
g7ctrl

Table of Contents

g7ctrl 2

Name

g7ctrl — Control daemon for a GM7 GPS Tracker

Synopsis

g7ctrl [options...]

Description

The g7ctrl(1) is a daemon that is used to manage a GM7 GPS tracker. The daemon has two main functions. One) it can receive incoming location messages on a specified TCP/IP port that the tracker sends and Two) It allows easy controlling of a GM7 tracker that is connected via USB to the server. Connecting the device with USB to the server is the easiest way to configure the tracker. The daemon defines a command set that is easier for the user compared with the native GM7 tracker protocol. It does not require the user to memorize all commands and their arguments but rather asks in clear text for what settings the user wants to change.

Even though it is perfectly possible to directly connect to the daemon on the specified command port it is recommended to use the supplied command shell g7sh(1). This will offer both full command line editing as well has history function. In addition it hides some of the extra protocol handling on the daemon to make it even more user friendly.

The configuration of the daemon such as what ports to listen on is controlled by the configuration file (by default this is found in `/etc/g7ctrl/g7ctrl.conf`).

Arguments

The following command line arguments are supported when starting the daemon

`-d, --daemon [n/y]`

Determine whether the server should be disconnected from the controlling terminal (a.k.a. running as a daemon). If the server is run as a daemon the logfile cannot be `'stdout'`. This command line option will override the `'daemon'` setting in the configuration file. The default setting is to run as a daemon.

`-h, --help`

Print help

`-i file, --inifile=file`

Use the specified file as configuration file. This will override the location specified at compile time. The default location is `'/etc/g7ctrl/g7ctrl.conf'`

`-l logfile, --logfile=file`

Define the name of the logfile. Two special names can be used, `stdout` which will cause the log to be written to the `stdout` and `syslog` which will cause the logs to be written using the default system logger. This command line option will override the `'logfile'` setting in the configuration file. The default location is `'/var/log/g7ctrl/g7ctrl.log'`

`-p n, --cmdport=n`

Specify what TCP/IP port number that the server will listening on for commands. This will override the `'port'` setting in the ini-file. By default the command port is 3100.

`-s n, --stty=n`

Specify what `ttyACM<n>` device to use when connecting over USB to device

`-t n, --trkport=n`

Specify TCP/IP port number for receiving tracker updates. This will override the setting in the config file. Default port is 3400

`-V n, --verbose`

Log verbosity level. 0 will only log error messages, 1 will log error and info messages, 2 will log errors, info and additional notices finally 3 will also log (possibly) high volume debug messages.

`-v, --version`

Print version information and exit

`---pidfile=file`

Specify PID lock file to be used. By default this is `/var/run/g7ctrl/g7ctrl.pid` so if the daemon is not started as root it must be specified since only root has access to `/var/run`.

`---datadir=directory`

Specify directory for static data files (default, is `/usr/share/g7ctrl`).

`---dbdir=directory`

Specify directory for dynamic data files (default, is `/var/lib/g7ctrl`).

The default arguments corresponds to starting the server with

```
g7ctrl -i "/etc/g7ctrl/g7ctrl.conf" -d Y -s 0 -p 3100 -t 3400 \  
-l "/var/log/g7ctrl/g7ctrl.log" V 2 --pidfile="/var/run/g7ctrl/g7ctrl.pid" \  
--dbdir="/var/lib/g7ctrl" --datadir="/usr/share/g7ctrl"
```

Starting and stopping the server

While strictly speaking not necessary the server should normally be started by the `root` user. As soon as the server doesn't need `root` privileges it will switch to run as the user specified in the configuration file, by default this is the user `g7ctrl` if the configuration file has not been changed. This user should for security reasons not be the 'root' user. This user should be a system user without login shell. Such a user can normally be created by running the command `'useradd'`. (If the daemon is installed via the RPM package this user will be automatically created).

If your system supports the command **useradd** then start by changing to root and add the user as so

```
$> useradd -r g7ctrl
```

Note

The user created must normally belong to the group `dialout` on most system in order to have access to the USB-serial device.

Then to start the server as root using the settings in the configuration file the simplest way is to just call

```
sudo g7ctrl
```

In order to stop the server and allow it to do a clean shutdown the `SIGINT` (or `SIGSTOP`) signal should be sent to the process running the server (assuming **g7ctrl** has been started as a daemon). The distribution includes two support scripts to help with this. This can of course also be done manually as (and reading the pid number from the standard pid-file)

```
sudo kill -SIGINT `cat /usr/share/g7ctrl/g7ctrl.pid`
```

Please note the `"`"` backward slashes.

While testing the server it is also possible to start the daemon as a "normal" user. As a test start the daemon from command line for example as

```
$g7ctrl -dN -V3 -lstdout
```

The commands will start the application on the terminal `-dN` (not as a daemon), with maximum logging (`-V3`) and print the log file directly to the terminal (`-lstdout`). This is a good way to do the initial testing since any

problem can usually be quickly seen in the log. Note that when the daemon is not started as the root user it can not switch to the daemon user specified in the configuration file.

Once the testing is completed the daemon can be setup to be started at the same time as the server. Depending on the system this can be done in a few different ways as the following two sections show.

For system using old style `init.d`

To start the daemon automatically the installation will install two utility scripts. The first script will be installed in `/etc/init.d/g7ctrl` and follows the normal guidelines for a `"init.d"` service start/stopping script according to LSB (Linux Standard Base). Using this script to start the server is then simply a matter of calling

```
sudo /etc/init.d/g7ctrl start
```

(or its alias **rcg7ctrl**) is given. The start-up script also supports the `"restart"` and `"status"` options.

If the server gives the error message : *"Server can not be started. See system log"* then more information about the details of the problem can be found in the system log, normally `/var/log/messages`. The most common reason for this message is that the daemon is already running.

To stop the server give the command (as root)

```
sudo /etc/init.d/g7ctrl stop
```

Since the `init.d` system has been abandoned in most modern system the `init.d` scripts will then be automatically redirected to the corresponding `systemd` service. In that case it is better to just use the `systemd` service directly as the following section shows.

Using `systemd`

If your system makes use of `systemd` then the second utility script will be installed as a `systemd` service. The configuration script will automatically figure out if your system supports this. In this case the daemon can be directly controlled via the usual **systemctl** command as shown below. See **man systemctl** for details.

```
sudo systemctl stop g7ctrl.service
```

```
sudo systemctl start g7ctrl.service
```

Note

The `g7ctrl` service needs to first be enabled after the initial installation with

```
sudo systemctl enable g7ctrl.service
```

If your system does not have the command **systemctl** then it is not using the `systemd` setup.

Using `upstart`

Newer version of Debian and Ubuntu now uses `"upstart"` to manage daemons. The build system fully supports this and after the usual **"sudo make install"** the daemon could be controlled by

```
sudo start g7ctrl
sudo stop g7ctrl
sudo status g7ctrl
```

or alternatively by

```
sudo service g7ctrl start
sudo service g7ctrl stop
```

```
sudo service g7ctrl status
```

Note

The configuration file for upstart is stored in `/etc/init/g7ctrl.conf`

Interfacing with the server

The primary interface to the server is via a TCP/IP connection to the server running '*g7ctrl*' via a dedicated port (by default 3100). The server is controlled via a simple (but efficient) command language which allows manipulation of future recordings and various other administrative tasks. All commands are extensively described in the section 'Command language' below.

The preferred way of interacting with the server on the command level is by using the supplied utility shell '*g7sh*'. This supports history and full command line editing and is recommended over a plain 'telnet' (nc command) or connections to the server.

See the section "Connecting to the server" below for more detailed examples.

Connecting directly to the server

By default the best way to communicate with the daemon is to use the **g7sh** command, see below. But the following information may be used for additional troubleshooting.

We start by describing the direct method that communicates with the daemon directly on the command port. This is for illustrative purpose only and can be a quick way to check that the daemon is really running when troubleshooting. For normal usage it is recommended to use the *g7sh(1)* utility shell program (see also next section).

Once the server is running the easiest way to connect to the command is by using the '*nc*' ('net cat') program. Assuming that the *g7ctrl* server is running on the localhost the following command will connect to the server (assuming the default port 3100 has not been changed (either in the configuration file (e.g. `/etc/g7ctrl.conf`) or by a command line argument).

```
$> nc localhost 3100
```

After a successful connection the server will reply with a short, one line, information message. The full returned information message at the connection time should look something similar to the following (the version and build might be different though):

```
GM7 Server version 3.0.0 (build 150108-110)
```

If password protection (see section "Security" below) the welcome message is not be shown until the correct password has been specified. Instead when password protection is enabled the connection sequence looks like this

```
$> nc localhost 3100
Password:xxxxxxx
GM7 Server version 3.0.0 (build 150108-110)
```

This means that the correct server password has to be given before the welcome message is seen. The server is now ready to accept commands on this connection.

To disconnect to the server give the command '**exit**'

Using the g7sh to connect to the daemon

The package also includes a full command line shell, **g7sh**. This allows much better editing (Emacs compatible editing) as well as full command line history. This shell can also be used as a system login shell and hereby making

it possible to connect to the daemon using **ssh** for a more secure communication by creating a unique user with g7sh as the login shell.

How to setup and use the g7sh command line shell is fully described in g7sh(1)

In its simplest form (assuming it is the same sever as the daemon) the shell can be started with

```
$ g7sh
```

Run **g7sh -h** for a list of supported options.

Directory structure

The daemon uses two distinct directories on the server; one data directory where the database and log files are stored and one configuration directory. The default location of these directories are specified at build but may be overridden by either the configuration file or by command line arguments. The package default locations are shown below

```
/etc/g7ctrl  
    Configurations file.
```

```
/usr/share/g7ctrl  
    This is the data directory where static data files (like mail templates and presets) are stored.
```

```
/var/lib/g7ctrl  
    This is the data directory where dynamic data files (like DB) are stored.
```

Security

In order to prevent unauthorized access to the server the server have a basic password protection. This means that a password can be configured to be used before a client can start giving command to the server. This is configured in the ini-file. The configuration uses the same password for all clients so the server does not support multiple users/multiple password. Since the password is stored in plain text format the configuration file should have restricted access. This should not be considered strong security but rather a weak "stop-messing-with-daemon" type of protection.

Sine the password are sent unencrypted to the daemon it is also susceptible to line snooping.

The setting whether password should be used or not and the actual password is specified in the ini-file.

After connecting to the server when the password has been enabled the first string sent back from the server will be the single string:

```
Password:
```

The rest of the reply is the same as when no password is used.

If security is of concern the daemon can also be configured to only allow connections from the localhost, i.e. connections must be made from the same machine that is running the daemon. This is done by enabling the option "only_localhost" in the config file.

Managing notifications mails

When configuring the daemon it is possible to set options that will send notification mails on important events in the daemon. This could be notifications on errors that occurred but also notifications on that a transcoding was finished or that the server was shutdown automatically while waiting for the next recording.

All mail settings are specified in the "[mail]" section of the configuration file. More information is available in the full application manual.

Commands (in alphabetic order)

The following is a list and detailed explanation of the commands that are recognized by the server.

Since there are quite a few commands we start with an overview with a one line explanation of each command. Further details can be found below

There commands are divided into three different areas

1. Device commands. They are commands that controls a connected device
2. Database commands. Commands to export and read the database
3. Preset commands. Commands to manipulate presets
4. Daemon commands. Commands to act on the daemon itself.

Device Command reference

Below is the commands possible together with there parameters. In case a command can accept argument the daemon will ask for each command in sequence. The prompt will always be indicated by a "?" mark. To abort a specific command just type a single dot "." and the command shell will return to the start position (outer level). Most commands only accept a selected value chosen from a list. All possible values in the lists are shown below.

For even more detailed command information please refer to the "*GM7 Protocol specification*" available from where you purchased your device.

address

Return the current location as an approximate street address

clrec

Erase all recorded locations in device memory. Use `dlrec` to read back the recorded location positions in device memory to the database.

comm

Set/Read device communication type and its parameters

- *CommSelect* - Select primary type of communication

Possible values and description:

- 0 - "Use USB"
- 1 - "Use SMS over GSM"
- 2 - "CSD (reserved)"
- 3 - "UDP over GPRS"
- 4 - "TCP/IP over GPRS"
- *SMS Base Phone* - SMS base number to call
- *CSD Base Phone* - CSD base number (reserved and not used)
- *GPRS APN* - The operators APN
- *GPRS User* - User name if required
- *GPRS Passwd* - Password if required
- *GPRS Server* - Server IP address where the device reports back to

- *GPRS Server Port* - TCP/IP Port to use on server
- *GPRS Keep Alive* - Interval (in sec) between Keep Alive Packets
- *GPRS DNS* - Optional DNS server for device to use.

config

Set/Read device ID, Password, and PIN Code of the SIM card

- *Device ID* - Set device ID (leftmost digit must always be 3)
- *Device Password* - Set device Password (numeric)
- *SIM Card PIN* - The SIM cards PIN code to use

dlrec

Download all recorded position in device memory to the database. Positions that have been downloaded from the device memory in this way will have an event id of 1.

Note

Please note that download and DB update takes a bit of time. As an example download and DB updated of 10,000 locations took 17s in one instance. The bulk of the time, ~95%, is spent downloading the data from the device over the serial line and ~5% was spent to update the DB with the new locations.

geofevt

Enable/Disable/Set/Read GEO-Fencing event

- *Event ID* - Event ID. Maximum of 50 events. In range 50-99
- *Enabled* - Enable/Disable event
- *Longitude* - Longitude for center of event zone
- *Latitude* - Latitude for center of event zone
- *Radius* - Radius of event zone in meters (0-65535)
- *ZoneControl* - How to trigger event

Possible values and description:

- 1 - "Entering zone"
- 2 - "Leaving zone"
- *Action* - What to do on event

Possible values and description:

- 1 - "Log to device"
- 2 - "Send to server"
- 3 - "Both log and send"

gfen

Enable/Disabling GEO-fencing functionality

- *Enabled* - Turn option on/off
- *Radius* - Radius in meter for fence

- *Zone trigger* - Trigger when leaving or entering zone

Possible values and description:

- 1 - "Inside zone"
- 2 - "Outside zone"
- *Report action* - What to do on event

Possible values and description:

- 0 - "Disabled"
- 1 - "Log to device"
- 2 - "Send to server"
- 3 - "Both log and send"
- *VIP mask* - Bitmask of VIP numbers to alert

imei

Query the IMEI number of the internal GSM module

led

Enable/Disable the LED indicator on/off

loc

Get latest location

log

Get the latest position as a Google map link

lowbatt

Set/Read the internal battery low level alert

- *Report action* - Action to perform at low battery voltage

Possible values and description:

- 0 - "Disable"
- 1 - "Log on device"
- 2 - "Send back to server"
- 3 - "Both log and send back ro server"
- *VIP mask* - Bit-mask of VIP numbers to alert

mswitch

Enable/Disable detached report

nrec

Return date range and number of location records stored in device memory

phone

Enable the function "Get the current location by making a phone call"

ps

Enable/Disable power saving mode

- *Mode* - Select when to enable sleep mode to save battery

Possible values and description:

- 0 - "Sleep after 3min of no movement. Wake up on movements. GSM=Standby, GPRS=GPS=Off, G-sensor=On"
- 1 - "Always enter sleep after 3min and wake every n seconds specified. GSM=GPRS=GPS=G-sensor=Off"
- 2 - "Always enter sleep after 3min and wake up on timer. GSM=GPRS=GPS=G-sensor=Off"
- 3 - "Always enter sleep after 3min and wake up on movement. GSM=GPRS=GPS=Off, G-sensor=On"
- *Sleep interval* - Used with sleep mode 1, Interval in minutes between wake-ups
- *Wake up report* - Action when awoken

Possible values and description:

- 0 - "Disable"
- 1 - "Log on device"
- 2 - "Send back to server"
- 3 - "Both log and send back to server"
- *VIP mask* - Bit-mask of VIP numbers to alert on awake
- *Timer 1* - Used with mode=2. 00-23 hr
- *Timer 2* - Used with mode=2. 00-23 hr
- *Timer 3* - Used with mode=2. 00-23 hr

reboot

Restart-up the device

rec

Enable/Disable/read logging function to the device

- *Mode* - When should logging be done

Possible values and description:

- 0 - "Disable"
- 1 - "Time mode, log every n seconds"
- 2 - "Distance mode, log every n meters"
- 3 - "Time AND Distance mode"
- 4 - "Time OR Distance mode"
- 5 - "Heading mode, logging starts when heading > set value"
- 6 - "Heading OR Time mode"
- 7 - "Heading OR Distance"
- 8 - "Heading OR (Time AND Distance)"

- *9* - "Heading OR Time OR Distance"
- *Time interval* - Time interval (in sec) used in mode setting
- *Distance interval* - Distance interval (in meters) used in mode setting
- *Number of reports* - The number of reports to send back on event (0=continuous logging)
- *Record basis* - Logging mode

Possible values and description:

- *0* - "Wait for GPS fix before sending back report"
- *1* - "Don't wait for GPS fix before sending back report"
- *Heading* - Heading value used in mode (10-90)

reset

Reset all parameters to the manufacture default settings

roam

Enable/Disable GPRS roaming function

sim

Query the identification of the SIM card

sleep

Enable/Disable "Sleeping Report"

- *Message mode* - Set how to handle message at when device enters sleep

Possible values and description:

- *1* - "log message to device"
- *2* - "send back message"
- *3* - "both log and send back message"

sms

Switch the SMS format (Text or PDU mode)

- *Mode* - Set format for SMS sent back

Possible values and description:

- *0* - "Use PDU mode"
- *1* - "Use text mode"

stopdl

Stop downloading logging data from the device

test

Device diagnostic function

track

Control if device should be sending back automatic tracking information.

- *Tracking mode* - Set condition for when tracking is started

Possible values and description:

- 1 - "Time mode"
- 2 - "Distance mode"
- 3 - "Time AND Distance mode"
- 4 - "Time OR distance"
- 5 - "Heading mode"
- 6 - "Heading OR time"
- 7 - "Heading OR distance"
- 8 - "Heading OR (Time AND Distance"
- 9 - "Heading OR Time OR Distance"
- *Timer interval* - Used for mode 1
- *Distance interval* - Used for mode 2-9
- *Number of tracking* - Use 0 for continuous tracking
- *Track basis* - Wait for GPS fix or not

Possible values and description:

- 0 - "Send report ONLY if GPS is fixed"
- 1 - "Send report regardless of GPS status"
- *Comm select* - Set type of report

Possible values and description:

- 0 - "Over USB"
- 1 - "Over GSM"
- 2 - "Over Reserved (not used)"
- 3 - "UDP over GPRS"
- 4 - "TCP/IP over GPRS"
- *Heading* - Used for modes 5-9

tz

Set the time zone information for the device

- *Sign* - Ahead or before GMT zone

Possible values and description:

- + - "Ahead of GMT"
- - - "Before GMT"
- *Hour* - Number of hours to adjust (0-9)

- *Min* - Minutes to adjust

Possible values and description:

- 0 - "0 min"
- 15 - "15 min"
- 30 - "30 min"
- 45 - "45 min"

ver

Get current firmware version

vip

Preset up to 5 SMS phone numbers for receiving different alerts

- *VIP 1* - Mobile number 1 (full number incl country prefix '+' if needed)
- *VIP 2* - Mobile number 2 (full number incl country prefix '+' if needed)
- *VIP 3* - Mobile number 3 (full number incl country prefix '+' if needed)
- *VIP 4* - Mobile number 4 (full number incl country prefix '+' if needed)
- *VIP 5* - Mobile number 5 (full number incl country prefix '+' if needed)

Database commands

The database commands are used to manage the stored location updates from the devices.

`db dist [dev=nnn] [ev=nnn] [from to]`

Calculate the **approximate distance** traveled using the selected data set of location points.

- `[dev=nnn]` - Filter on a specific device-id. If not set all devices will be considered.
- `[ev=nnn]` - Filter on specific event. If not set all events will be considered.
- `[from to]` - Filter on a specific date range. Dates are given in YYYY-MM-DD format with optional time specification as HH:MM in 24h format (military time). If omitted all dates are considered.

Calculation of distance is approximation of the distance between two GPS points (assuming WGS-84 projection) and returns the result using both the *Haversine* method (great circle distance) and *Vincenty* method (iterative). The reply from the daemon is given as two lines exemplified by

```
Calculating approximate distance using 344 points.  
25.3 km (alt. 25399.5 m)
```

The second distance in m are calculated using the *Vincenty* method and the first distance (in km) is calculated using the *Haversine* method.

Due to the number of error sources in getting location updates and limitations inherent in this type of distance calculations **we can not in general assume to have perfect accuracy**. The accuracy will also vary depending on the absolute location due to differences in the earth radius. The way traveled between recorded positions is assumed to follow the great circle between the points so the more points the better approximation of the actual real distance.

Note

The precise volatility in the distance figures are very difficult to quantify due to the large number of factors that influence the GPS signal and the way the distance can be calculated. However, practical

tests (with the GM7 device) have shown that in general it is safe to assume that the true distance (as measured in the middle part of Sweden around lat 57°) in more than 80% of the cases have less than 5% error compared with the distance as measured by an odometer. If more accuracy is needed in a particular area then field tests with known distances needs to be performed and correction factors introduced.

Often the distance is slightly underestimated and the longer time interval between each measure the more the distance is underestimated. This is due to the fact that the distance travelled between each measured point is assumed to be the shortest (great circle) distance and in practice the real distance is always longer.

(The *Vincenity* method will generally have better accuracy assuming perfect GPS positions.)

Examples:

- `db dist`

Use all the data points and calculate the total traveled distance

- `db dist ev=1 2013-08-01 2013-08-01`

Calculate the traveled distance from the previously imported data points from the internal memory of the device (they will have eventid=1) for the 1:st of August 2013.

- `db dist 2013-01-01 2013-12-31 13:00`

Calculate the total distance in DB for 2013 regardless of what event triggered the location update

`db export [dev=nnn] [ev=nnn] [from to] (gpx|kml|xml|csv|json) [filename]`

Exports the database in the selected format

- `[dev=nnn]` - Filter on a specific device id. If not set all devices will be considered.
- `[ev=nnn]` - Filter on specific event. If not set all events will be considered.
- `[from to]` - Filter on a specific date range. Dates are given in YYYY-MM-DD format. If omitted all dates are considered.
- `(gpx|kml|xml|csv|json)` - Export format to use
- `[filename]` - Full filename path to export to. If omitted the data will be exported to `/var/lib/g7ctrl/export.<format>`

Examples:

- `db export gpx`

Export all stored location in GPX format to `/var/lib/g7ctrl/export.gpx`

- `db export dev=3000000001 2013-02-01 2013-02-28 xml /tmp/myexport.xml`

Export all location reported in February for device 3000000001 in g7ctrl own XML format to the file `/tmp/myexport.xml`

`db lastloc`

Display the latest recorded position

`db mailpos`

Mail the last recorded position

`db mailgpx`

Mail a compressed version of a GPX export of the entire DB to the predefined mail address.

db mailsrv

Mail a compressed version of a CSV export of the entire DB to the predefined mail address.

Preset commands

The preset commands are used to manage the stored commands in the presets

preset list

List all define presets

preset refresh

Re-read all presets from disk

preset use <preset-name>

Execute the specified preset. A shortcut for this is "@<preset-name>"

preset help <preset-name>

Print help for the specified preset

Daemon commands

Daemon commands are internal commands to work with the daemon itself.

.ver

Display the version of the daemon

.dev

Show if the device is connected or not

.lc

List active command connections

.ld

List all connected devices, both over GPRS and on USB

.target <dev>

List or set device connected over GPRS to receive commands

.usb <dev>

List or set device connected on USB to receive commands

.nick

Specify a new nick name for device

.ln

List all defined nick names

.dn

Delete the specified nick

Files

Note: The system configuration directory to be used is a compile time settings and specified with the '*configuration*' argument '--sysconfdir'.

/var/log/g7ctrl/g7ctrl.log

The logfile. The log-directory is specified in the ini-file. It can also be specified as a command line argument (-l, --logfile) which will then override any config file setting. By default the log file will be stored under /var/log/g7ctrl/g7ctrl.log.

`/var/run/g7ctrl/g7ctrl.pid`

The pid of the started 'g7ctrl' server. This is used to easily find the pid in order to shutdown the server. This is used by the *stop-g7ctrl* script. Note that in order to use this directory the daemon has to be started by the root user.

`/etc/g7ctrl/g7ctrl.conf`

Default configuration file

`/var/lib/g7ctrl/gm7tracker_db.sqlite3`

The sqlite3 database where all locations are stored..

Limitations

- All information, error messages and loggings are hard coded in English.

Bugs

- The configuration file has too many settings.

Notes

Any errors that are discovered in the start-up phase before a logfile has been established will be written to the default system log (normally `/var/log/messages` or `/var/log/syslog`)

Resources

- SourceForge: <<http://sourceforge.net/projects/g7ctrl/>>

Author

Written and maintained by Johan Persson <<johan162@gmail.com>>

Copyright

Free use of this software is granted under the terms of the GNU General Public License (GPL v3), see the file `COPYING` included in the distribution archive for details.

See also

`g7sh(1)`