Windows moving or deleting of open files is impossible and therefore the logfile grows indefinitely. default: /tmp/digital_controller.data
-runfile f	Set the runfile to f. If an other process is running with a given pid and writing to the same device, the program will not start. Setting f="" will disable this function. default: /tmp/digital_controller.pids
-ip n	Set the IP/host n to listen. If ip == "" the default behavior will be used; typically listen on all possible addresses. default: localhost
-port p	Set the port p to listen. If p == 0 the default behavior will be used; typically choose a port. default: 15112
-timedelay t	Set the time between 2 actions to t seconds. default: t = 0.05
-choosenextport	By specifying this flag the next available port after

## plc - PlasmaLabControl

```
the given one will be choosen. Without this flag a
socket.error is raised if the port is not available.
-A d      Set the default values for the digital controller port
          A; "0" for channel off and "1" for channel on;
          "10000000" means only channel 0 to ON. default: d =
          "00000000"
-B d      Set the default values for the digital controller port
          B; "0" for channel off and "1" for channel on;
          "10000000" means only channel 0 to ON. default: d =
          "00000000"
-C d      Set the default values for the digital controller port
          C; "0" for channel off and "1" for channel on;
          "10000000" means only channel 0 to ON. default: d =
          "00000000"
-D d      Set the default values for the digital controller port
          D; "0" for channel off and "1" for channel on;
          "10000000" means only channel 0 to ON. default: d =
          "00000000"
-debug debug_level Set debug level. 0 no debug info (default); 1 debug to
STDOUT.
-simulate  By specifying this flag a random sleep simulates the
communication to the device.
```

Author: Daniel Mohr

Date: 2013-03-14

License:

Over the given port on the given address a socket communication is lisening with the following commands: (This is a prefix-code. Upper or lower letters do not matter.)

```
p[pickle data] : Set all setpoints at once. You have all setpoints in one
object:
a = {'A': 8*[False], 'B': 8*[False],
     'dispenser': {'n': False, 'ton': False, 'shake': False,
                   'port': False, 'channel': False, 'toff': False},
     'C': 8*[False], 'D': 8*[False]}
Now you can generate the [pickle data]==v by:
s = pickle.dumps(a,-1); v='%d %s' % (len(s),s)
s[unsigned char][unsigned char][unsigned char][unsigned char] :
set the 4 ports to the On/Off values on the ports
[A|B|C|D][0|1][0|1][0|1][0|1][0|1][0|1][0|1] :
set the channels on the port [A|B|C|D] to On/Off
_dispenserPC00111222 : Choose the values for the dispenser shake. P is the
port [A|B|C|D] and C is the channel [0|1|2|3|4|5|6|7].
00 are 2 digits for the number of shakes; 111 are 3
digits for the T_on time in milliseconds; 222 are 3
digits for the T_off time in milliseconds.
!dispenser : shake the dispenser with the choosen values
!w2d : trigger writing setvalues to the external device
getact : sends the actual values back as [pickle data]
timedelay000 : set the time between 2 actions to 000 milliseconds.
quit : quit the server
version : response the version of the server
```

### 3.5 digital\_controller\_client.py

The help output is:

```
usage: digital_controller_client.py [-h] [-ip n] [-port p]
                                   [-debug debug_level]

digital_controller_client is a client to speak with the socket server
digital_controller_server.py to control the digital controller on an
serial interface.

optional arguments:
  -h, --help            show this help message and exit
  -ip n                  Set the IP/host n. default: localhost
  -port p                Set the port p. default: 15112
  -debug debug_level    Set debug level. 0 no debug info (default); 1 debug to
                        STDOUT.

Author: Daniel Mohr
Date: 2013-03-06
License:

Type help() for interactive help, or help(object) for help about object.
```

### 3.6 multi\_purpose\_controller\_server.py

The help output is:

```
usage: multi_purpose_controller_server.py [-h] [-device dev] [-logfile f]
                                         [-runfile f] [-ip n] [-port p]
                                         [-timedelay t] [-choosenextport]
                                         [-DO d] [-R d] [-U05 d] [-U15 d]
                                         [-U24 d] [-DAC d]
                                         [-debug debug_level]

multi_purpose_controller_server is a socket server to control the
multi purpose controller on an serial interface. On start every
settings are assumed to 0 or the given values and set to the device.
A friendly kill (SIGTERM) should be possible.

optional arguments:
  -h, --help            show this help message and exit
  -device dev            Set the external device dev to communicate with the box.
  -logfile f            Set the logfile to f. The WatchedFileHandler is used.
                        This means, the logfile grows indefinitely until an
                        other process (e. g. logrotate or the user itself) move
                        or delete the logfile. Under Windows moving or deleting
                        of open files is impossible and therefore the logfile
                        grows indefinitely. default:
                        /tmp/multi_purpose_controller.log
  -datalogfile f        Set the datalogfile to f. Only the measurements will be
                        logged here. The WatchedFileHandler is used. This means,
                        the logfile grows indefinitely until an other process
```

## plc - PlasmaLabControl

```
(e. g. logrotate or the user itself) move or delete the
logfile. Under Windows moving or deleting of open files
is impossible and therefore the logfile grows
indefinitely. default:
/tmp/multi_purpose_controller.data
-runfile f      Set the runfile to f. If an other process is running
                with a given pid and writing to the same device, the
                program will not start. Setting f="" will disable this
                function. default: /tmp/multi_purpose_controller.pids
-ip n           Set the IP/host n to listen. If ip == "" the default
                behavior will be used; typically listen on all possible
                addresses. default: localhost
-port p        Set the port p to listen. If p == 0 the default behavior
                will be used; typically choose a port. default: 15113
-timedelay t   Set the time between 2 actions to t seconds. default: t
                = 0.05
-choosenextport By specifying this flag the next available port after
                the given one will be choosen. Without this flag a
                socket.error is raised if the port is not available.
-DO d          Set the default values for the multi purpose controller
                port DO; "0" for channel off and "1" for channel on;
                "0001" means only channel 1 to ON. default: d = "0000"
-R d           Set the default values for the multi purpose controller
                port R; "0" for channel off and "1" for channel on; "01"
                means only channel 1 to ON. default: d = "00"
-U05 d         Set the default values for the multi purpose controller
                port U05; "0" for channel off and "1" for channel on.
                default: d = "0"
-U15 d         Set the default values for the multi purpose controller
                port U15; "0" for channel off and "1" for channel on.
                default: d = "0"
-U24 d         Set the default values for the multi purpose controller
                port U24; "0" for channel off and "1" for channel on.
                default: d = "0"
-DAC d         Set the default values for the multi purpose controller
                port DAC. default: d = "-10,-10,-10,-10"
-debug debug_level Set debug level. 0 no debug info (default); 1 debug to
                STDOUT.
```

Author: Daniel Mohr

Date: 2013-02-04

License:

Over the given port on the given address a socket communication is  
lisening with the following commands: (This is a prefix-code. Upper  
or lower letters do not matter.)

```
p[pickle data] : Set all setpoints at once. You have all setpoints in one
                  object:
                  a = {'DO':4*[False], 'R':2*[False], 'U05':False,
                      'U15':False, 'U24':False, 'DAC':4*[0.0]}
                  Now you can generate the [pickle data]==v by:
                  s = pickle.dumps(a,-1); v='%d %s' % (len(s),s)
!w2d : trigger writing setvalues to the external device
getact : sends the actual values back as [pickle data]
```

```

timedelay000 : set the time between 2 actions to 000 milliseconds.
quit : quit the server
version : response the version of the server

```

### 3.7 multi\_purpose\_controller\_client.py

The help output is:

```

usage: multi_purpose_controller_client.py [-h] [-ip n] [-port p]
                                         [-debug debug_level]

multi_purpose_controller_client is a client to speak with the socket
server multi_purpose_controller_server.py to control the multi purpose
controller on an serial interface.

optional arguments:
  -h, --help            show this help message and exit
  -ip n                  Set the IP/host n. default: localhost
  -port p                Set the port p. default: 15113
  -debug debug_level    Set debug level. 0 no debuginfo (default); 1 debug to
                        STDOUT.

Author: Daniel Mohr
Date: 2012-10-15
License:

Type help() for interactive help, or help(object) for help about object.

```

### 3.8 camera\_server.py

The help output is:

```

usage: camera_server.py [-h] [-listcams] [-guid id] [-mode m]
                        [-color_coding c] [-framerate i] [-logfile f]
                        [-runfile f] [-ip n] [-port p] [-choosenextport]
                        [-ringbuf n] [-recvbuf n] [-debug debug_level]

camera_server is a socket server to control a camera on firewire. A friendly
kill (SIGTERM) should be possible.

optional arguments:
  -h, --help            show this help message and exit
  -listcams              Only list available cams and exit.
  -guid id               A camera with this guid will be used.
  -mode m                Set the camera mode. default: FORMAT7_0
  -color_coding c        Set the color_coding for the camera. default: Y8
  -framerate i           Set the framerate for the camera.
  -logfile f             Set the logfile to f. The WatchedFileHandler is used.
                        This means, the logfile grows indefinitely until an
                        other process (e. g. logrotate or the user itself) move
                        or delete the logfile. Under Windows moving or deleting
                        of open files is impossible and therefore the logfile

```

```

grows indefinitely. default: /tmp/camera.log
-runfile f      Set the runfile to f. If an other process is running
                with a given pid and writing to the same device, the
                program will not start. Setting f="" will disable this
                function. default: /tmp/camera.pids
-ip n           Set the IP/host n to listen. If ip == "" the default
                behavior will be used; typically listen on all possible
                addresses. default: localhost
-port p         Set the port p to listen. If p == 0 the default behavior
                will be used; typically choose a port. default: 15114
-choosenextport By specifying this flag the next available port after
                the given one will be choosen. Without this flag a
                socket.error is raised if the port is not available.
-ringbuf n      Set the number of buffers in the ring buffer of dc1394.
                default: 16
-recvbuf n      Set the number of Bytes to receive at once by the socket
                communication. default: 4096
-debug debug_level Set debug level. 0 no debug info (default); 1 debug to
                STDOUT.

```

Author: Daniel Mohr

Date: 2013-02-27

License:

Over the given port on the given address a socket communication is listening with the following commands: (This is a prefix-code. Upper or lower letters do not matter.)

```

listcams : This will list all available cams and send back this information
            as [pickle data]. The data start with a decimal number describing
            the number of bytes for the pickled data block; then comes a space
            and the pickled data itself. The data is an array of dicts.
getvalues : This sends back the settings of the camera as [pickle data]. It
            is the same format as for listcams. The data is a dict.
setvalues : This sets the settings of the camera as [pickle data]. This is
            the same format as for getvalues. If s are the [pickle data],
            you should send "setvalues %d %s" % (len(s),s)
startcam  : This starts the camera.
getlframe : This sends the actual frame back.
startrec  : This starts recording.
stoprec   : This stops recording.
stopcam   : This stops the camera.
setpathes : Set the pathes/prefixes to write the images to. It is the same
            format as for setvalues
quit      : quit the server
version   : response the version of the server

```

If you have problems with your camera or firewire system, try:

```
"DC1394_DEBUG=1 camera_server.py -d 1"
```

### 3.9 camera\_client.py

The help output is:

```

usage: camera_client.py [-h] [-ip n] [-port p] [-recvbuf n]
                        [-update_img_delay a] [-debug debug_level]

```

camera\_client is a client to speak with the socket server camera\_server.py to control a camera attached to the server by firewire.

optional arguments:

-h, --help	show this help message and exit
-ip n	Set the IP/host n. default: localhost
-port p	Set the port p. default: 15114
-recvbuf n	Set the number of Bytes to receive at once by the socket communication. default: 4096
-update_img_delay a	Set the minimum time delay between displaying 2 images. default: 6
-debug debug_level	Set debug level. 0 no debug info (default); 1 debug to STDOUT.

Author: Daniel Mohr

Date: 2013-03-12

License:

crop function: By clicking with the left mouse button on the picture and release the mouse button on an possibly other position, the resulting rectangle will be displayed and the margin will be cropped. You come back to the original view by clicking with the right mouse button.

### 3.10 acceleration\_sensor\_logger.py

The help output is:

```
usage: acceleration_sensor_logger.py [-h] [-logfile f] [-idVendor x]
                                     [-idProduct x] [-listsensors]
                                     [-SerialNumber x] [-id i]
                                     [-debug debug_level]
```

acceleration\_sensor\_logger.py logs measurements from the JoyWarrior24F14 to a logfile. You need access to the device.

optional arguments:

-h, --help	show this help message and exit
-logfile f	Set the logfile to f. The WatchedFileHandler is used. This means, the logfile grows indefinitely until an other process (e. g. logrotate or the user itself) move or delete the logfile. Under Windows moving or deleting of open files is impossible and therefore the logfile grows indefinitely. default: /tmp/acceleration.log
-idVendor x	Set the idVendor of the acceleration sensor. default: 0x07c0
-idProduct x	Set the idProduct of the acceleration sensor. default: 0x1116
-listsensors	Will list the acceleration sensor(s) and exit.
-SerialNumber x	Set the SerialNumber of the acceleration sensor. If given try to find this sensor otherwise use the one given by id.
-id i	Set the id to i. If there are more than 1 acceleration sensor and there is no SerialNumber or the SerialNumber

```

                                was not found, the i-th one will be choosen.
-debug debug_level      Set debug level. 0 no debug info (default); 1 debug to
                                STDOUT.

```

Author: Daniel Mohr

Date: 2012-11-30

License:

This is a simple command line program. You can plot the logfile with gnuplot.

A few examples:

```

plot 'acceleration.log' using 1:2 with lines title 'x',\
'acceleration.log' using 1:3 with lines title 'y',\
'acceleration.log' using 1:4 with lines title 'z'

plot 'acceleration.log' using ($1-1350411408.795725):2 with lines title 'x',\
'acceleration.log' using ($1-1350411408.795725):3 with lines title 'y',\
'acceleration.log' using ($1-1350411408.795725):4 with lines title 'z'

set xdata time ; set timefmt '%s' ; set format x '%H:%M'
plot 'acceleration.log' using 1:2 with lines title 'x',\
'acceleration.log' using 1:3 with lines title 'y',\
'acceleration.log' using 1:4 with lines title 'z'

```

You need access to the device of the sensor.

For example you can use the following udev rule:

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="07c0", ATTRS{idProduct}=="1116", MODE:="666", GROUP="users"
```

## 3.11 acceleration\_sensor\_server.py

The help output is:

```

usage: acceleration_sensor_server.py [-h] [-idVendor x] [-idProduct x]
                                     [-listsensors] [-SerialNumber x] [-id i]
                                     [-logfile f] [-datalogfile f]
                                     [-datalogformat f] [-maxg x] [-runfile f]
                                     [-ip n] [-port p] [-choosenextport]
                                     [-debug debug_level]

```

acceleration\_sensor\_server.py is a socket server to read and log the measurements from the acceleration sensor JoyWarrior24F14. A friendly kill (SIGTERM) should be possible.

optional arguments:

-h, --help	show this help message and exit
-idVendor x	Set the idVendor of the acceleration sensor. default: 0x07c0
-idProduct x	Set the idProduct of the acceleration sensor. default: 0x1116
-listsensors	Will list the acceleration sensor(s) and exit.
-SerialNumber x	Set the SerialNumber of the acceleration sensor. If given try to find this sensor otherwise use the one given by id.
-id i	Set the id to i. If there are more than 1 acceleration sensor and there is no SerialNumber or the SerialNumber was not found, the i-th one will be choosen.
-logfile f	Set the logfile to f. The WatchedFileHandler is used. This means, the logfile grows indefinitely until an



```

other process (e. g. logrotate or the user itself) move
or delete the logfile. Under Windows moving or deleting
of open files is impossible and therefore the logfile
grows indefinitely. default: /tmp/acceleration.log
-datalogfile f      Set the datalogfile to f. Only the measurements will be
                    logged here. The WatchedFileHandler is used. This means,
                    the logfile grows indefinitely until an other process
                    (e. g. logrotate or the user itself) move or delete the
                    logfile. Under Windows moving or deleting of open files
                    is impossible and therefore the logfile grows
                    indefinitely. default: /tmp/acceleration.data
-datalogformat f    Set the log format for the data: 0 raw format; 1 value
                    in g. default: 0
-maxg x             Set the measurement range in g. default 2 for +-2g
-runfile f          Set the runfile to f. If an other process is running
                    with a given pid and reading the same SerialNumber, the
                    program will not start. Setting f="" will disable this
                    function. default: /tmp/acceleration_sensor.pids
-ip n               Set the IP/host n to listen. If ip == "" the default
                    behavior will be used; typically listen on all possible
                    addresses. default: localhost
-port p             Set the port p to listen. If p == 0 the default behavior
                    will be used; typically choose a port. default: 15123
-choosenextport     By specifying this flag the next available port after
                    the given one will be choosen. Without this flag a
                    socket.error is raised if the port is not available.
-debug debug_level  Set debug level. 0 no debug info (default); 1 debug to
                    STDOUT.

```

Author: Daniel Mohr  
 Date: 2013-02-28  
 License:

Over the given port on the given address a socket communication is listening with the following commands: (This is a prefix-code. Upper or lower letters do not matter.)

```

getact : sends the actual values back as [pickle data]
quit   : quit the server
version : response the version of the server

```

## 3.12 acceleration\_sensor\_client.py

The help output is:

```

usage: acceleration_sensor_client.py [-h] [-ip n] [-port p] [-bwgraphics i]
                                     [-colorgraphics i] [-diagram i]
                                     [-resolution p] [-sleep s] [-shadow n]
                                     [-diagramlength n] [-maxg x]
                                     [-update_display_delay a]
                                     [-debug debug_level]

```

acceleration\_sensor\_client is a client to speak with the socket server  
 acceleration\_sensor\_server.py to control the acceleration sensor.

```
optional arguments:
  -h, --help            show this help message and exit
  -ip n                  Set the IP/host n. default: localhost
  -port p                Set the port p. default: 15123
  -bwgraphics i          Setting this flag to 1 enables black/white graphics.
                        default: 1
  -colorgraphics i       Setting this flag to 1 enables color graphics.
                        default: 0
  -diagram i             Setting this flag to 1 enables the diagram graphics.
                        default: 0
  -resolution p          Set the width and height of the graphics to p pixel.
                        default: 400
  -sleep s               Set the sleep time in seconds between reading new
                        values from the server. Shorter than 0.008 is useless.
                        default: 0.035
  -shadow n              Set length of the shadow. default: 16
  -diagramlength n        Set length of the diagram. default: 320
  -maxg x                Set the measurement range in g. default 2 for +-2g
  -update_display_delay a Set the minimum time delay between displaying new
                        values. default: 6
  -debug debug_level      Set debug level. 0 no debug info (default); 1 debug to
                        STDOUT.

Author: Daniel Mohr
Date: 2013-02-22
License:

Type help() for interactive help, or help(object) for help about object.
```

### 3.13 pressure\_mks\_651\_client.py

The help output is:

```
usage: pressure_mks_651_client.py [-h] [-ip n] [-port p] [-debug debug_level]

mks_651_client is a client to speak with the socket server mks_651_server.py to
control the series 651 pressure controller on a serial interface.

optional arguments:
  -h, --help            show this help message and exit
  -ip n                  Set the IP/host n. default: localhost
  -port p                Set the port p. default: 15122
  -debug debug_level      Set debug level. 0 no debug info (default); 1 debug to
                        STDOUT.

Author: Richard Schlitz
Date: 2013-01-22
License:

Type help() for interactive help, or help(object) for help about object.
```

### 3.14 pressure\_mks\_651\_server.py

The help output is:

```
usage: pressure_mks_651_server.py [-h] [-device dev] [-logfile f] [-runfile f]
                                  [-ip n] [-port p] [-timedelay t]
                                  [-choosenextport] [-debug debug_level]
```

mks\_651\_server is a socket server to control the MKS-Typ 651C controller on a serial interface. On start all settings are fetched from the controller and the gui is initialized with these. A friendly kill (SIGTERM) should be possible.

optional arguments:

-h, --help	show this help message and exit
-device dev	Set the external device dev to communicate with the box.
-logfile f	Set the logfile to f. The WatchedFileHandler is used. This means, the logfile grows indefinitely until an other process (e. g. logrotate or the user itself) move or delete the logfile. Under Windows moving or deleting of open files is impossible and therefore the logfile grows indefinitely. default: /tmp/mks_651_controller.log
-runfile f	Set the runfile to f. If an other process is running with a given pid and writing to the same device, the program will not start. Setting f="" will disable this function. default: /tmp/pressure_controller.pids
-ip n	Set the IP/host n to listen. If ip == "" the default behavior will be used; typically listen on all possible addresses. default: localhost
-port p	Set the port p to listen. If p == 0 the default behavior will be used; typically choose a port. default: 15122
-timedelay t	Set the time between 2 actions to t seconds. default: t = 0.05
-choosenextport	By specifying this flag the next available port after the given one will be choosen. Without this flag a socket.error is raised if the port is not available.
-debug debug_level	Set debug level. 0 no debug info (default); 1 debug to STDOUT.

Author: Richard Schlitz

Date: 2013-01-22

License:

Over the given port on the given address a socket communication is listening with the following commands: (This is a prefix-code. Upper or lower letters do not matter.)

```
h : sends a signal to stop the vent to the server
c : sends a signal to close the vent to the server
o : sends a signal to open the vent to the server
p : gets the actual pressure value from the server
v : gets the actual vent position from the server(in %)
setact : sets the setpoint values as [pickle data]
getact : gets the actual values back as [pickle data]
timedelay000 : set the time between 2 actions to 000 milliseconds.
quit : quit the server
version : response the version of the server
```

### 3.15 pressure\_mks\_900\_client.py

The help output is:

```
usage: pressure_mks_900_client.py [-h] [-ip n] [-port p] [-debug debug_level]

pressure900_client is a client to speak with the socket server
pressure900_server.py to control the series 900 pressure controller on a serial
interface.

optional arguments:
  -h, --help            show this help message and exit
  -ip n                  Set the IP/host n. default: localhost
  -port p                Set the port p. default: 15121
  -debug debug_level    Set debug level. 0 no debug info (default); 1 debug to
                        STDOUT.

Author: Richard Schlitz
Date: 2013-01-03
License:

Type help() for interactive help, or help(object) for help about object.
```

### 3.16 pressure\_mks\_900\_server.py

The help output is:

```
haha
usage: pressure_mks_900_server.py [-h] [-device dev] [-logfile f] [-runfile f]
                                   [-ip n] [-port p] [-timedelay t]
                                   [-choosenextport] [-PR d] [-U d] [-GT d]
                                   [-debug debug_level]

pressure900_server is a socket server to control the MKS-PDR900-1 controller on
a serial interface. On start all settings are fetched from the controller and
then reset with the initialization values. A friendly kill (SIGTERM) should be
possible.

optional arguments:
  -h, --help            show this help message and exit
  -device dev            Set the external device dev to communicate with the box.
  -logfile f             Set the logfile to f. The WatchedFileHandler is used.
                        This means, the logfile grows indefinitely until an
                        other process (e. g. logrotate or the user itself) move
                        or delete the logfile. Under Windows moving or deleting
                        of open files is impossible and therefore the logfile
                        grows indefinitely. default:
                        /tmp/pressure_controller.log
  -runfile f             Set the runfile to f. If an other process is running
                        with a given pid and writing to the same device, the
                        program will not start. Setting f="" will disable this
                        function. default: /tmp/pressure_controller.pids
  -ip n                  Set the IP/host n to listen. If ip == "" the default
                        behavior will be used; typically listen on all possible
                        addresses. default: localhost
  -port p                Set the port p to listen. If p == 0 the default behavior
```

```

will be used; typically choose a port. default: 15121
-timedelay t      Set the time between 2 actions to t seconds. default: t
                  = 0.05
-choosenextport    By specifying this flag the next available port after
                  the given one will be choosen. Without this flag a
                  socket.error is raised if the port is not available.
-PR d             Set the default values for the filament to use for
                  measuring the pressure (PR1,PR2,PR3); default: d = "PR3"
-U d              Set the default unit for the pressure controller to use
                  (MBAR,TORR,PASCAL); default: d = PRESET
-GT d             Set the default gas type for the pressure controller to
                  use (ARGON,NITROGEN,AIR,HYDROGEN,HELIUM); default: d =
                  PRESET
-debug debug_level Set debug level. 0 no debug info (default); 1 debug to
                  STDOUT.

```

Author: Richard Schlitz  
 Date: 2013-01-24  
 License:

Over the given port on the given address a socket communication is listening with the following commands: (This is a prefix-code. Upper or lower letters do not matter.)

```

p : returns the pressure to the client
setact : sets the setpoint to a given setpoint received as [pickle data]
getact : sends the actual values back as [pickle data]
timedelay000 : set the time between 2 actions to 000 milliseconds.
quit : quit the server
version : response the version of the server

```

### 3.17 check\_real\_time\_difference\_server.py

The help output is:

```

usage: check_real_time_difference_server.py [-h] [-ip n] [-port p] [-wait]

check_real_time_difference_server.py is a small program to check the real times
on different computers.

optional arguments:
  -h, --help  show this help message and exit
  -ip n       Set the IP/host n to listen. If ip == "" the default behavior
              will be used; typically listen on all possible addresses.
              default: localhost
  -port p     Set the port p to listen. If p == 0 the default behavior will be
              used; typically choose a port. default: 15124
  -wait       By specifying this flag the server tries to use the given port
              until it is possible.

```

Author: Daniel Mohr  
 Date: 2013-01-28  
 License:

### 3.18 check\_real\_time\_difference\_client.py

The help output is:

```
usage: check_real_time_difference_client.py [-h] [-ip n] [-port p] [-nn nn]
                                           [-n n] [-dn n]
                                           [-debug debug_level]

check_real_time_difference_client.py is a small program to check the real
times/dates between different computers.

optional arguments:
  -h, --help            show this help message and exit
  -ip n                  Set the IP/host n. default: localhost
  -port p                Set the port p. default: 15124
  -nn nn                 Do the n communications nn times. default: 10
  -n n                   Do the communication n times. default: 1000
  -dn n                  Sends n bytes every time. default: 1
  -debug debug_level    Set debug level. 0 no debug info (default); 1 debug to
                        STDOUT.

Author: Daniel Mohr
Date: 2013-01-28
License:
```

### 3.19 translation\_stage\_scan.py

The help output is:

```
usage: translation_stage_scan.py [-h] [-direction xyz] [-repeat n] [-steps n]
                                  [-delay t] [-set_zero_position x]
                                  [-go_back x] [-go_direct_back x]
                                  [-device dev] [-baudrate n] [-databits n]
                                  [-stopbits n] [-logfile f]

translation_stage_scan.py is a simple tool to perform a scan with the
translation stage. The device must be already powered.
This script initialize repeatedly some steps and a delay. Optionally the
default position should be reached after all.

A quick and dirty measurement gives us 1000000 steps for 7.9 cm in about
33 seconds.

The timestamps of the positions in the log file are only based on the
commands. From 'initiated next position' it takes some time to perform your
choosen steps. They should be reached exactly in this time. The position
information is given at the time of the answer from the device; not when it
is reached! So again, the timestamp of 'initiated next position' added by the
necessary time delay to perform your choosen steps should be the time when
the device reached the next position.

optional arguments:
  -h, --help            show this help message and exit
  -direction xyz         Set the directions. (0: x; 1: y; 2: z) default: 1
  -repeat n              Set the number of repeatations. default: 2
```

## plc - PlasmaLabControl

```
-steps n          Set the number of steps to do each time. default: 100
-delay t          Set the delay between the repetitions in seconds.
                  default: 0.1
-set_zero_position x If set to 1 the zero position will be set at the
                  beginning of the communication. default: 1
-go_back x         If set to 1 go back to the start position after all.
                  default: 1
-go_direct_back x  If set to 1 go direct back to the start position after
                  all. default: 1
-device dev        Set the external device dev to communicate with the
                  box. default: /dev/TSCftBNKEKX
-baudrate n        Set the baudrate. default: 9600
-databits n        Set the databits. default: 8
-stopbits n        Set the stopbits. (possible values: 1, 1.5, 2)
                  default: 1
-logfile f         Set the logfile to f. The WatchedFileHandler is used.
                  This means, the logfile grows indefinitely until an
                  other process (e. g. logrotate or the user itself)
                  move or delete the logfile. Under Windows moving or
                  deleting of open files is impossible and therefore the
                  logfile grows indefinitely. default:
                  /tmp/translation_stage_scan.log
```

Author: Daniel Mohr

Date: 2013-03-05

License:

Examples: (After the given delays the position should be reached.)

```
translation_stage_scan.py -repeat 1 -steps -1000000 -delay 33
translation_stage_scan.py -repeat 10 -steps -100000 -delay 3.4
translation_stage_scan.py -repeat 100 -steps -10000 -delay 0.4
translation_stage_scan.py -repeat 1000 -steps -1000 -delay 0.1
translation_stage_scan.py -repeat 10000 -steps -100 -delay 0.036
translation_stage_scan.py -repeat 100000 -steps -10 -delay 0.036 # caution: heat!!
translation_stage_scan.py -repeat 1000000 -steps -1 -delay 0.036 # caution: heat!!!
```

## 3.20 environment\_sensor\_5\_logger.py

The help output is:

```
usage: environment_sensor_5_logger.py [-h] [-logfile f] [-datalogfile f]
                                     [-devicename dev] [-sleep s]
                                     [-baudrate n] [-runfile f]
                                     [-debug debug_level]
```

environment\_sensor\_5\_logger.py logs measurements from the environment sensor 5 to a logfile. You need access to the device.

optional arguments:

```
-h, --help          show this help message and exit
-logfile f          Set the logfile to f. Setting f to an empty string
                    disables logging to file. The WatchedFileHandler is
                    used. This means, the logfile grows indefinitely until
                    an other process (e. g. logrotate or the user itself)
                    move or delete the logfile. Under Windows moving or
                    deleting of open files is impossible and therefore the
```

```
logfile grows indefinitely. default:
/tmp/environment_sensor_5.log
```

## plc - PlasmaLabControl

```
-datalogfile f      Set the datalogfile to f. Only the measurements will be
                    logged here. The WatchedFileHandler is used. This means,
                    the logfile grows indefinitely until an other process
                    (e. g. logrotate or the user itself) move or delete the
                    logfile. Under Windows moving or deleting of open files
                    is impossible and therefore the logfile grows
                    indefinitely. default: /tmp/environment_sensor_5.data
-devicename dev     Set the devicename to dev. default: /dev/ESFTGAB745
-sleep s           If communication to device is not possible, sleep s
                    seconds before retrying. default: 3.0
-baudrate n        Set the baudrate to n. default: 9600
-runfile f         Set the runfile to f. If an other process is running
                    with a given pid and reading the same device, the
                    program will not start. Setting f="" will disable this
                    function. default: /tmp/environment_sensor_5.pids
-debug debug_level Set debug level. 0 no debug info; 1 debug to STDOUT
                    (default).
```

Author: Daniel Mohr  
Date: 2013-03-13  
License:

This is a simple command line program to get the data from the environment sensor 5  
[http://www.messpc.de/sensor\\_alphanumerisch.php](http://www.messpc.de/sensor_alphanumerisch.php) . You can plot the logfile with gnuplot.

You need access to the device of the sensor.

For example you can use the following udev rule:

```
ACTION=="add", KERNEL=="ttyUSB*", ATTRS{product}=="TTL232R-3V3", ATTRS{manufacturer}=="FTDI",\
ATTRS{serial}=="FTGAB745", SYMLINK+="ES%s{serial}", GROUP="users"
```

## 4 Extern Bugs

Here are a few known bugs in extern components we are using or could use instead of the one we are using now:

- Race condition in WatchedFileHandler leads to unhandled exception:

<http://bugs.python.org/issue14632>

- TimedRotatingFileHandler:

<http://thinlight.org/2011/08/10/python-logging-from-multiple-processes/>

Our solution(s)/workaround(s) at the moment:

- `plc_tools.plclogclasses.QueuedWatchedFileHandler`