PC App User Manual

LACERTA MGEN-3

for PC App Version 1.1

Table of Contents

1. About PC App.....2
    1.1. The main window.....3
2. What is the API Server?.....4
    2.1. About TCP/IP configuration.....6
    2.2. Using the official Windows Applications.....7
        2.2.1. Direct connection.....8
        2.2.2. API Server connection.....10
    2.3. How to set up an API Server.....11
        2.3.1. Setting the conn. options for the server.....11
        2.3.2. Start the server.....12
        2.3.3. Stopping the server.....14
        2.3.4. Behaviour.....14
3. Settings of PC App.....16
4. FileSystem access and Log file processing.....17
    4.1. FileSystem part.....17
        4.1.1. Downloading files.....19
    4.2. Log file processing part.....21
        4.2.1. Processing log files.....21
        4.2.2. Output files.....22
    4.3. The “guided exposure” output.....23
        4.3.1. If DEC axis was disabled.....25

1. About PC App

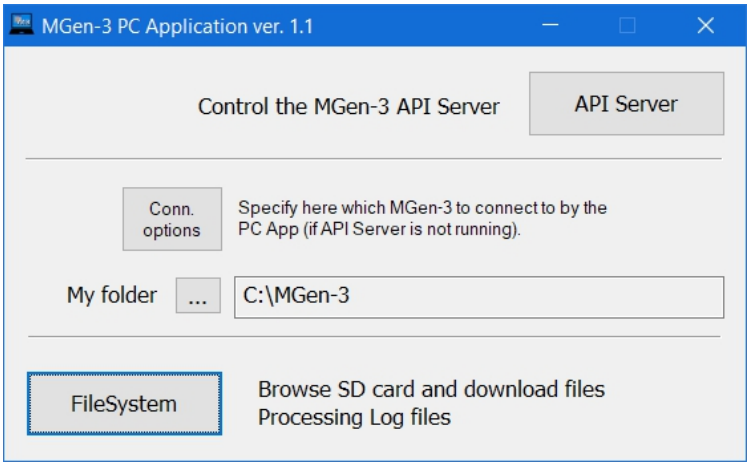
This application is an official main program for Windows to provide specific functions and sub-programs for the MGen-3 AutoGuider. Other official Windows apps. doing separate and dedicated functions that may be run in parallel with other apps. are released as separate executable.

What the current version 1.1 of PC App offers:

- Hosting / implementing an API Server.
  You can set up and run an API Server to enable serving of more client applications for concurrent access of a MGen-3 HC.

- Browse the SD card on the HC and download files. (Sub-program.)
  Make local copies onto your PC of the files stored on the SD card plugged into the MGen-3 HC.
  You can also process the downloaded log files for guided exposure info (graphs and data extracted).

1.1. The main window



This is the main window of the PC App. It is separated into sections by the separate functions offered:

- At the top there is the control dialog box available for the API Server.
  Details are at this section of the manual.

- On the middle you can set the connection options of the provided functionalities of PC App, and specify your local MGen-3 directory in which files are stored and loaded from.
  Details are at this section of the manual.

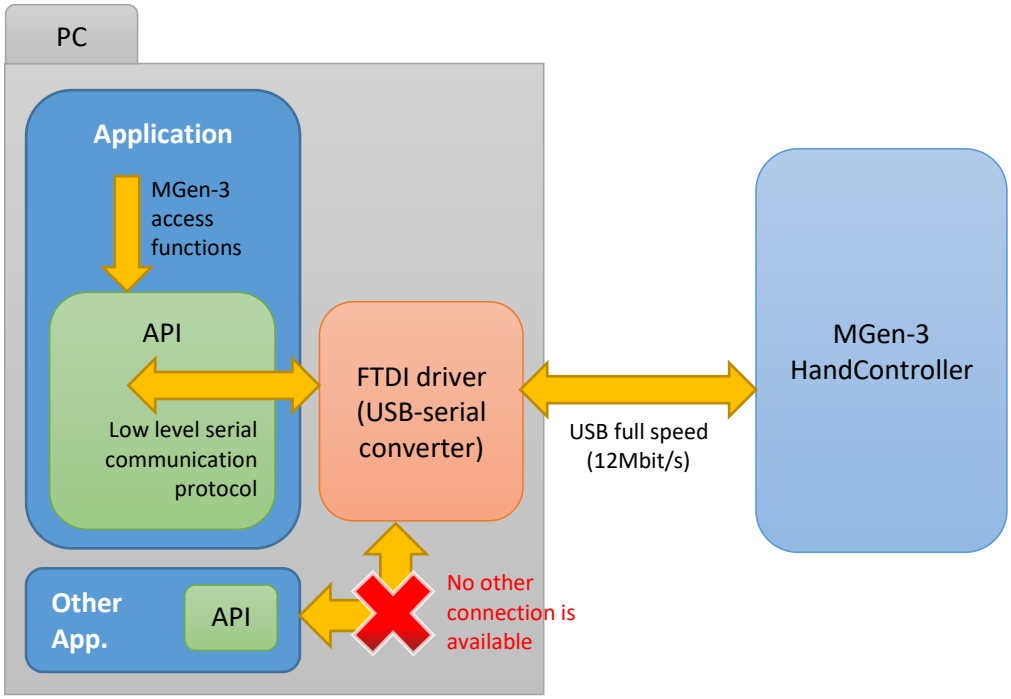
- At the bottom there are different sub-programs available. Currently the only one is the FileSystem's access (and log file processing) that enables browsing the SD card plugged into the HC and make copies of the files onto your local MGen-3 directory and the ability to create output with graphs for the guided exposures by the log files.
  Details are at this section of the manual.

2. What is the API Server?

The abbreviation “API” stands for “Application Programming Interface”. Devices can be accessed and controlled from an external device or program through their Application Programming Interfaces available. This is done the same way for LACERTA MGen-3 HandController (HC).

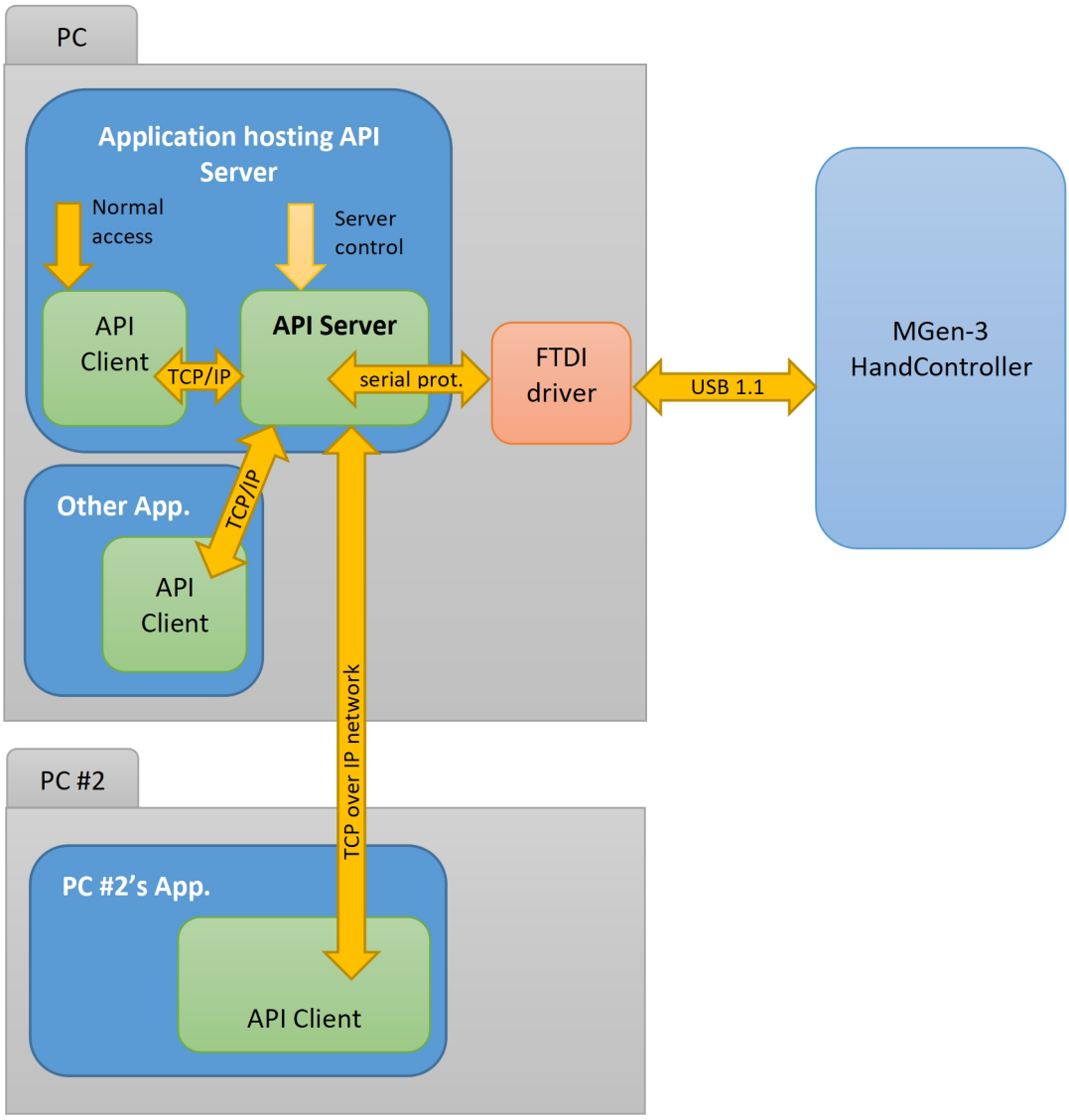
Normally when you start an application running on your computer (PC) that can connect to a MGen-3 HandController via USB, the connection is opened to the HC using the MGen-3 API. Users should not care about the API and the underlying layers, only use the application for some kind of work.

Before the new API (that contains the API Server mode) is introduced, an application could connect to a HC only directly, which means that the application allocates the serial comm. channel to the HC (using the FTDI chips’ diver) and no other program (using the same API) is able to connect to or even to detect the HC plugged into the PC. The block diagram below shows the connection and comm. channels for this case:



This is the point when the API Server comes in view. It is able to provide an extended interface for the applications that allows multiple applications being connected to the same HC at the same time without disturbing each other’s functionality, if used properly by the Apps.

In practice this means that setting up an API Server by the new API and the applications also using the new API the user can run e.g. the SkyEmulator, the Virtual UI and other main control applications at the same time. The abilities of this structure is shown below on the block diagram:



Each green API block represents an MGen-3 API instance created by the Applications. There are three different API modes for this instances in the new API:

- Direct (normal) mode: the API connects directly to the FTDI driver and allocates the control only for itself (to a MGen-3 HC instance).

- Server mode: the API server works as the Direct mode from the view of connecting to a MGen-3 HC but also creates a TCP server, where multiple API Clients can connect to and will be served to control the connected HC.

- Client mode: the API Client connects to an API server through the TCP/IP stack of the PC, that can be from the same PC or from any PC over the world that can be reached with IP (Internet Protocol). The HC accessed will be the one that is connected to the API Server.

The API Server handles the exclusive accesses for the Clients so they won't mess up the single low level communication protocol. The Server also introduces "section locking", which means the usage of a given set of API functions will lock the access for only one Client and

the others can't issue such commands until that Client releases it. This is also essential for some function sets to overcome disturbance/control interference between the Clients.

The "Server control" arrow on the diagram shows that the hosting application controls the Server in the mean of start/stop/connect etc. and of course it calls the API Server's "Run" function periodically, which processes the data streams from/to the clients and does other basic operations.

When the user has set up an API Server using an App. that supports it, the Clients (Apps) can be freely started and stopped to control the HC that is connected (or supposed to be connected) to the API Server. An advantage of the API Server is that the API Client versions can be newer or older than the API Server, all new functions will be available for the Client (except some very special ones). So the API Clients and their Apps connected to the Server can differ in version, this should not cause trouble.

The function set of the API don't change much in the new version, only a few cases must be handled different by the Apps using it, so adapting the Apps to the new API modes must be done easy by the app. developers.

2.1. About TCP/IP configuration

The TCP (Transmission Control Protocol) is a bidirectional stream-based connection protocol that works over IP (Internet Protocol). When we talk about a TCP connection it can be between two applications running on the same computer or between any computers that are connected together by the global internet.

The server-client structure comes from the TCP itself. A server must be initiated that is waiting for connection of clients and after accepting them and initializing their protocol, normal communication can take place in parallel for each client.

After starting the Server creates a TCP "listening port", where the Client apps. can connect to. Such a TCP port is addressed as an IP address (or domain name) and a port number. For example on a local network: (let the MGen-3 device's serial ID be "M3S00195")

- The PC is already connected to the network and so has an IP address, for example 192.168.0.73. An application hosting the API Server is started with the default port number (that is 5937) and now it is waiting for clients connecting to this port. This server is set up to open the serial data connection to the HC "M3S00195" or is already opened at initialization time.

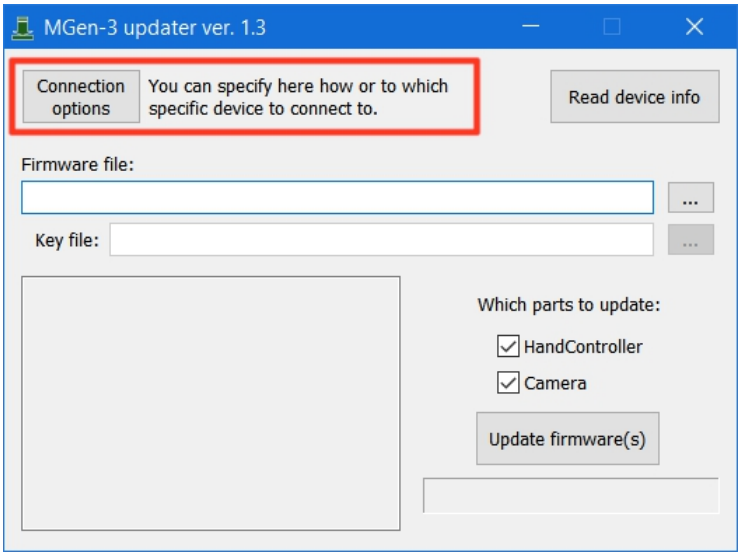
- An application that can use the API Client mode wants to open the connection to the given MGen-3. It knows that the HC "M3S00195" is behind a running API Sever, so in contrast to the direct mode (when the API searches for direct serial connection to "M3S00195") the app. should address the server's listening TCP port opened when opening the connection to the "M3S00195" MGen-3 device. So the application connects as a client to "192.168.0.73 : 5937".

- If succeeds, the application using the API Client can call all API functions to control the HC, just as if it opened it in direct mode (as by the earlier APIs). Only slight extensions or modifications (handling new error codes) must be carried out on the application using the API Client, only to know about problems that occur over the TCP/IP (client-server) level.

2.2. Using the official Windows Applications

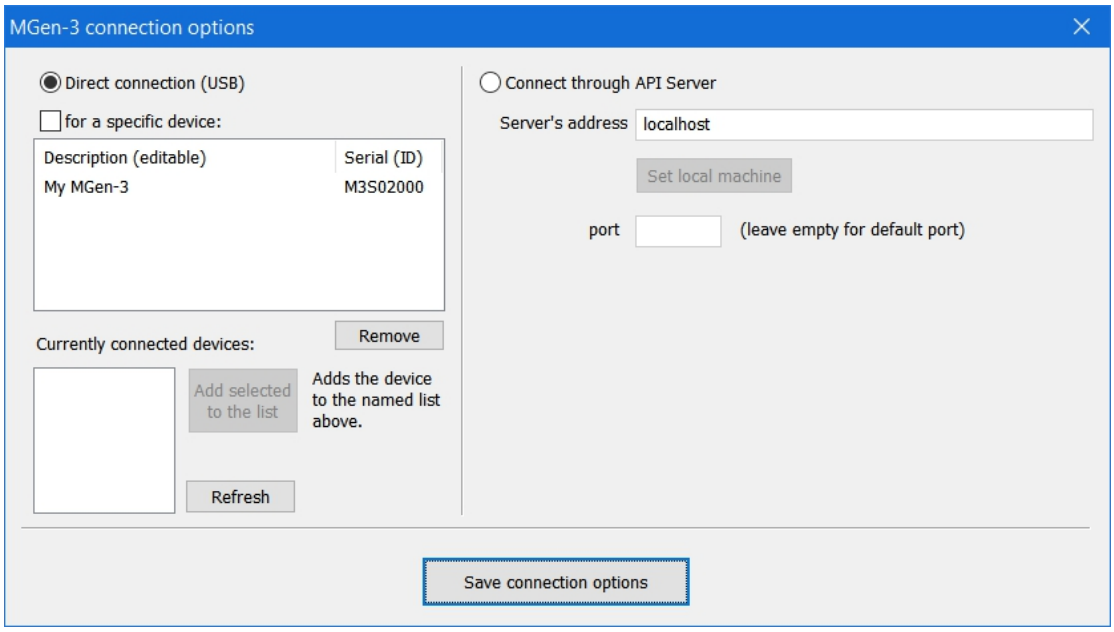
Official applications for Windows system have new versions that support the access of a MGen-3 HC through an API Server. The common configurations of these is described in this section.

After starting such an app., each has a button dedicated to configure the connection mode to the MGen-3 HC, named “Connection options” (or “Conn. opts.”). As an example let’s see this at the “MGen-3 updater v. 1.3” app.’s window:



By clicking on the top left button you get to a common dialog box to set the connection options for this application. Each application will have separate options saved, changing one’s options will not affect the others’ but the dialog box is the same.

The common dialog box looks like this:

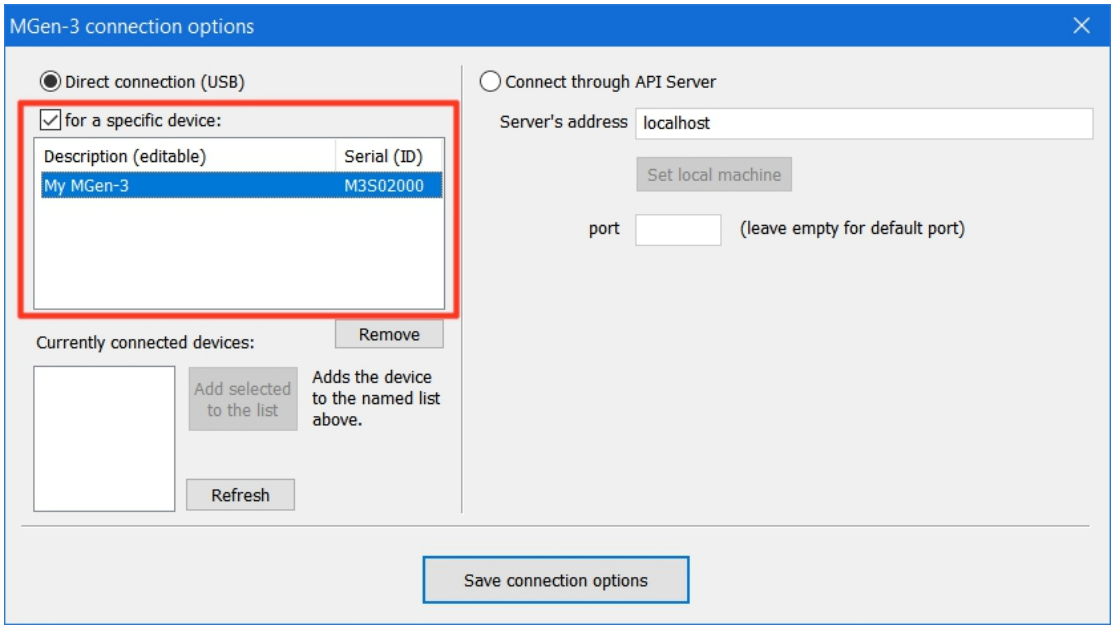


There are two different modes to connect to a MGen-3 HC device: direct or through an API Server. You can choose from these using the two topmost radio buttons “Direct connection (USB)” or “Connect through API Server”. While the left side of the window contains the direct mode’s options only, the right side contains the API Server’s connection data. You can edit whichever side is actually chosen.

You must click on “Save connection options” button to store the new options and close the dialog, otherwise the previous options are unchanged by simply closing the window clicking on “X”.

2.2.1. Direct connection

In direct connection mode you must specify if you want to open a known MGen-3 HC or only the first available connected to the PC through the USB cable. Old applications connected always to the first available HC, in new apps that have these connection options you can specify a known device. To specify a device you must check the “for a specific device” and select one of the devices in list (detailed later):



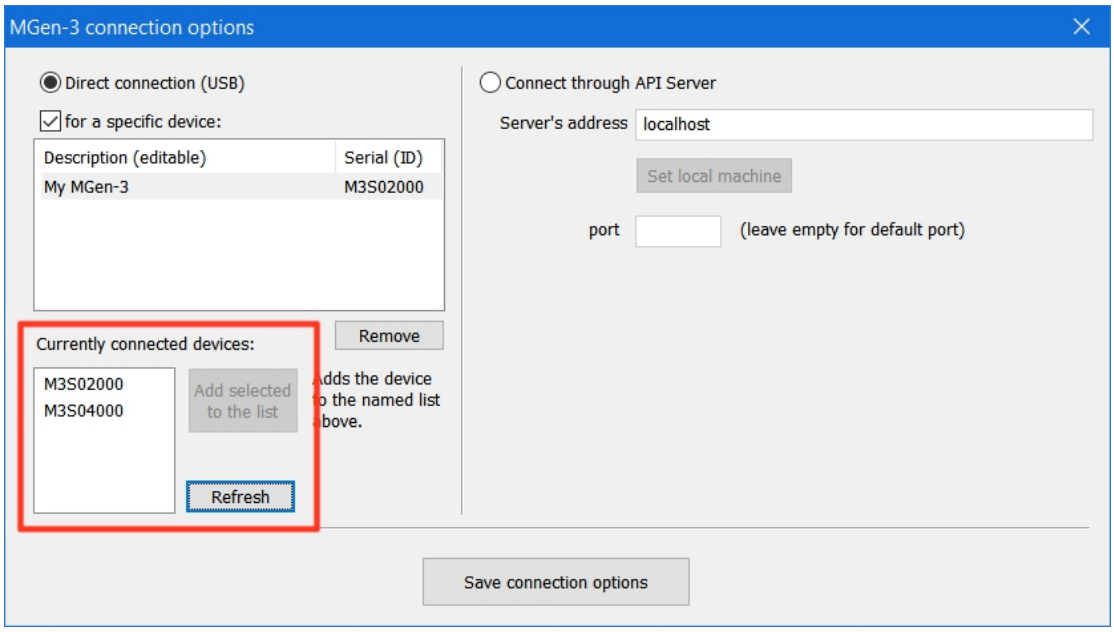
Once you save this state, the application will only try to open the specified MGen-3 HC. The identification is done by its serial ID (FTDI chip’s serial, a different one to the serial number on the stickers). In this example the serial is “M3S02000” but you can name the devices whatever you want: after selecting the device, click again on the “description” field of it and you can change the name/description of it. The different applications will show you this name/description for any purpose instead of the serial ID. IDs are used only by the API when a connection must be established.

The list of devices are common to all the applications having this option dialog. If you change the list or (re)name a device, you’ll get the changes for all the applications’ conn. options.

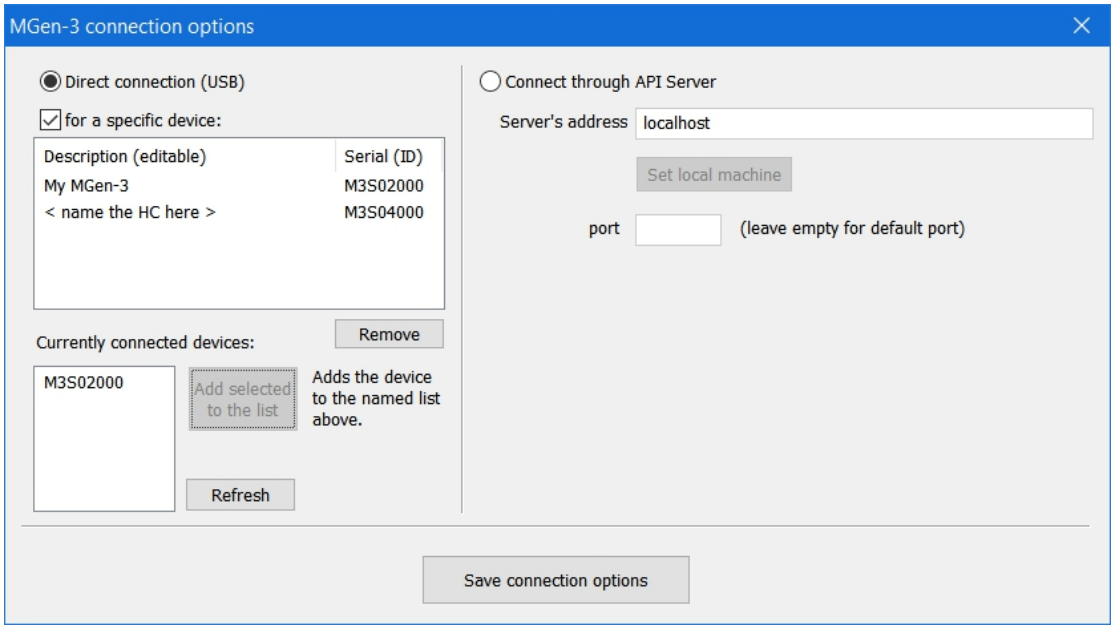
By renaming a device and accepting it with the enter key, it is instantly stored in registry and doesn’t need the “Save connection options” button to be clicked.

To remove an entry form this list, select the device and click on the “Remove” button below it.

To add a new device to the list you need to have the new device plugged in to your PC. (Not required to be turned on.) After this click on the “Refresh” button to list all the available devices connected to the PC:

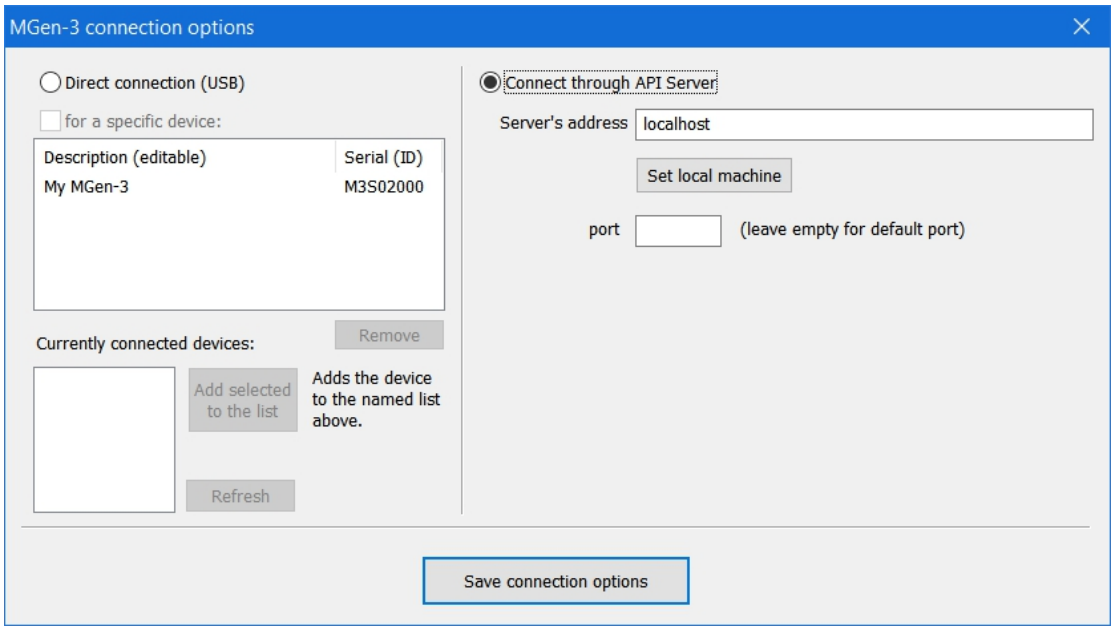


From this bottom list you can select a device by its FTDI chip serial ID and add it to the top list from the button on the right (“Add selected to the list”). The new device is inserted but not yet named. Give a description or name for this HC after.



2.2.2. API Server connection

An MGen-3 device is hidden behind an API Server. You have no option to choose a specific MGen-3 HC by the app in this conn. mode, instead you set this at the application that implements (runs) an API Server. But you must enter the address of the API Server to be able to connect to it:



The address is composed of the IP address (or domain name) and the TCP port number of the server.

If you run the API Server on the same PC as the application whose options you are changing now, you can specify the “localhost” as the IP address. The button “Set local machine” sets the address field to this unique address.

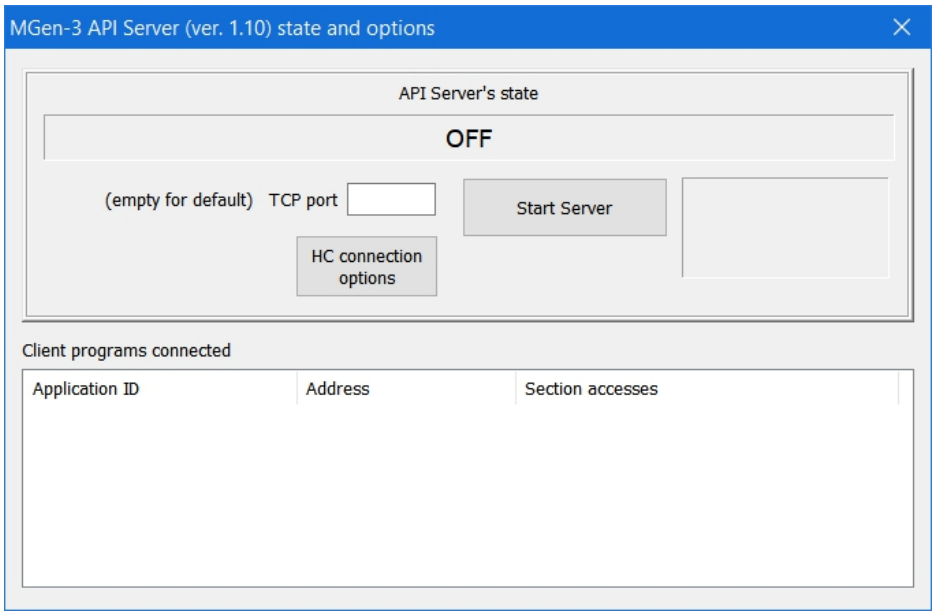
The port number field can be left empty, in this case the default port number of MGen-3 API Server is used, which has been defined as 5937.

Having this connection mode chosen your application will try to connect to the API Server at the given address each time when accessing the HC. This is done through TCP/IP stack and therefore a higher timeout or delay may be observed for any operation, because the network, the API Server and also an active antivirus software can slow down the process. This is mostly valid for connection opening time, particularly for the first time.

2.3. How to set up an API Server

The topmost item on the main window of the PC App leads to the API Server control dialog that we need to set up and make a server running. Click the “API Server” button.

The control dialog box opens:



At the top the API Server’s current state is displayed with a bigger font. The following states are possible:

- **OFF**
  This means that there is no API Server constructed and running in the background.

- **Active**
  If only this is shown, the API Server is constructed and the client APIs of the different applications can connect to it. However there is no HC opened by the server yet and tries to open the connection to the HC is carried out by the time of a client requesting the connection to the server (and so to the HC).

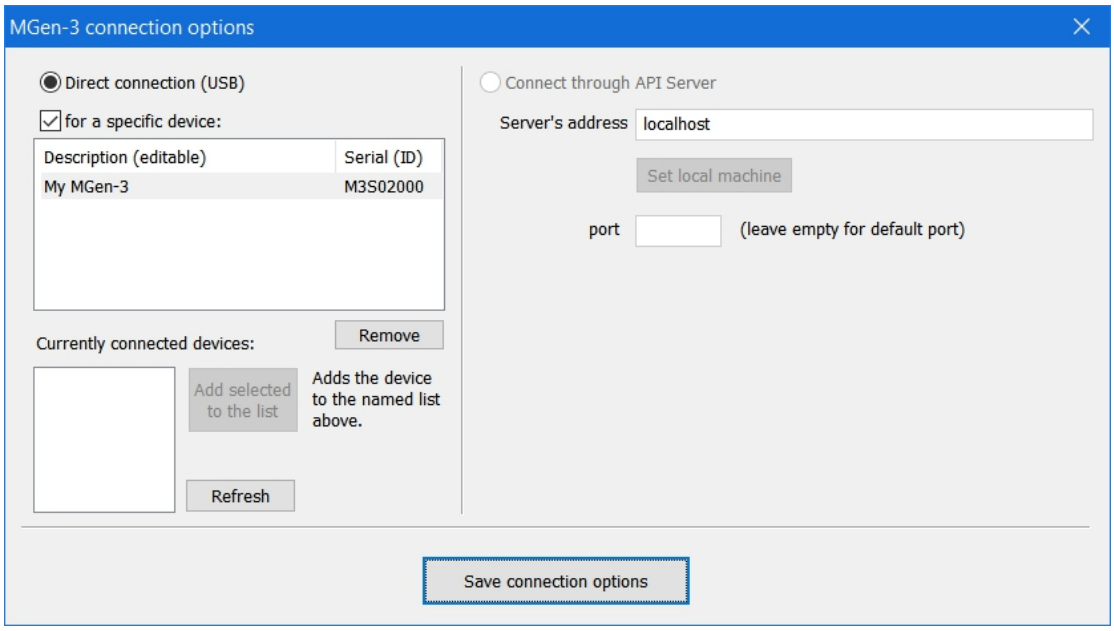
- **Active (open to ‘…’)**
  The API Server has been connected to the shown MGen-3 HC (of what name/description you gave for it). Client applications connecting to the server will have access to the HC opened by the server.

2.3.1. Setting the conn. options for the server

First of all you need to set the connection options of the API Server, just as you must do for the applications: click on the “HC connection options” button. (Only available when the server is not running (**OFF**).)

The option dialog box is the same as described before. The difference is that the direct connection mode is enabled only, as a server can't connect to another server. The direct connection is carried out by the API Server according to these settings.

For this example our options are the following:



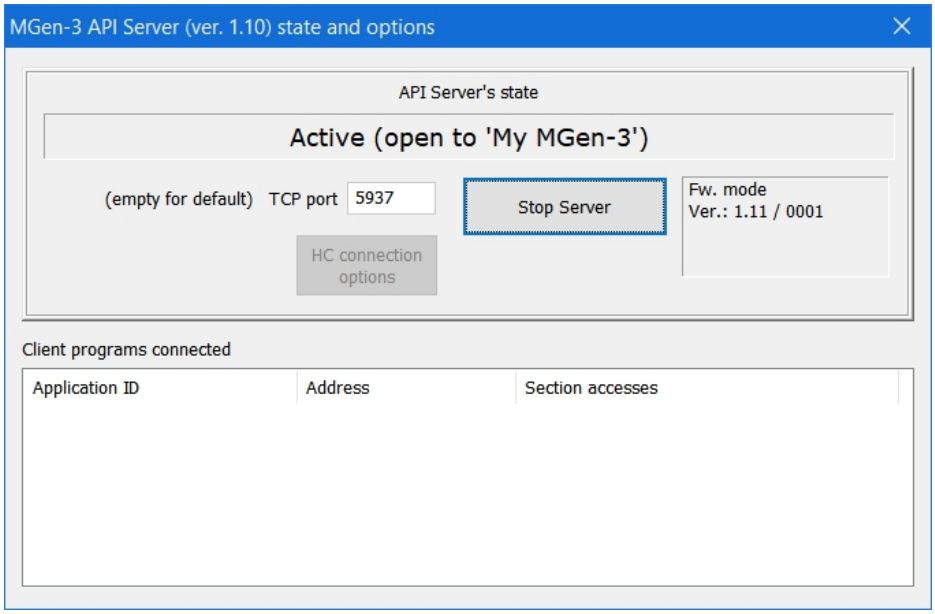
The device named "My MGen-3" (with the FTDI serial of "M3S02000") is selected to be opened by this server.

2.3.2. Start the server

You can optionally define the TCP port number of the API Server. If you set this value to e.g. 1984, the applications connecting to this API Server must use the port number of 1984 too. You can leave this field empty to use the default port number.

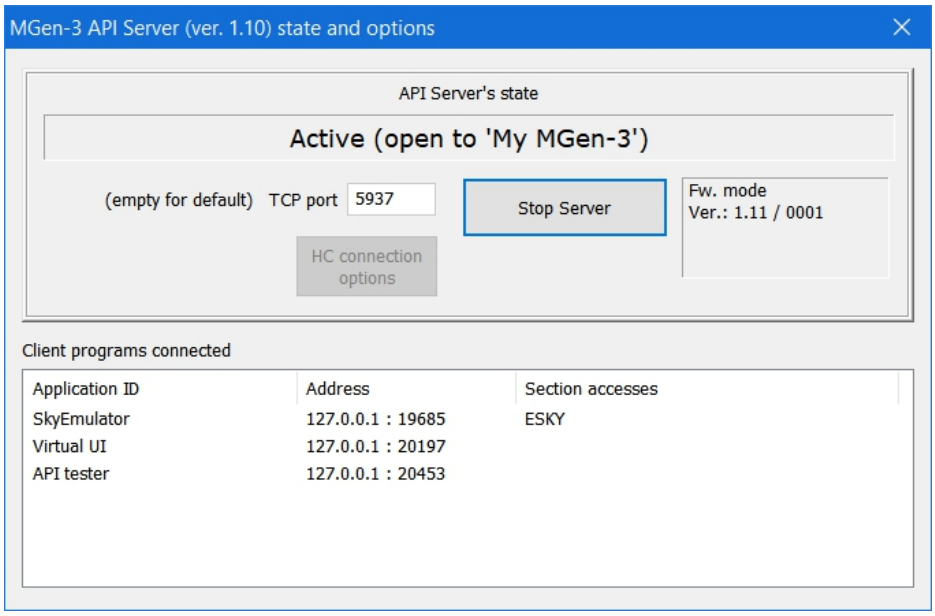
Now ther API Sever is ready to start. You may turn your MGen-3 HC on before starting the API Server, we assume this for the example now, the HC is running a firmware.

By clicking the "Start Server" button the API Server is constructed in the background and as the HC is turned on, the server can open the connection to it right after its creation. Now the **Active (open to '…')** state can be seen and also some information about the connection on the right text box, which mode the HC is in (BOOT or Firmware). In this case the HC is in Firmware mode with the version of 1.11. (The number after is not relevant for you, it's the extra version number encoding miscellaneous info.)



From this point the applications can connect to this API Server to get concurrent access to the HC. The list of the connected clients are shown on the bottom of the window. Here you can monitor what applications have some kind of access to the HC.

For example, after three applications have connected to the server and actively using it you'll see:



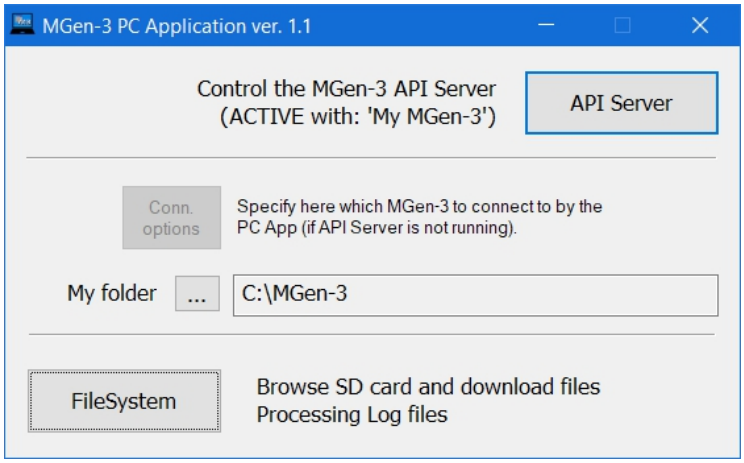
The column "Application ID" shows the name what the client apps. provided. This field can be empty if the app. doesn't specify this info for the server.

All apps. connect from the same PC so their IP address is 127.0.0.1 that is the numbered form of "localhost". They have different TCP port numbers shown (following the IP addr.), this info is not relevant, maybe for expert/debug use.

The "Section accesses" column shows whether an app has exclusive access to some set of functions. This is a must to avoid mess up between the clients' requests. The above example shows that the official SkyEmulator app. has the exclusive access to provide

“external data of the sky frames” (“ESKY”). This section access info is also not relevant for a user but for a developer using the MGen-3 API.

If you close this window, you return to the main PC App window but the API Server will be still running, which is indicated by the altered text next to the button:



2.3.3. Stopping the server

The main button now shows “Stop Server” and by clicking it the API Server is immediately stopped. (The running client apps. will get different error codes that indicates that the HC is not accessible anymore.) All clients are dropped and the connection to the MGen-3 HC is closed. Stopping the server should be done only when there are no active client connections.

2.3.4. Behaviour

Some info to know about the API Server’s behaviour.

- The API Server does not issue commands to the connected HC by itself, it won’t query the current state of the HC so you’ll see the last queried state of the HC on the API Server control dialog. This remains until a client issues a command to the HC, then it is refreshed.
  For example if you turn off the HC manually (using the HC’s buttons) and there’s no client app. connected who is periodically issuing commands, you won’t see on the state display that the HC has been turned off but nothing wrong will happen. At the next time when a client connects to the server and wants to communicate with the HC, the server will notice the change and change its displayed state (“**OFF**” state). The client app. will receive (and should handle) a result value indicating that the device has been powered off since the last command.

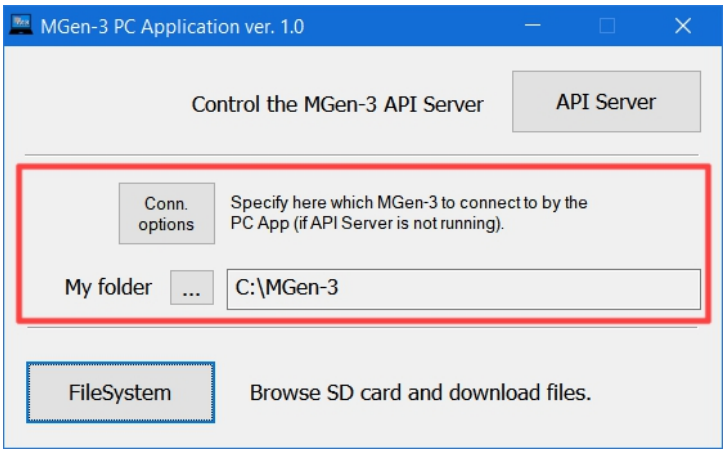
- When there is no HC opened by the API Server, e.g. because the HC was turned off when the server started, the connection to the HC is tried to be established at the next command issued by a client. So the opening the connection to a HC is automatic from the simple “**Active**” state.

Note that clients apps. have the ability to turn-on the HC at anytime through the server if the required HC behind the server was turned off.

- The API Server running in the background is part of the PC App executable so you need to have the PC App running to keep the server alive. If you try to exit PC App and the server is running, you will be asked to ensure exiting because exiting would stop the API Server too.

3. Settings of PC App

Right below the API Server's section you can see the common options of PC App parts.



There is the "connection options" common dialog box available just like for the other official apps. but these options are valid for the sub-programs or parts of the PC App. When a sub-program is started it tries to connect to a MGen-3 HC according to these options.

Important: if an API Server is hosted/run by this PC App, its sub-programs like FileSystem access etc. will connect to this API Server hosted, regardless to the "connection options" set for them. Connection directly to a HC or to another API Server (hosted and run by another app.) is possible only when this PC App you are using does not have a server running.

At the "My folder" part you can select a local folder / directory on your PC that will be the root directory storing or reading files by the sub-programs. It should be set once, the folder must be existing.

Different MGen-3 HC instances will use separate sub-directories so you don't need to change this root folder each time you act with another MGen-3 HC.

4. FileSystem access and Log file processing

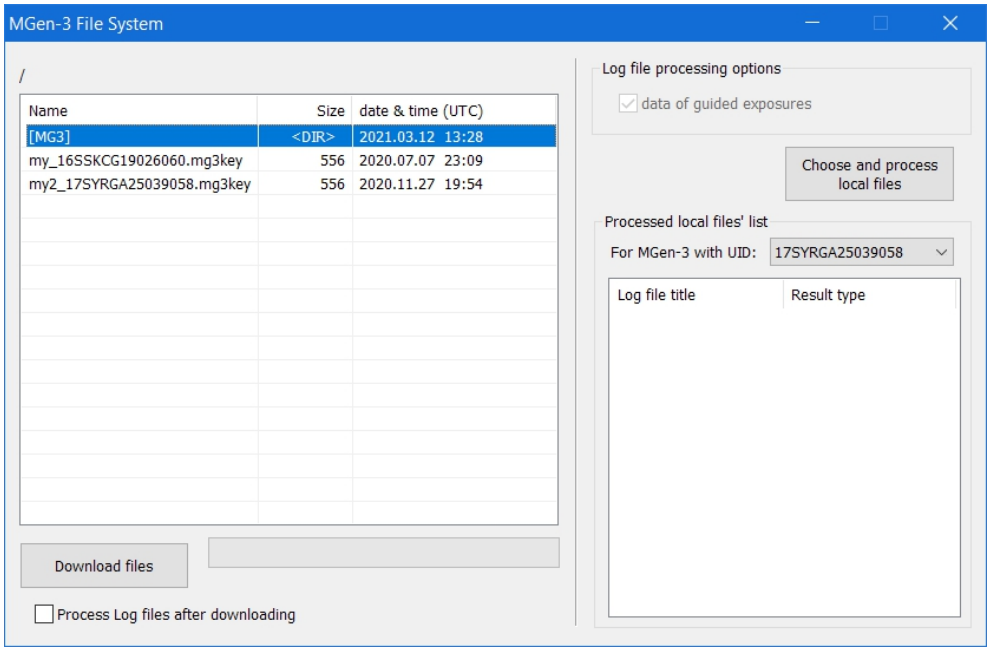
To run this sub-program and have access to the FileSystem click on the “FileSystem” button on the PC App’s main window.

If you are about to browse the FileSystem, these are some prerequisites:

- The HC must be running in Firmware mode and the minimal version is 1.11.
- An SD card must be present in the HC that is in usable state.

If you are only about to process locally stored log files, you should not have any MGen-3 connected to the PC, though the sub-program will try to that takes some time to fail. After the notification is showed that the app. can’t connect to the HC, the sub-program’s windows appears but with the FileSystem browsing parts disabled.

This sub-program’s window looks like the following:



On the left part you can find the FileSystem browser and downloading functions. On the right the log file processor’s controls can be seen.

These two parts are not strongly connected so will be described separately.

4.1. FileSystem part

On the top, right above the list view you can see the current working directory that is listed. As there can be only one file system present on the HC, there’s no drive letter used. The root directory is “/” as shown on this picture. (The separator character used by the HC’s FileSystem is slash instead of backslash used commonly in Windows.)

Below there are the entries listed of the current working directory. Only sub-directories and normal files are listed, hidden files, system files are not listed.

A directory's name is put between square brackets ("[…]") and at the "Size" column "<DIR>" is shown. Directories are always listed first then the files. The directory and file entries are ordered by their name.

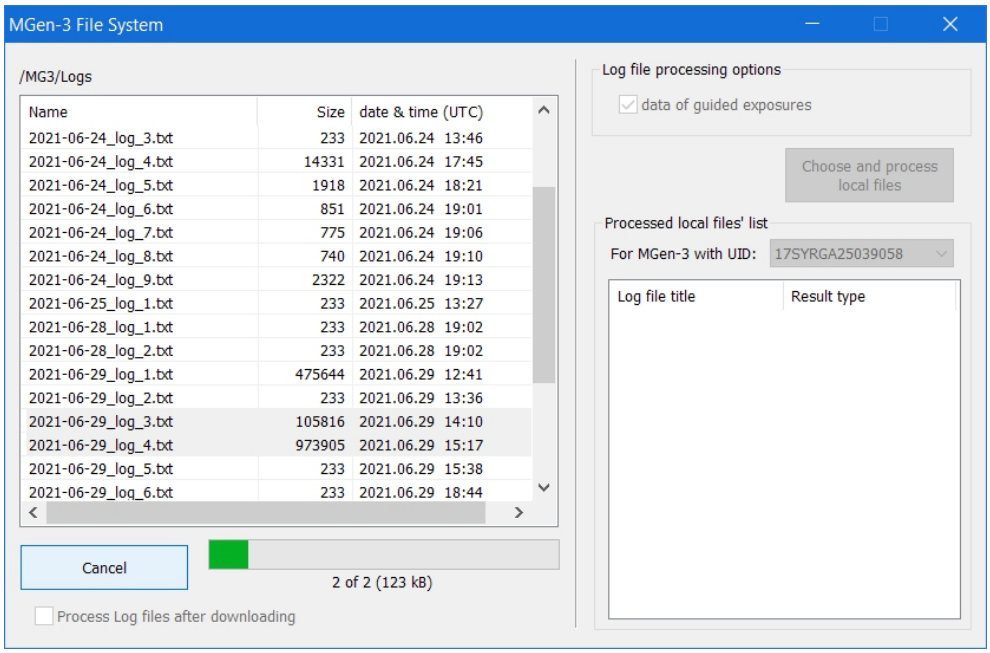
"date & time" column shows the last modification time of the item. Date and time is always UTC for the MGen-3. These time values may differ when you browse the card with PC or another device due to daylight saving compensation. The time value stored onto the card is always what the MGen-3's internal realtime clock showed that time.

You can navigate through the directories and files by pressing up or down arrow keys and Enter. Pressing Enter on a directory item will change the directory to that relative path. (For non-root directories there is always a "[..]" entry that leads up to the parent directory.)

The current working directory is stored by the HC. At the first time (after the Firmware was started) this directory is the root directory. Then after changing it by browsing the files, another start of this sub-program will start up in the directory where it has been left. If the listing takes noticeable amount of time, you may observe a delay before the sub-program window opens.

4.1.1. Downloading files

You can select as many items as you want using the mouse left button along with Shift and Ctrl keys. When done, click on “Download files” button and the selected files will be transferred into the local base directory set. The progress bar will show how much of percentage is already transferred.



To cancel the downloading you can press the button “Cancel”. The file partially downloaded will be deleted on the PC.

Of course you need to have file and directory creation rights to the given base folder to carry out the operation.

Files that are already existing by the same name will be overwritten by the new file data.

Selected empty (zero size) files and directories are ignored from the selection.

The actual place of the files downloaded is constructed as the following:

```
<my folder>\<uID>\SD\<file’s absolute path>
```

where

`<my folder>`	is your base folder set in PC App’s main window.
`<uID>`	is the User ID of the HandController that uniquely identifies the HC instance.
`<file’s absolute path>`	is the absolute path of the file on the SD card.

In our example for the first file

<my folder>	C:\MGen-3
<uID>	17SYRGA25039058
<file's absolute path>	MG3\Logs\2021-06-29_log_3.txt

…and so the actual local place of the files will be:

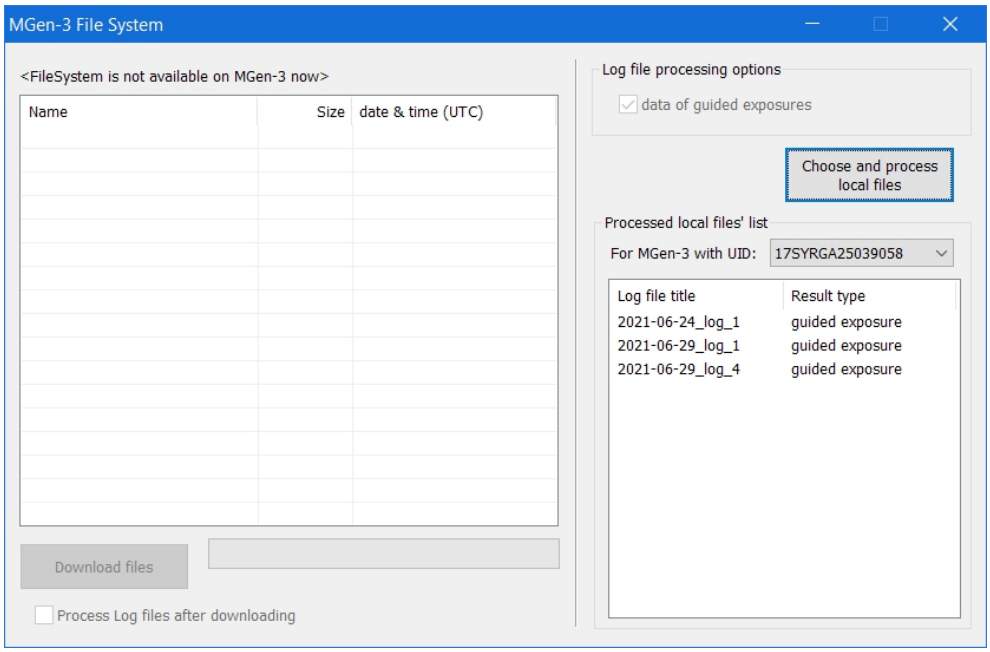
```
C:\MGen-3\17SYRGA25039058\SD\MG3\Logs\2021-06-29_log_3.txt
```

The bandwidth of the USB device port of the MGen-3 is low as it was not designed to handle huge data transfers. The maximal speed is 300k Bytes/s, though the actual speed is yet some lower and might depend on the SD card too. You will be asked if too much time (ca. over 1 minute) is estimated for the transfer before starting it.

If you want to process the selected log files automatically after downloading, you can check the "Process Log files after downloading" checkbox. With this you can spare choosing these files again in your local folders. Note that if you download a file that is not a valid log file, the processing will attempt to read it but will fail.

4.2. Log file processing part

The current version 1.1 of PC App can only extract output for guided exposure data from the log files. Thus this only option is disabled yet at the top. (The FileSystem controls can be seen disabled on this screenshot, because there were no HC connected. We process only local files now.)

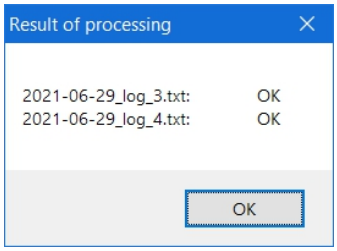


The output format for such data is a set of HTMLs that can be viewed by any browser. You can see there all the data and graphs separately for each exposures taken, when the autoguiding was active. One set of HTMLs are created for each log file. Data will be created for every exposure present in the log file that has guiding info too, even ones that are not "real" or useful (stopped, broken, unintentional etc.). Timestamps of the exposures can be used to identify and pair them with your astro imaging exposures.

4.2.1. Processing log files

By clicking the "Choose an process log files" button you can manually select one or more log files for processing from your local MGen-3 folder, that opens for first. (Go into the uID (HC instance) folder that you need, then the `SD\MG3\Logs` folders and choose from the .txt files there.) Only log files named conventionally will be listed by default. If you rename a log file after downloading, it may not be visible for processing.

After choosing them the processing starts and it may take a short time to proceed. When done, the sumary of the result is shown:



When a processing succeeds, it displays “OK”, otherwise some error text is shown for that log file. The new outputs will be visible at the output files’ list. (Next section of this doc.)

There may be that a log file is processed “OK” but will not create any output. This happens when the log file was valid but it didn’t contain any guided exposure data. Though the output folders are created, they will store no new files.

4.2.2. Output files

At the bottom part there’s a list of the already processed outputs, stored locally. “Log file title” is the title of the log file (name without extension) that was processed creating that output. The type of these entries are “guided exposure” as this is the only option now.

Double clicking one entry will open the main output HTML of that set with your default browser.

The list of the processed files are filtered by the uID folders present locally. By the above dropdown list you can switch between more HCs’ processed outputs.

Log files are meant to be stored locally at a specific folder inside your main MGen-3 folder (see at this section).

Processed outputs’ local folder is created by the input log file’s path, for example:

```
<my folder>          C:\MGen-3

<uID>                17SYRGA25039058

<output path>        Processed\2021-06-29_log_4
```

So the output HTML files could be found at

```
C:\MGen-3\17SYRGA25039058\Processed\2021-06-29_log_4
```

for the log file at

```
C:\MGen-3\17SYRGA25039058\SD\MG3\Logs\2021-06-29_log_4.txt
```

4.3. The “guided exposure” output

This is a set of HTMLs, each HTML shows data of one guided exposure from the log file. For better interpretation here’s some detail about the output’s view.

Don’t forget that all time values are shown and meant in UTC! The date is not displayed for every time value. If a time value wraps around 24 hours, the date/day change won’t be displayed, as one log file is meant to hold data for one night only. Time lengths and graph creation handle the date/day change but is only displayed once at the top bar, where the creation date and time of the source log file is shown.

At the left sidebar you can switch between the guided exposures’ data. The overall length of the exposure and the start time of it is displayed.

After choosing one, the main area has the details of this exposure and guiding, split into sections.

Exposure & guiding info:

Exposure info is straightforward: length is shown with tenth of seconds resolution (as the Firmware). This length is of the actual active exposure signal put out by the HC at its 3.5mm jack connector. Start and stop timestamps are also noted.

Guiding info needs more explanation:

- Guider optics’ focus: the focal length set on the HC when this exposure has been started. This value is used to convert all the drift and length values as arcseconds instead of pixels. Set it properly as precise as possible to have precise and valid arcseconds values.

- AG mode is the autoguiding mode used for RA axis. (“manual” or “auto trim”)

- RA prms. at start/end: P, I and tolerance values used for guiding at the beginning and at the end of the exposure. “Auto trim” mode trims the parameters automatically.

- DEC prms. are the parameters for DEC. These don’t change automatically.

- # of frames: the number of new frames received from the Camera during the active exposure time interval.

- Frames guided: number of frames used for guiding. Lower than the number of new frames only if autoguiding had been stopped during the exposure.

- Frames with star: number of frames that had a guide star present on it. (On the “final”, averaged guiding frame.) This number is less than the number of new frames if the star disappears for any reason during the exposure.

RA guiding graph:

On the left there are time graphs of the observed RA drifts and output correction signal lengths for all camera frames. Each bar shows a value for one camera frame.

The top graph is the RA drift graph, which shows how far the guide star was at that frame (in RA direction) from the autoguiding center position, where the star should be held at. So this graph will spread around zero and the lower the average drifts are, the better the guiding is. (This graph can easily show "only noise" caused by high seeing condition, then the guiding accuracy is mostly limited by the seeing.)
On the right side there are the scale values of the maximal drifts (a full +/- bar) shown, in arcseconds unit.

If a frame's (bar) background is brownish orange, it means that the autoguiding was disabled when that camera frame arrived. No bar is shown for that frame as there were no drifts calculated.
If the background is dark red, it means that there were no guiding star and no drift value available on that frame, but autoguiding was enabled.

The bottom graph is the output autoguiding signals duty of active state after each frame. 100% RA+ value means that the RA+ signal was active until the next frame came. 50% RA- means that the RA- signal was active only for 50% of the frame period time.

On the right there are statistical values calculated for these graphs:

- RA std.dev.: is the average of the all the drift's absolute value, in arcseconds. This expresses the amplitude of "noise" of the position in RA direction of the guiding star, seen by the guider camera.

- [min, max.] stands for the minimum and maximum drifts during the exposure.

- corrections done: counts how many of the frames with guiding star led to non-zero length autoguiding signals put out.

- avg. corr. pulse: shows how long the correction pulses were (either in RA+ or RA- direction) in average, in seconds unit. In bracket, the average duties of these signals are shown.

DEC guiding graph:

Exactly the same as for RA axis but for the DEC direction.

Tracking error graph:

By the known calibration values used for guiding and the output correction signal lengths, the mount's tracking error can be approximated. This shows mostly a part or more periods of the "PE curve" of the mount, when well guided.
If there were no guiding, the mount would probably be that far from the optimal RA position as the sky is rotating.

The scale value in arcseconds is an approximation for this tracking error's peak-to-peak amplitude, observed during this exposure.

Note that if the mount does not react precisely to the RA guiding signals, this approximation may be imprecise or even wrong. Backlash-like behaviour present in the system leads also to invalid tracking error graph.

Guide star FWHM:

A simple graph showing the measured FWHM of the guide star for each frame. If the bars are green, there was a guide star present with the shown size (FWHM). If red with dark red background, a calculated FWHM value is shown but it has been treated not to be a valid star, because it had too low or too high FWHM or was too faint.

Spread images:

The left image is simply the calibration normal vectors for the RA and DEC axes. Horizontally and vertically the sensor's X and Y axes are, while the yellow arrow shows the direction where the guide star moved if the RA guiding signals are active and the blue one for the DEC axis.

The right image is the 2D point spread chart drawn of the guiding drift values. RA drifts are the horizontal positions and DEC drifts are the vertical ones for each frame that had a valid star on it. The background pattern is rotated as the camera sensor's pixel array is and its scale (the width of one block) is shown over the image in arcseconds unit.

For optimal guiding, RA and DEC deviations are nearly the same. (Numerically the statistics should show nearly the same std.dev. values.) In this case this spread chart looks like a round set of random points. (Same width as height.)
When the guiding is far from optimal, the spread would look like a distorted round set of random points, with visible different width or height.

Keep in mind the scale value! Even if the guide star drifts show much worse guiding in RA than in DEC (due to high tracking errors to be compensated), your astro raw images may not show any problem, because the resolution limit of your telescope or the seeing conditions give high FWHM for your stars (like e.g. 2 arcseconds), while the average guiding deviation (std.dev.) is only some tenth of arcseconds. In this case it's not a big difference if the RA std.dev. was say 0.5" and the DEC is only 0.2". The stars on your astro raw images may be still round enough.

4.3.1. If DEC axis was disabled

If the DEC axis was disabled, there will be no DEC graph and statistics shown, as well as no 2D spread chart available. All the other is the same as described above.