

Package Automation v1.1.4

Roland Spitzer

roland@familie-spitzer.de

Content

1. [Fli4l as home automation server](#)
 - 1.1. [Start and stop of systems \(group-processing\)](#)
 - 1.2. [Additional commands during start and stop](#)
 - 1.3. [Poweroff by switched power outlet](#)
 - 1.4. [Start and stop of systems \(child-processing\)](#)
 - 1.5. [Don't use hibernate](#)
 - 1.6. [Executing script-files of all kinds](#)
 - 1.7. [Switching power outlets](#)
 - 1.8. [Sending infra-red signals](#)
 - 1.9. [Supported versions](#)
 - 1.10. [News in version v1.0.2](#)
 - 1.11. [News in version v1.0.3](#)
 - 1.12. [News in version v1.1.0](#)
 - 1.13. [News in version v1.1.1/v1.1.2](#)
 - 1.14. [News in version v1.1.3](#)
 - 1.15. [News in version v1.1.4](#)
2. [General](#)
 - 2.1. [Communication using ssh](#)
 - 2.2. [Holding status of automation in files](#)
 - 2.3. [Scripts](#)
 - 2.4. [Used software](#)
 - 2.5. [Can replace other OPT's](#)
 - 2.6. [Many Params](#)
3. [Variables](#)
4. [Configuration-Samples](#)
 - 4.1. [Definitions for the router](#)
 - 4.2. [Windows-Server \(W2K\)](#)
 - 4.3. [Linux-Server \(Debian\)](#)
 - 4.4. [Laptop and streaming client](#)
 - 4.5. [EZcontrol](#)
 - 4.6. [IRTrans](#)
 - 4.7. [AV-Receiver](#)
 - 4.8. [Switched power outlet](#)
 - 4.9. [Logfile-Adapter](#)
5. [Around ssh and netcat](#)
 - 5.1. [Router](#)

- 5.2. [Windows](#)
- 5.3. [Linux](#)
- 5.4. [Windows and netcat](#)
- 6. [Mini-httpd Desktop automation.cgi](#)
- 7. [Description of automation_batch.cgi](#)
- 8. [License](#)

1. Fli4l as home automation server

- 1.1. Start and stop of server, which are needed by other devices. For example start/stop one or more server when the laptop or streaming client is used or stopped. ([look sc200 group](#)). This works by using a magic-package (ether-wake) or switching of the power by using switched power outlets (BIOS-parameter enable boot after power-failure). The stop-commands can get a delay avoiding the stop of clients during temporary boot.
- 1.2. Execute additional commands during start and stop for example “net use” for drives. It is possible to define different commands for start/stop initiated by the automation or manual. For example don't switch the power outlet when manual initiated. ([look pc001 stop](#))
- 1.3. Reducing power consumption during standby by using EZControl and switching power outlets. (I need 17 watt for every pc during standby). The router should be the only system which runs permanently. Also it is possible to use a dedicated system for automation.
- 1.4. Start and stop of server when other systems are not available. For example when all systems are down also stop the fli4l, if needed controlled with time delay (like OPT_AUTOSHUTDOWN). ([look router child](#))
- 1.5. Problems due to unintended stop by windows energy management. Some background processes don't inhibit the activation of hibernation modes. Therefore no control by energy management but rather by using the automation.
- 1.6. Automation of every process executing directly or scheduled by crontab. Every process gets a name like „backup“. With this name the process will be submitted or addressed. For every process a script is generated, which can be submitted by the automation or manual. I use one of these scripts to make unattended backups for all devices.

example : Switch the power on for PC001, start PC001, make some backups on PC001 using rsync, stop PC001 and power off the electricity. This runs scheduled by crontab every day at 03.00 am. ([look pc001 backup](#))

All automation-scripts are placed on the router and can be submitted from outside by any pc using plink.

- 1.7. By using a system like T-10 from Rose and Herleth (www.ezcontrol.de) many other circuits can be realized (i use it for the management of pond- pump or light). I make no promotion for any hardware, i only talk about my experiences with T-

10. I have no other equipment for testing. The use of the program which is needed to send control codes is allowed by R&H and compiled for the router.

1.8. Also I use the IRTrans LAN-Modul (www.irtrans.de) for sending infra-red commands to different devices. Even this modul is compiled for fli4l with permission of the developer. With the irclient I am able to send all learned remote control codes using the automation. That means I can merge infra-red and T-10 commands.

1.9. Supported are all fli4l-versions 3.0.x till 3.1.x

1.10. News in version 1.0.2

- Using fping instead of ping. Fping don't wait 10 seconds for an unreachable device.
- New device-type 0 and 2
- Sleep parameter to influence the router load.
- Make adjustments for fli4l version 3.1
- Diverse optimizations of the scripts (e.g. avoiding subshells by pipes)
- Integrate actorpoll for checking of switched power outlets
- New functions like group_start, group_stop for managing only dependent devices and only_start, only_stop for controlling only the server without influence dependent devices.
- Enhancement of the http-frontend
- Setting some variables in the generated scripts to use them for controlling. HOST contains the actual hostname of the device, FUNKTION the started function and STATUS the current state. Therefore this variables can be used in the conf-file for example in an if-construct.

1.11. News in version 1.0.3

- Integration new parameter AUTOMATION_AUTO_HOSTS. If this parameter is set to "yes" additional hosts are integrated by reading the /etc/hosts.
- Multi-language-support for the web-frontend and an English translation of the documentation. If there is anyone, who can provide other languages I will integrate them. (constants in the lang/-directory and documentation)
- Description of an alternate procedure for windows using netcat instead of ssh.

1.12. News in version 1.1.0

- New device type 3 (logfile-adapter)

- Integrating logfile-adapter for any logfile. The search-command is built by using grep and cut to get the maximal flexibility. Knowledge of these commands is needed, but I gave also some examples. It is possible to define the number of actions per hour or day to prohibit unintended executions of commands. Otherwise a logfile-adapter have the same behaviour like other devices, you can start and stop it and switch it in maintenance mode.

1.13. News in version 1.1.1/1.1.2

- Redesign of different functions for example split automation.cgi into smaller files. Putting start- and stop-functions in own files. Solve some errors.
- New stop-script in rc0.d/rc511.automation to execute some functions during shutdown of the router.
- New function to save the status-files. In this way all states of the managed devices survived the boot.
- Install a „default“-command with an input option in mini-httpd.
- Possibility to switch Logging on or off in a command..
- Extension of the CRON-parameter to execute commands during boot, shutdown and stopping, starting of the automation..
- C-Programm irlogger to receive the IR-Commands of IRTrans and write them into a logfile.

1.14. News in version 1.1.3

- Changes for fli4l v3.1.2. Because the filesystem-layout with 3.1.2 is different, the ssh-files are moved from \$SSH to /etc/ssh abgelegt. In order to be variable a new parameter [SSH](#) contains the right path..
- Considerable Changes are made for the actions boot, shutdown of the router or start, stop the automation.
(assistance and test by T.Albrecht)
- New parameter [AUTOMATION_REFRESH](#) to refresh the host-display in the mini-http.

1.15. News in version 1.1.4

- Changes in mini-httpd views
- Multiple [CRON-Statements in einem Parameter getrennt durch Semikolon](#).
- Neuer [Parameter http](#). Somit kann aus dem mini-httpd in der Host-Anzeige eine beliebige http-Seite für jeden Host hinterlegt und aufgerufen werden.

- Neues [Programm dlogger](#) zur Überwachung von Directories. (Kernel muß mit DNOTIFY yes erstellt werden).
- IRTRANS Parameter entfernt und Start/Stop über automation.txt ([boot/shutdown](#)).
- automation_batch.cgi [mit Status-Abfrage ergänzt](#).

2. General

- 2.1. Communication between devices is handled by ssh. Therefore different tools are used. I made good experiences on windows-systems with the ssh-server copSSH (copSSH can be downloaded from SourceForge.net). I am using key-files without passphrase so that's why no user-id and password is required. When this method seems to be not secure enough, it is possible to generate key-files accepting only the needed command.
- 2.2. All status-information is stored in files. There are two directories. The first one /var/run/automation/pcs. Here exists one file for every pc containing the state, the date and a timestamp of the last change. The second one /var/run/automation/maintenance. Here exists one file for every device which is in maintenance mode..
- 2.3. I have placed the functions in different script-files which are stored in /usr/local/bin/automation. This directory also contains the generated script-files containing the commands for every function.
- 2.4. For stopping of the windows server I use psshutdown.exe from sysinternals.
- 2.5. The package can substitute the packages OPT_AUTOSHUTDOWN and OPT_WOL. This was not my intention. Because these functions were easy to integrate and very important for the automation I decide to consolidate them.
- 2.6. Some advices. I think the OPT isn't easy (there are many parameters, but it can also be used in a simple manner). If you have questions or constructive suggestions please send me a mail. The OPT is quite young and I am thinking about new features like a logfile-adaptor, which starts functions due to records in logfiles, or the connection of receiver like R-10 from EZControl (at the moment not available) or FHZ1000 to get other conditions.
- 2.7. (Doku completed by T.A.)
Executing of scripts depends on a detection of a change.

3. Variables

OPT_AUTOMATION

Activate Automation ?

AUTOMATION_START_AT_BOOT_TIME

Start Automation at boot time ?

AUTOMATION_LOG

Set this parameter to "yes" when logging is required, "no" if not. When logging is activated logfiles are shown on screen (mini-httpd).

AUTOMATION_DEBUG

With "yes" debugging is turned on. Output is also written in the logfile. Much output will be produced in case of debugging.

AUTOMATION_LOGFILE

Absolut path of the logfile. Can be placed on ram-disk, hard-disk or cf-card.

AUTOMATION_WORK

Path for the workfiles, usually '/var/run/automation'.

AUTOMATION_WORK_SAVE (new since v1.1.1)

Path for the backup-workfiles for example '/data/automation' (Harddisk or CF-Card). In this way all states will be saved during reboot.

AUTOMATION_IRTRANS_LOGFILE ([since v1.1.4 moved to boot/shutdown](#))

AUTOMATION_IRTRANS_PORT ([since v1.1.4 moved to boot/shutdown](#))

AUTOMATION_SLEEPV (new since v1.0.2)

Time in seconds waiting after every ping. It reduces the load. When AUTOMATION_SLEEPV='0' the load is equally without peaks. Default with Fli4l-version 3.0.x is (1) for version 3.1 (0). I think this parameter only plays a major role for devices with low performance.

AUTOMATION_SLEEPV (new since v1.0.2)

Time in seconds waiting after every roundtrip (every device is once processed). When the parameter is set and AUTOMATION_SLEEPV='0' the roundtrip is faster with short

peaks. Default is “0“. I think this parameter only plays a major role for devices with low performance.

AUTOMATION_REFRESH (new since v1.1.3)

Time in seconds for refreshing the host-display in mini-http.

AUTOMATION_AUTO_HOSTS (new since v1.0.3)

Completes the monitored devices from /etc/hosts. Default is “no“.

AUTOMATION_HOSTS_N

Number of monitored hosts.

AUTOMATION_HOST_%

Hostname or IP-address of the host.

AUTOMATION_HOST_%_DELAY

Time in seconds. This parameter has two functions. First this is the wait-time when dependent devices (GROUP) are stopped after the primary host is offline. In this way we prevent that devices will be stopped when the primary server only is rebooted or a temporary network-fault occurs. Second this value is used to determine the time to stop for example the router if no host is available (CHILD). Default is “0“.

AUTOMATION_HOST_%_BOOTTIME

Time in seconds. Here is defined the approximately boot-time of thos device. The wait-script waits this time during start/stop before further steps will be executed. If different systems are started concurrent it is possible to check dependencies between them. Default is “0“.

AUTOMATION_HOST_%_HTTP (new since v1.1.4)

Here an [URL](#) can be specified. This URL can be accessed from the Host-View.

AUTOMATION_HOST_%_TYPE (new since v1.0.2)

Different host-types.

- Type 0 will not be pinged or polled. A device which will be not monitored (virtual). I use it for the devices which are managed by infra-red (Trans).
- Type 1 is a normal network-device monitored with fping.
- Type 2 is a switch power outlet (EZcontrol) polled by actorpoll.
- Type 3 is a logfile adapter polled by “ps ax”. (new since v1.1.0)

With introducing of type 0 and 2 it is possible to manage infra-red or power devices like network devices. That also means support of Group, Child, Delay etcetera. With a type 0

– device (virtual) it is possible to simply gather a couple of commands which don't belong directly to a device. Default is "1".

AUTOMATION_HOST_%_TYPE0_DEFAULT (new since v1.0.2)

Used by type 0 – devices. Status of this device is set to „start“ or „stop“ during init of automation.

AUTOMATION_HOST_%_TYPE2_SERVER (new since v1.0.2)

Used by type 2 – devices. Here the device-name or ip-address of the EZcontrol-server for this power outlet is defined.

AUTOMATION_HOST_%_TYPE2_PARM (new since v1.0.2)

Used by type 2 – devices. Number of the switching power outlet EZcontrol (1-20).

AUTOMATION_HOST_%_TYPE3_FILE (new since v1.1.0)

Used by type 3 – devices. Specify the pathname of the logfile for example /var/log/syslog.log

AUTOMATION_HOST_%_GROUP_N

Number of dependent hosts which will be parallel started or stopped. Default is "0".

AUTOMATION_HOST_%_GROUP_%

Name or ip-address of the dependent host.

AUTOMATION_HOST_%_CHILD_N

Number of child hosts. If all child hosts are down the parent will also be stopped after delay time. Default is "0".

AUTOMATION_HOST_%_CHILD_%

Name or ip-address of the child hosts.

AUTOMATION_HOST_%_PORT_N

Additional to the monitoring of child hosts, ports can be checked as well. As long as these ports are active the parent host will not be stopped after delay. Default is "0".

AUTOMATION_HOST_%_PORT_%

Port which is monitored.

AUTOMATION_HOST_%_COMMAND_N

Number of commands for this host. The commands „start“ and „stop“ have a special meaning. Both of them are used during automatic start-/stop-actions controlled by group and child as start- and stop-sequence. Also the commands „manual_start“ and „manual_stop“ have a special meaning. These commands were used by manual actions,

for example in automatic mode stop and switch off the power, in manual mode only stop the device. No own of these four scripts must be defined. If no action is required defaults are used.

For the rest every command-sequence is possible. Because every command or command-sequence produces a normal script-file, placed in the automation-directory, normal bash-syntax and bash-commands can be used (for example. sleep 10 or if-constructs). Fine is that every automation script can be used in every command-sequence. For example in a backup script for many devices we use the predefined „start“ and „stop“-sequences. Default is “0”.

AUTOMATION_HOST_%_COMMAND_% (appendixed since v1.1.1)

Name of the command for example. start, stop or backup.
(New since v1.1.1) is the command “[default](#)“. Like “start“ and “stop“ the command “default“ has a spezial meaning. To reduce the number of definitions it is possible to work with a default-command. The delivered parameter is put into the variable \$VALUE. ([look at the example](#)). If you choose this command in mni-httpd you will be prompted to give a value for the parameter.

AUTOMATION_HOST_%_COMMAND_%_CRON (appendixed since v1.1.1)

Crontab-String for this command if required. Default empty.
(New ab v1.1.1). You can define since version 1.1.1 the following strings „boot“ to execute this commands during boot, „shutdown“ during stop, „automation_stop“ at stopping and „[automation_start](#)“ during start of the automation.

(New since v1.1.4) [multiple Cron-parameter in one statement](#)

AUTOMATION_HOST_%_COMMAND_%_SEARCH (new since v1.1.0)

Search-String for the log processing of type 3 - devices. Here you can define a number of grep- and cut-commands to filter portions of a logfile. See the examples.

AUTOMATION_HOST_%_COMMAND_%_COUNT (new since v1.1.0)

Count-String for the log processing of type 3 - devices. Here you can specify how many actions will be performed by this command. See the examples.

AUTOMATION_HOST_%_COMMAND_%_N

Number of rows belonging to this command. Default is “0”.

AUTOMATION_HOST_%_COMMAND_%_%

Command-row.

(New since v1.1.1) With the command LOG=“no“ or LOG=yes the logging can be temporarily switched off or on. This is helpful when we use longer commands (whole scripts).

An example for a script that should run in the background :

```
AUTOMATION_HOST_1_COMMAND_1_N='7'  
AUTOMATION_HOST_1_COMMAND_1_1='LOG="no"'
```

```
AUTOMATION_HOST_1_COMMAND_1_2='(while [ true ]  
AUTOMATION_HOST_1_COMMAND_1_3='do'  
AUTOMATION_HOST_1_COMMAND_1_4='  fping www.t-online.de >/dev/null  
2>&1'  
AUTOMATION_HOST_1_COMMAND_1_5='  sleep 60'  
AUTOMATION_HOST_1_COMMAND_1_6='  fping www.vodafone.de >/dev/null  
2>&1'  
AUTOMATION_HOST_1_COMMAND_1_7='done) &'
```

Look at the brackets before while and at the end with the ampersand. This script runs in the background.

4. Configuration-Example

Explanations by sample definitions.

4.1. The definitions for the [Router](#) (fli4l) itself.

- A very high delay-time of 86400 seconds (24 hours). Hereby we achieve that the router is switched off until no other device is running.
- All available devices are defined as children. That means when no device is reachable, no traffic on specified ports and the delay-time is reached the router is switched off, too.
- Also the traffic of port 22 (ssh) is monitored.
- As soon as one device is switched on during delay-time the time counter starts at the beginning.

4.2. Samples for a [W2K-Server](#) (pc001)

- The server needs approximately 2 minutes for boot (with a little buffer 180 seconds).
- Three commands are defined. Start works by switching on the power outlet. There are other ways. At the beginning I use ether-wake with the mac-address of the device. Therefore the pc must support wake-on-lan-functions. For the stop I use pssshutdown.exe from www.sysinternals.com. After the pc is down, the switched power outlet ist shut-off. (only with EZControl). Before any pc is switched off, all groups are checked and when this device is used in another group and the server of this group is running, the device remains online.
- The backup can be interested. It is registered in the crontab and runs every day at 3.00 a clock. The defined start-command is used, waiting for online of the device and set maintenance mode so that nothing can disturbe. Then different backups are scheduled using ssh (simple sample). After the backup, maintenance mode off and pc down.

4.3. The [Linux-Server](#)

- By the linux-server is mentionable, that during startup we wait for the windows-server. When booting is ready for both, samba-directories are mounted on the windows-server using “net use” commands. Also we make some mounts for devices where the mount on the linux-system don't work. I think that some mounts come too early (lvm or usb-devices). I have connected some devices at one switched power outlet to power on the server and different usb-connected harddisks together.

4.4. Devices which are started and stopped manually like [Laptop](#) or [Showcenter](#)

- These devices for example the laptop (ethernet or WLAN) needs one or more server for their data (look group-definitions).

4.5. Switching of power outlets with [EZControl](#)

- In this example the compiled EZconsole is used to switch on the power outlet of the windows-server. These commands belongs to the start-/top-sequence of the device pc001. This is only a sample. The automation also works without other hardware, but it was simple to integrate. The advantage of T-10 is to open new possibilities. I use it to switch the pond pump or circulating pump, the pcs and my audio-components. Using EZcontrol (T-10) nice things can be build.

(p.s. in the C'T from 01/2007 is mentioned, that the T-10 is not available. I have bought it two weeks ago, ordered online, no problem. The receiving device R-10 is not available at the moment).

4.6. Transmitting infrared-commands using [IRTrans](#)

- This example shows the integration of IRTrans to submit infrared-commands. This is also a nice possibility to supplement the automation, reducing the great number of remote controls. I have made the whole management easy by gathering the various commands and integrate it in a html-page using automation_batch.cgi. The example shows only a short extract of my definitions (hearing „[radio](#)“). In advance you have to learn the commands of the remote controls (this is described in the documentation of IRTrans). The irclient is compiled for the fli4l allowed by the manufacturer.
- (New since 1.1.1) To receive IR-commands from the IRTrans ethernet with database a C-program is developed, that listen on the specified port (AUTOMATION_IRTRANS_PORT) and writes the Infrared-commands in a logfile (AUTOMATION_IRTRANS_LOGFILE). This logfile can be analyzed using a [Logfile-Adapter](#) (Type 3) and through this it is possible to trigger all automation-function with a remote control. I use this to manage my switched power outlet (EZ-Control) with a remote control. With the command „killall -HUP irlogger“ an event is scheduled, which cause the program to switch the logfile. The program is started when AUTOMATION_IRTRANS_LOGFILE is specified.

- (New since v1.1.4) No start with AUTOMATION_IRTRANS_... parameter, but with Start/Stop commands in the area [boot/shutdown](#).

4.7. Sample type 0-device [AV-Receiver](#)

- This example shows a type 0-device, that can't be monitored (no response). It is a av-receiver which received commands thru infrared and IRTrans. All functions like groups, children can be used. The actual state can't be monitored. It is possible to define all infrared-commands for this device.

4.8. Example for a type 2-device [Switch](#)

- Here is a type 2-device defined. A switched power outlet which is defined at the EZcontrol T1001 with the address number 5. This sample shows that all functions are supported. Switch dependent power outlet schalter1 and schalter2 in parallel. Switch schalter1 off when 2 and 3 are offline with a delay of 20 seconds. The monitoring is implemented by using actorpoll of EZcontrol. The whole thing only works when no manual switching with a separate remote control is used.

4.9. Example for a type 3-device [Logfile](#)

- In this example the syslog is monitored, therefore the hostname "syslog" is used. A logfile-adapter can search any string. Every command (search-string) get a name for example "ip" or "test" and every command can have various command-records, like other devices, too. The processing use the tail-command and starts always at the end of the file. Tail must be installed (**OPT_TAIL**) from the tools-package. When records are written in the log the logfile-adapter gets control and analyse the lines. If the search-pattern matches the defined commands are executed. The type 3 – devices are handled in the httpd like all other devices.
- A sample for a search-command. Assuming the logfile have the following record :

„Feb 18 15:01:37 router dropbear[18158]: Child connection from 192.168.0.40:3389“

and you want to write a record in a file for every connect to the router. Then you must filter those records using **grep "Child connection from"**. Now you want to get the ip-address of the device. This happens with **cut -f 9 -d ' '**. This command separates with blank and filter the ninth word. The result is 192.168.0.40:3389. And now erase the port with **cut -f 1 -d ':'** and ready. When the hole thing is gathered and put in a command using pipe „|“ the search filter is ready to use.

AUTOMATION_HOST_%_COMMAND_%_SEARCH='grep "Child connection from" | cut -f 9 -d " " | cut -f 1 -d ":"'

I recommend to log-on to the router and test the search-filter for example **cat /var/log/syslog.log | 'grep "Child connection from" | cut -f 9 -d " " | cut -f 1 -d**

":". When the result is ok you can put it in the conf-file.

- This string “* 20” is an example for using the count-String. The first value is used for the hour, the second one for the day. The explanation is unlimited processing in the hour and maximal 20 times in a day. Another example is “3 *”. This means only three times per hour and know limit per day. Using this count-string you can delimit the number of processing.

AUTOMATION_HOST_%_COMMAND_%_COUNT='3 *'

- To use the result of the search-String the findings are put in a variable called \$VALUE. In this manner you can use it for example **echo \$VALUE >> /tmp/myfile** to put the results in a file.
- The „start“ and „stop“-commands must be changed. In the start-command the hostname must be specified '**\$BIN/logfile.sh syslog**'. In the stop-command 'kill \$(ps ax | grep "tail" | grep "**/var/log/syslog.log**" | cut -c 1-5)' you have to change the logfile.
- I am sorry, it looks a little bit complicated. But using this search-mode I get the highest flexibility. You can examine with the logfile-adapter any file and submit any command initiated only by a new record in a file.

4.10. Controlling of directories by dlogger (New since v1.1.4)

- With the program irlogger directories can be monitored. All changes (Open/Close/New Files/Dropped Files) in the specified directory are written in a logfiler. This logfile can be monitored by a logfile-adapter and corresponding actions can be executed. Start and stop of the irlogger can be specified in [boot/shutdown](#) similiar to irlogger.

```
Start :      $BIN_ROOT/dlogger /tmp /var/log/dlogger_tmp.log.
           Here the directory /tmp is monitored and all changes are written
in  /var/log/dlogger_tmp.log.
Stop  : /usr/bin/killall -HUP dlogger
           for pid in `ps | grep "dlogger" | grep -v grep | cut -c 1-5`; do kill
$pid; done
```

```
#####
###
### General
#####
###
OPT_AUTOMATION='yes'
AUTOMATION_START_AT_BOOT_TIME='yes'
AUTOMATION_LOG='yes'
AUTOMATION_DEBUG='no'
AUTOMATION_LOGFILE='/var/log/automation.log'
AUTOMATION_WORK='/var/run/automation'
AUTOMATION_WORK_SAVE='/data/automation'
##### AUTOMATION_IRTRANS_LOGFILE='/var/log/irlogger.log'
```

```

#####AUTOMATION_IRTRANS_PORT='21001'
AUTOMATION_REFRESH='15'
AUTOMATION_SLEEPH='1'
AUTOMATION_SLEEPV='0'
AUTOMATION_AUTO_HOSTS='no'
AUTOMATION_HOSTS_N='7'
#####
###
### router (FLI4L)
#####
###
AUTOMATION_HOST_1='router'
AUTOMATION_HOST_1_DELAY='86400'
AUTOMATION_HOST_1_BOOTTIME='120'
AUTOMATION_HOST_1_CHILD_N='7'
AUTOMATION_HOST_1_CHILD_1='pc001'
AUTOMATION_HOST_1_CHILD_2='pc002'
AUTOMATION_HOST_1_CHILD_3='pc004'
AUTOMATION_HOST_1_CHILD_4='pc004w'
AUTOMATION_HOST_1_CHILD_5='sc200'
AUTOMATION_HOST_1_CHILD_6='t1001'
AUTOMATION_HOST_1_CHILD_7='irt01'
AUTOMATION_HOST_1_PORT_N='1'
AUTOMATION_HOST_1_PORT_1='22'
AUTOMATION_HOST_1_COMMAND_N='3'

AUTOMATION_HOST_1_COMMAND_1='start'
AUTOMATION_HOST_1_COMMAND_1_N='1'
AUTOMATION_HOST_1_COMMAND_1_1='ether-wake 00:B0:D0:A9:E1:91'

AUTOMATION_HOST_1_COMMAND_2='stop'
AUTOMATION_HOST_1_COMMAND_2_N='1'
AUTOMATION_HOST_1_COMMAND_2_1='shutdown -h now'
AUTOMATION_HOST_1_COMMAND_3='ping'

AUTOMATION_HOST_1_COMMAND_1='boot'
AUTOMATION_HOST_1_COMMAND_1_CRON='boot'
AUTOMATION_HOST_1_COMMAND_1_N='1'
AUTOMATION_HOST_1_COMMAND_1_1='$BIN_ROOT/irlogger 21001
/var/log/irlogger.log'
AUTOMATION_HOST_1_COMMAND_1_2='$BIN_ROOT/dlogger /tmp
/var/log/dlogger_tmp.log'

AUTOMATION_HOST_1_COMMAND_2='shutdown'
AUTOMATION_HOST_1_COMMAND_2_CRON='shutdown'
AUTOMATION_HOST_1_COMMAND_2_N='4'
AUTOMATION_HOST_1_COMMAND_2_1='/usr/bin/killall -HUP irlogger'
AUTOMATION_HOST_1_COMMAND_2_2='for pid in `ps | grep "irlogger" | grep -v
grep | cut -c 1-5`; do kill $pid; done'
AUTOMATION_HOST_1_COMMAND_2_3='/usr/bin/killall -HUP dlogger'

```

```
AUTOMATION_HOST_1_COMMAND_2_4='for pid in `ps | grep "dlogger" | grep -v  
grep | cut -c 1-5`; do kill $pid; done'
```

```
AUTOMATION_HOST_1_COMMAND_3_CRON='automation_start'  
AUTOMATION_HOST_1_COMMAND_3_N='1'  
AUTOMATION_HOST_1_COMMAND_3_1='fping www.t-online.de >/dev/null 2>&1'  
#####  
###  
### pc001 (Windows 2000)  
#####  
###  
AUTOMATION_HOST_2='pc001'  
AUTOMATION_HOST_2_BOOTTIME='180'  
AUTOMATION_HOST_3_HTTP='http://pc001'  
AUTOMATION_HOST_2_COMMAND_N='3'
```

```
AUTOMATION_HOST_2_COMMAND_1='start'  
AUTOMATION_HOST_2_COMMAND_1_N='1'  
AUTOMATION_HOST_2_COMMAND_1_1='$BIN/command.sh t1001  
Schalter_01_On'
```

```
AUTOMATION_HOST_2_COMMAND_2='stop'  
AUTOMATION_HOST_2_COMMAND_2_N='3'  
AUTOMATION_HOST_2_COMMAND_2_1='ssh -i $SSH/router_rsa -l administrator  
pc001 D:/Start_Stop/psshutdown.exe -k -t 5'  
AUTOMATION_HOST_2_COMMAND_2_2='$BIN/wait.sh pc001 off'  
AUTOMATION_HOST_2_COMMAND_2_3='$BIN/command.sh t1001  
Schalter_01_Off'
```

```
AUTOMATION_HOST_2_COMMAND_3='backup'  
AUTOMATION_HOST_2_COMMAND_3_CRON='00 03 * * *,00 12 * * *'  
AUTOMATION_HOST_2_COMMAND_3_N='6'  
AUTOMATION_HOST_2_COMMAND_3_1='$BIN/command.sh pc001 start'  
AUTOMATION_HOST_2_COMMAND_3_2='$BIN/wait.sh pc001 on'  
AUTOMATION_HOST_2_COMMAND_3_3='$BIN/status.sh pc001 maint_on'  
AUTOMATION_HOST_2_COMMAND_3_4='ssh -i $SSH/router_rsa -l administrator  
pc001 D:/Rsync-Backup/PC001/PC001_Doku.bat'  
AUTOMATION_HOST_2_COMMAND_3_5='$BIN/status.sh pc001 maint_off'  
AUTOMATION_HOST_2_COMMAND_3_6='$BIN/group.sh pc001 stop'  
#####  
#####  
### pc002 (Linux)  
#####  
#####  
AUTOMATION_HOST_3='pc002'  
AUTOMATION_HOST_3_BOOTTIME='180'  
AUTOMATION_HOST_3_COMMAND_N='2'
```

```
AUTOMATION_HOST_3_COMMAND_1='start'  
AUTOMATION_HOST_3_COMMAND_1_N='5'  
AUTOMATION_HOST_3_COMMAND_1_1='ether-wake 00:0C:76:57:A6:C9'
```



```
AUTOMATION_HOST_3_COMMAND_1_2='$BIN/wait.sh pc002 1'
AUTOMATION_HOST_3_COMMAND_1_3='$BIN/wait.sh pc001 1'
AUTOMATION_HOST_3_COMMAND_1_4='ssh -i $SSH/router_rsa -l administrator
pc001 D:/Start_Stop/PC002_Mount.bat'
AUTOMATION_HOST_3_COMMAND_1_5='ssh -i $SSH/router_rsa -l administrator
mount /dev/sdb1 /Test'
```

```
AUTOMATION_HOST_3_COMMAND_2='stop'
AUTOMATION_HOST_3_COMMAND_2_N='3'
AUTOMATION_HOST_3_COMMAND_2_1='ssh -i $SSH/router_rsa -l administrator
pc001 D:/Start_Stop/PC002_UMount.bat'
AUTOMATION_HOST_3_COMMAND_2_2='sleep 5'
AUTOMATION_HOST_3_COMMAND_2_3='ssh -i $SSH/router_rsa -l root pc002
shutdown -h now'
```

```
#####
#####
```

```
### pc004 (Laptop Ethernet)
```

```
#####
#####
```

```
AUTOMATION_HOST_4='pc004'
AUTOMATION_HOST_4_DELAY='10'
AUTOMATION_HOST_4_BOOTTIME='180'
AUTOMATION_HOST_4_GROUP_N='1'
AUTOMATION_HOST_4_GROUP_1='pc001'
```

```
#####
#####
```

```
### pc004w (Laptop WLAN)
```

```
#####
#####
```

```
AUTOMATION_HOST_5='pc004w'
AUTOMATION_HOST_5_DELAY='10'
AUTOMATION_HOST_5_BOOTTIME='180'
AUTOMATION_HOST_5_GROUP_N='1'
AUTOMATION_HOST_5_GROUP_1='pc001'
```

```
#####
#####
```

```
### sc200 (Showcenter)
```

```
#####
#####
```

```
AUTOMATION_HOST_6='sc200'
AUTOMATION_HOST_6_DELAY='40'
AUTOMATION_HOST_6_BOOTTIME='60'
AUTOMATION_HOST_6_GROUP_N='2'
AUTOMATION_HOST_6_GROUP_1='pc001'
AUTOMATION_HOST_6_GROUP_2='pc002'
```

```
#####
#####
```

```
### t1001 (Schalter)
```

```
#####
#####
```

```
AUTOMATION_HOST_7='t1001'
```

AUTOMATION_HOST_7_COMMAND_N='2'

AUTOMATION_HOST_7_COMMAND_1='Schalter_01_On'

AUTOMATION_HOST_7_COMMAND_1_N='1'

AUTOMATION_HOST_7_COMMAND_1_1='/usr/local/bin/EZconsole t1001 -p 2 255'

AUTOMATION_HOST_7_COMMAND_2='Schalter_01_Off'

AUTOMATION_HOST_7_COMMAND_2_N='1'

AUTOMATION_HOST_7_COMMAND_2_1='/usr/local/bin/EZconsole t1001 -p 2 0'

#####

irt01 (IRTrans)

#####

AUTOMATION_HOST_7='t1001'

AUTOMATION_HOST_7_COMMAND_N='4'

AUTOMATION_HOST_7_COMMAND_1='kathrein_power'

AUTOMATION_HOST_7_COMMAND_1_N='1'

AUTOMATION_HOST_7_COMMAND_1_1='/usr/local/bin/irclient irt01 kathrein
power'

AUTOMATION_HOST_7_COMMAND_2='receiver_poweron'

AUTOMATION_HOST_7_COMMAND_2_N='1'

AUTOMATION_HOST_7_COMMAND_2_1='/usr/local/bin/irclient irt01 receiver
poweron'

AUTOMATION_HOST_7_COMMAND_3='receiver_dbs'

AUTOMATION_HOST_7_COMMAND_3_N='1'

AUTOMATION_HOST_7_COMMAND_3_1='/usr/local/bin/irclient irt01 receiver dbs'

AUTOMATION_HOST_7_COMMAND_4='radio'

AUTOMATION_HOST_7_COMMAND_4_N='3'

AUTOMATION_HOST_7_COMMAND_4_1='\$BIN/command.sh irt01 kathrein_power'

AUTOMATION_HOST_7_COMMAND_4_2='\$BIN/command.sh irt01

receiver_poweron'

AUTOMATION_HOST_7_COMMAND_4_3='\$BIN/command.sh irt01 receiver_dbs'

#####

AV-Receiver

#####

AUTOMATION_HOST_8='3805'

AUTOMATION_HOST_8_TYPE='0'

AUTOMATION_HOST_8_COMMAND_N='4'

AUTOMATION_HOST_8_COMMAND_1='start'

AUTOMATION_HOST_8_COMMAND_1_N='1'

AUTOMATION_HOST_8_COMMAND_1_1='/usr/local/bin/irclient irt01 3805 poweron'

AUTOMATION_HOST_8_COMMAND_2='stop'

AUTOMATION_HOST_8_COMMAND_2_N='1'
AUTOMATION_HOST_8_COMMAND_2_1='/usr/local/bin/irclient irt01 3805 poweroff'

AUTOMATION_HOST_8_COMMAND_3='tv'
AUTOMATION_HOST_8_COMMAND_3_N='1'
AUTOMATION_HOST_8_COMMAND_3_1='/usr/local/bin/irclient irt01 3805 tv'

AUTOMATION_HOST_8_COMMAND_4='default'
AUTOMATION_HOST_8_COMMAND_4_N='1'
AUTOMATION_HOST_8_COMMAND_4_1='/usr/local/bin/irclient irt01 3805
\$VALUE'

Schaltsteckdosen Nr. 1

#####

AUTOMATION_HOST_9='schalter1'
AUTOMATION_HOST_9_DELAY='20'
AUTOMATION_HOST_9_TYPE='2'
AUTOMATION_HOST_9_TYPE2_SERVER='t1001'
AUTOMATION_HOST_9_TYPE2_PARM='5'
AUTOMATION_HOST_9_GROUP_N='2'
AUTOMATION_HOST_9_GROUP_1='schalter2'
AUTOMATION_HOST_9_GROUP_2='schalter3'
AUTOMATION_HOST_9_CHILD_N='2'
AUTOMATION_HOST_9_CHILD_1='schalter2'
AUTOMATION_HOST_9_CHILD_2='schalter3'
AUTOMATION_HOST_9_COMMAND_N='2'

AUTOMATION_HOST_9_COMMAND_1='start'
AUTOMATION_HOST_9_COMMAND_1_N='1'
AUTOMATION_HOST_9_COMMAND_1_1='/usr/local/bin/EZconsole t1001 -p 5 255'

AUTOMATION_HOST_9_COMMAND_2='stop'
AUTOMATION_HOST_9_COMMAND_2_N='1'
AUTOMATION_HOST_9_COMMAND_2_1='/usr/local/bin/EZconsole t1001 -p 5 0'

syslog Logfile-Adapter

#####

AUTOMATION_HOST_10='syslog'
AUTOMATION_HOST_10_TYPE='3'
AUTOMATION_HOST_10_TYPE3_FILE='/var/log/syslog.log'
AUTOMATION_HOST_10_COMMAND_N='4'

AUTOMATION_HOST_10_COMMAND_1='start'
AUTOMATION_HOST_10_COMMAND_1_N='1'
AUTOMATION_HOST_10_COMMAND_1_1='\$BIN/logfile.sh syslog'

AUTOMATION_HOST_10_COMMAND_2='stop'

```

AUTOMATION_HOST_10_COMMAND_2_N='1'
AUTOMATION_HOST_10_COMMAND_2_1='kill $(ps ax | grep "tail" | grep
"/var/log/syslog.log" | cut -c 1-5)'

AUTOMATION_HOST_10_COMMAND_3='ip'
AUTOMATION_HOST_10_COMMAND_3_SEARCH='grep "139.7.30.125:53" | cut -f 8
-d " " | cut -f 1 -d ":"'
AUTOMATION_HOST_10_COMMAND_3_COUNT='* *'
AUTOMATION_HOST_10_COMMAND_3_N='1'
AUTOMATION_HOST_10_COMMAND_3_1='echo "$VALUE" > /tmp/ip.tmp'

AUTOMATION_HOST_10_COMMAND_4='test'
AUTOMATION_HOST_10_COMMAND_4_SEARCH='grep "cpmvrmllog_check" | cut -
c 1-40'
AUTOMATION_HOST_10_COMMAND_4_COUNT='3 *'
AUTOMATION_HOST_10_COMMAND_4_N='1'
AUTOMATION_HOST_10_COMMAND_4_1='echo "$VALUE" >> /tmp/test.tmp'
#####
#####
# irtrans Logfile-Adapter
#####
#####
AUTOMATION_HOST_24='irtrans'
AUTOMATION_HOST_24_TYPE='3'
AUTOMATION_HOST_24_TYPE3_FILE='/var/log/irlogger.log'
AUTOMATION_HOST_24_COMMAND_N='3'

AUTOMATION_HOST_24_COMMAND_1='start'
AUTOMATION_HOST_24_COMMAND_1_N='1'
AUTOMATION_HOST_24_COMMAND_1_1='$BIN/logfile.sh irtrans'

AUTOMATION_HOST_24_COMMAND_2='stop'
AUTOMATION_HOST_24_COMMAND_2_N='1'
AUTOMATION_HOST_24_COMMAND_2_1='kill $(ps ax | grep "tail" | grep
"/var/log/irlogger.log" | cut -c 1-5)'

AUTOMATION_HOST_24_COMMAND_3='command'
AUTOMATION_HOST_24_COMMAND_3_SEARCH='grep "21000" | cut -f 4 -d ":" |
cut -f 2- -d " " | grep " "'
AUTOMATION_HOST_24_COMMAND_3_COUNT='* *'
AUTOMATION_HOST_24_COMMAND_3_N='1'
AUTOMATION_HOST_24_COMMAND_3_1='$BIN/command.sh $VALUE'
##### Ende
#####

```

5. Some things around ssh

Examples for a ssh-environment. I am using rsa-keys. The whole thing is a little bit tricky.

5.1. Router

- Create new hostkeys for the router as described in the documentation of OPT_SSHD (SSHD_CREATEHOSTKEYS). Put these keys in the installation-directory `..\etc\ssh`. I work with no passphrase for avoiding the input of the access key. The keys must be kept carefully.
- The server uses the public-keys, the client-side retains the private-keys. On the server-side all valid public-keys belonging to this user are stored in a file called `authorized_keys`. This is handled under windows with an editor under linux with the `cat`-command, to put the generated key in the `authorized_keys`-file.
- In order to submit commands to other devices the ssh-server (f141) needs the public-keys of the target systems. This is done by copying these public-keys also to the directory `..\etc\ssh` and have to be defined in the config-file `sshd.txt` as (`SSHD_PUBLIC_KEYFILES_N` and `SSHD_PUBLIC_KEYFILE_X`).
- Also I have accepted the `known_hosts` to `../etc/ssh` as described under (OPT_SSH_CLIENT).

5.2. Windows

- copSSH as SSH-Server for Windows-Systems from sourceforge.net.
- After the installation of copSSH-Servers on the windows-system, a user must be defined. Here copSSH recommends the available windows-user. This user is used for the execution of the ssh-commands from the router. Therefore this user needs the corresponding rights.
- After that a copSSH-directory exists like `\copSSH\home\User\.ssh`. In this directory all public keys of the clients are placed which connects to this ssh-server using this user. These keys from the directory `\copSSH\home\User\.ssh` must be gathered with an editor in a file named `authorized_keys`.
- Also the rsa-public key generated by copSSH in `\copSSH\etc\ssh_host_rsa_key.pub` must be placed in the router-directory `..\etc\ssh` and be copied as described under 5.1.

5.3. Linux

- Generating keys under linux with `ssh-keygen -t rsa -b 2048 ~/rsa_key`. All questions confirm with return. The output are two files `rsa_key` and `rsa_key.pub`. Again this generated public-key must be placed on the router in `..\etc\ssh` (look at 5.1.).

5.4. Windows and netcat

- A possible alternative to ssh in the windows-environment can be the use of netcat. That doesn't come up to the need of security like the ssh-solution but is easier to implement.
 - Download netcat for windows from www.vulnwatch.org/netcat/ and install. Only nc.exe is needed. With nc.exe -h you get a short description.
 - Use netcat on fli4l from the tools-package.
 - Make some entries in the windows-registry under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run as a character-String for example
 1. String -> **automation_shutdown** Value -> **C:\Programme\nc.exe -d -l -p 4711 -t -e "C:\Programme\psshutdown.exe -k -t 10"** to shutdown the system using psshutdown when port 4711 is addressed.
 2. String -> **automation_backup** Value -> **C:\Programme\nc.exe -d -L -p 4712 -t -e "C:\WINDOWS\system32\cmd.exe /c backup.bat"** to start a bat-file nes bat-files when port 4712 is addressed. (The parameter -L allow multiple calls of the bat-file).
 - Now you can stop the device by sending the command "netcat pc002 4711" in the automation.
-

6. Mini-httpd script automation.cgi

The application for the httpd is self-explanatory. Here the OPT-AUTOMATION can be stopped and started and the logfiles can be viewed. Also the parameter are shown and the state of the devices are displayed with different colours. All commands of all devices are shown and can be submitted. Devices in maintenance-mode don't get automation-commands.

7. Beschreibung des automation_batch.cgi

There is another cgi-script, that can be used by a browser to submit commands for example

http://router/automation_batch.cgi?action=pc001,stop,back
[http://router/automation_batch.cgi?action=pc001,stop,back.](http://router/automation_batch.cgi?action=pc001,stop,back)

This modul is build to make html-pages with embedded commands (without logon). The parms are the destination-device and the command from the conf-file and the appendix back = return, echo = output the information and quiet = no output. Next I will build some html-pages which are managed by a Sony PSP or Nokia 770 connected by WLAN. For the back-parameter Java-Script must be enabled in the browser. For starting an automation-script from a pc I use another method using plink, for example

```
C:\Programme\PuTTY\plink.exe -load router "exec /usr/local/bin/automation/command.sh  
pc001 stop"
```

(New since v1.1.4) Installation of a status-query.

```
status=`$BIN/status.sh $host status`  
case $status in  
  0)   status_text="off" ;;  
  1)   status_text="on" ;;  
  2)   status_text="starting" ;;  
  3)   status_text="stopping" ;;  
  4)   status_text="group_wait" ;;  
  5)   status_text="parent_wait" ;;  
  6)   status_text="parent_wait" ;;  
  9)   status_text="init" ;;  
  10)  status_text="manual_off" ;;  
  11)  status_text="manual_on" ;;  
  12)  status_text="manual_starting" ;;  
  13)  status_text="manual_stopping" ;;  
  14)  status_text="manual_group_wait" ;;  
esac
```

8. License

Copyright 2006-2007 Roland Spitzer ()

This program is free software. You can use it under the conditions of the GNU General Public License released from the Free Software Foundation, pass on and/or modified either under version 2 of the license or (when you like) any future version.

The publishing of this program takes place in the hope that it can help you, but WITHOUT ANY GUARANTEE - even without the implicit warranty of the MARKET MATURITY or the USABILITY FOR A SPECIFIC INTENTION. Details you can find in the GNU General Public License. A version is shipped with the basic fli4l-package.

Also the text of the GNU General Public License is published under <http://www.gnu.org/licenses/gpl.txt>. An official german translation is placed in <http://agnes.dida.physik.uni-essen.de/~gnu-pascal/gpl-ger.html>. This translation should only be used to get a better understanding of the GPL, legally binding is only the english version.

http://router/automation_batch.cgi?action=pc001,stop,back

Roland Spitzer roland@familie-spitzer.de