MountWizzard4

Release 3.0.1a0

Michael Würtenberger

Feb 08, 2023

CONTENTS

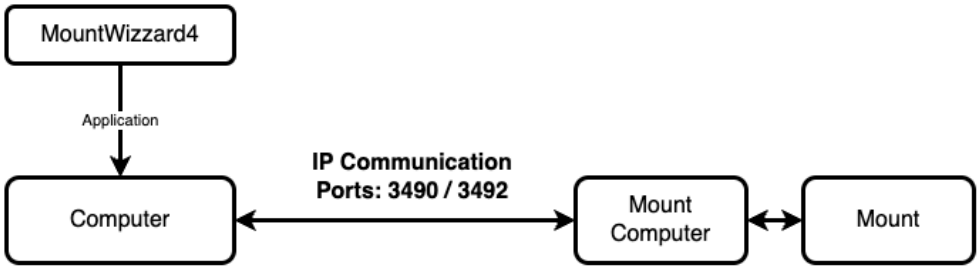
MW4 is a general utility for 10micron users for improving the workflow for astronomy work. It runs on Windows11, Windows10 (Win7 should be fine, but it will be not tested), Mac OSX (beginning from 10.12 to 11.x) including M1 variants if Rosetta is used and Linux (Ubuntu from 16.04 to 20.04). If you have some knowledge around Raspberry Pi's and other SOC, you might be able to install MW4 on a RPi3+, RPi4.

CHAPTER

ONE

BEFORE STARTING

First let us have a look to the basic architecture: MW4 is an application installed on your external computer which is connected to the mount computer via an IP connection. The best choice is to use a wired connection. As the 10micron mounts also support a serial line, please be reminded MW4 does not! Many of the features are handled on the mount computer itself and MW4 does the GUI frontend for the user by using the command protocol provided by 10micron.



The basic idea is that MW4 will try to generate “digital twin” for the mount. All parameter changes for the mount will be sent to it and changes of it’s state are polled to make status visible in MW4. Therefore regular polling of data is needed.

CHAPTER

TWO

OVERVIEW

Beside this documentation there is a youtube channel available with descriptions, previews, explanations:

https://www.youtube.com/channel/UCJD-5qdLEcBTCugltqw1hXA

For full operation MW4 supports several frameworks: INDI / INDIGO, ASCOM, Alpaca and in addition Sequence Generator Pro and N.I.N.A. as camera device.

CHAPTER

THREE

KNOWN LIMITATIONS

MW4 does support python 3.8 - 3.10 right now. The reason for that is the lack of precompiled packages. Some features are limited to windows as they need the original 10micron updater program for execution.

On windows please check if you are working in a 32bit or 64bit environment. You need to choose the ASCOM setup (drivers etc.) and the python install accordingly.

If you are using the 10micron updater features on windows, MW4 remote controls the updater application. Please do not interrupt this automation.

CHAPTER

FOUR

REPORTING ISSUES

To have an eye on your setup here are some topics which you could check:

- Mount connection available and stable. Wifi might have performance problems. Look for right network settings in mount and local setup.
- Good counter check is review settings, status bars, message window if something is going wrong.

To improve quality and usability any feedback is highly welcome! To maintain a good transparency and professional work for my, please respect the following recommendations how to feed back.

Note: Please report issues / bugs here:

https://github.com/mworion/MountWizzard4/issues.

And if you have feature requests discussions or for all other topics of interest there is a good place to start here:

https://github.com/mworion/MountWizzard4/discussions

In case of a bug report please have a good description (maybe a screenshot if it‘s related to GUI) and add the log file(s). Normally you just could drop the log file (or PNG in case of a screen shot) directly to the webpage issues on GitHub. In some cases GitHub does not accept the file format (unfortunately for example FITs files). I this case, please zip them and drop the zipped file. This will work. If you have multiple files, please don‘t zip them to one file! I need them separated and zipped causes more work.

If changes are made due to a feedback, new releases will have a link to the closed issues on GitHub.

4.1 Feature Overview

For being fully operational, MW4 needs either:

- INDI server(s) (see: https://indilib.org) where your devices are connected to.
- INDIGO server(s) (see: http://www.indigo-astronomy.org) where your devices are connected to.
- ASCOM Alpaca remote server (see: https://ascom-standards.org/FAQs/Index.htm) abstracting your ASCOM devices or devices which speak native ASCOM Alpaca if you want to connect over IP with your environment.
- Running versions of Sequence Generator Pro or N.I.N.A. as frontend for camera device.
- For the core devices there is native ASCOM support (Windows platform only). Please be reminded, that ASCOM has 32bit and 64bit driver implementations and MW4 could also be installed in 32bit or 64 bit python environment. They could be not be mixed! 32bit python supports only 32bit drivers and vice versa . Normally this should not be an issue…
- In addition an internet connection is used for some services which might be very helpful.

It is recommended to use mount firmware 3.x or later as some of the functions don't work with older firmware versions. It should not be a problem using older versions. A HW pre2012 might also have some issues. MW4 supports also older firmwares from 2.x onwards, but with limited features and untested.

Combinations of operating systems, architectures and python versions		
	Python 3.7 - 3.9	Python 3.10
WIndows (x86) 32 bit and 64 bit	OK	OK
Mac (x86) ARM M1 with Rosetta emulator	OK	OK
Linux (x86) 64 bit	OK	OK
Linux (aarch64) Ubuntu 64 bit	Only MW4 version 2.x	Not OK (No precompiled packages available)
Linux (armV7) Raspi 32 bit	Only MW4 version 2.x (Only with install script version 2.x)	Not OK (No precompiled packages available)

It is recommended to use mount firmware 2.16 or later as some of the functions don't work with older firmware versions.

Here is an overview of the functionality available in MW4:

- Many settings and features of the mount can be shown and changed.
- Control movement of the mount as well as tracking speeds.
- Coordinates in J2000 as well as in JNow.
- Virtual keypad
- Model building with different model setups and model generating capabilities. Sorting points for effective slew paths or dome situations.
- Model building is done in parallel threads (imaging, plate solving, slewing) to reduce time.
- Show the actual model and alignment error. Give hints on how to improve the raw polar alignment.
- Model optimisation: deleting points, automatic removing point for target RMS etc.
- Manage models stored in the mount (save, load, delete).
- Dome geometry integration (MW4 knows about target flip side and slews dome correctly as well as any geometrical constraints).
- Environment data: MW4 shows data from OpenWeatherMap, ClearOutside, External Sensors like MBox, Stickstation, UniHedronSQR as well as direct linked sensors like MGBox.
- Refraction handling external / internal from the above sources.
- Satellite: searching, displaying, programming, updating tracking.
- Tools: FITS Header renaming, Park positions, etc.
- Remote shutdown of MW4 and Mount via IP commands.
- Measurements and CSV saving for most environment and mount data
- Imaging: control of connected camera / cooler / filter.
- WOL (wake on LAN) boot for mount. MW4 catches MAC address automatically on first manual start.

- Audio signals for different events (end slew, finished modeling, alert, etc.)
- Updater for all MW4 functions.
- Generate / load / save as many profiles as you would like.
- Show alignment stars. Choose and automatically center for polar or orthogonal adjustments.
- Imaging: expose one or N images, auto solve or auto stack these images.
- Imaging: show distortion grid, astrometric calculations (flux, roundness, sharpness)

4.2 Installing

If you on the way of installing MW4 to your windows system, please be aware of the 32bit / 64bit limitations of ASCOM / drivers and python. If you are using 64bit drivers (most likely with the new large scale CMOS cameras), you need to install 64bit python as well as windows does not mix both variants flawless.

Warning: I strongly recommend not using whitespace in filenames or directory paths. Especially in windows handling them is not straight forward and I hardly could do all the tests needed to ensure it's functionality.

4.2.1 Install Python

MW4 is a python3 application based on some python libraries and uses Qt as framework for GUI. Different to past versions of MW there will be no one box solution (MAC bundle, EXE File, etc.) available. As MW4 is python3 and comes with internal update functionality, it uses a standard python3 environment. Ideally it is recommended in a virtualenv.

MW4 is tested on python 3.8 - 3.10. The first step is to install the python3 .8.x package if not already installed. For all platforms there is an installer available. Please follow the descriptions that comes with the installers. To give a short overview here are some quick installation hints for all platforms. The installers for Windows and OSx can be downloaded from python.org.

Warning: Please do not use a newer version of python than 3.9 if you would use MW4 on other platforms than Windows, Mac or x86 Linux. Some libraries bring precompiled binaries with them and they might not be available for a newer python version.

If you already have python 3.8 - 3.10 installed, you can skip this section and go directly to the MW4 installation process. If you have to install python3.8 this has to be done only once for as many MW4 installations you might want.

There is a video on youtube with the install process python: https://youtu.be/xJxpx_SmrVc.

Note: On windows there are some new features which supports comet, earth rotation and asteroids update for the mount. These functions are available from python 3.8.2 on. Earlier python versions have issues. If you would like to upgrade an older python installation, please see the comments below for windows. On other OS there is no need for doing that.

Windows

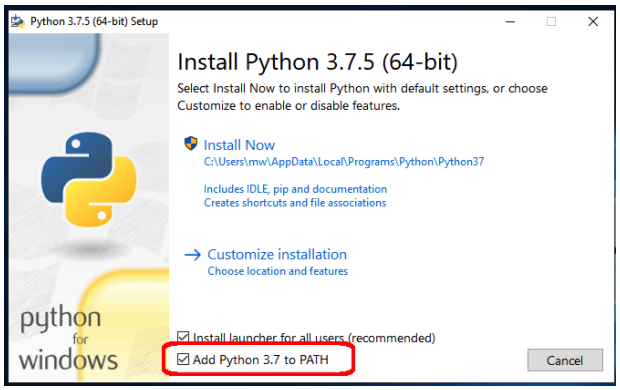
Warning: Windows makes a hard split between 32bit and 64bit versions. If your drivers and setup uses 64bit solutions, please install 64bit python!

Depending on your Windows version please download or directly run the web installer from:

https://www.python.org/downloads/windows/

and follow the installation procedure.

Warning: Please take care that during the installation the checkbox "Add Python Path" is selected and to install for a single user if you want to use the scripts.



Depending on your preference you could install python 3.10 for a single user or for all users. MW4 does not need admin rights to run, so please choose the variant for a single users if you want wo use the installation scripts. They depend on access rights as a normal user and you might run into troubles using different modes!

Mac OSx

Depending on your OSx version please download the installer for 3.10 from:

https://www.python.org/downloads/mac-osx/

and follow the installation procedure. Depending on your preference you could install python3 for a single user or for all users. MW4 does not need admin rights to run, so please feel free to choose the variant you would like to use.

Warning: Using a Mac with Apple silicon need special treatment. There is rather any experience with these setups. Actually MW4 only support Intel architecture so you need to use the Rosetta emulator.

Ubuntu

Referring to Ubuntu 18.04 LTS as it comes with python3.6. This should work, but you could upgrade to python 3.10. This could be done by adding an appropriate repo, which enables this version.

Hint: If you update to a higher python version, please update to python 3.10 if you want in a way, which fits best to your environment. There are many descriptions out, so please search for it in case you don't know exactly.

An example is from: https://linuxize.com/post/how-to-install-python-3-7-on-ubuntu-18-04/

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install python3.10
```

Please check the right version and the availability of virtualenv in your setup. If virtualenv is not present in your setup, please install it prior to run the install scripts with:

```
sudo apt-get install python3-virtualenv
```

Updating python in your existing environment

This is a step which should be done if you are familiar with some pc experience. Hence the steps are not complicated, the setups of you environment might be somehow special and need a adjusted treatment. The following steps explain a standard procedure.

Update python version on your windows computer

Please go to the python website an download the appropriate python version. On windows please check the selection of the 32bit or 64bit correctly. It should be the version you have already chosen.

Start the python installer. If everything went right, it will show an update offer . If so, please chose that and you get the upgrade. If you would like to switch from 32bit to 64bit or vice versa, the updater only shows a new install. In this case please deinstall the old version manually. Than it's like a new python installation, please see above.

Having your python version updated on you computer, you have to update the new version to you work environment(s), too. There are two ways to do that. First you could use the install script provided and install MW in a new work dir. You could copy all you settings (except the 'venv' folder) to the new workdir. Another way is to open a command window, change to your work directory and run the command:

```
python -m venv --upgrade venv
```

This will upgrade your work environment to the python version of your computer (so the updated one)

Note: Before doing any changes or updates, please do a backup of your environment to be safe in case of errors in the update process. This could simply be done by making a copy of your work folder.

4.2.2 Install MW4

When starting with the installation of MW4, python 3.8 - 3.10 should be successful installed. To check, open a terminal (available on all platforms) and run the command

```
python3 --version
virtualenv --version
```

On windows you can't call python3, but you have to run the command

```
python --version
```

In one of the choices you should see the version number of the installed and available packages. For python it should say 3.8.x ... 3.10.x.

Hint: MW4 does not need admin rights to install or run. To avoid problems with accessing directories or file please ensure, that you run install and MW4 itself as normal user!

To install MW4 on your computer, there are some support available for Windows, OSx and Ubuntu to make it a little bit easier to install and run MW4. The scripts are online, and available from Github.

Installing with installer version 3.x:

The install procedure also got improved: You will have only a single compressed python script (startup.pyz) which is valid for all platforms and does all things the different existing scripts stand for. Please download the package and unzip it to get the content. You will find three files:

- startup.pyz -> the script for doing all the work
- mountwizzard4.desktop -> support for ubuntu / linux running the script
- mw4.png -> icon for mountwizzard4.desktop
- mw4.ico -> icon to customize the link in windows for running the script

https://github.com/mworion/MountWizzard4/blob/master/support/3.0/startupPackage.zip

On windows you should be able to start the script just be double click on it, in all other platforms you start it with:

```
python3 startup.pyz
```

On windows please use the command:

```
python startup.pyz
```

Warning: The new script 3.x supports only Windows, Mac and x86 Linux distributions! If you need other support, please use the actual scripts 2.x.

Windows 10/11:

Version 2.x:

https://github.com/mworion/MountWizzard4/blob/master/support/2.0/Windows_Scripts.zip

MacOSx (please use Rosetta for M1 Macs):

Version 2.x:

https://github.com/mworion/MountWizzard4/blob/master/support/2.0/MacOSx_Scripts.zip

Ubuntu:

Version 2.x:

https://github.com/mworion/MountWizzard4/blob/master/support/2.0/Ubuntu_Scripts.zip

UbuntuMate ARM64 (64bit):

Version 2.x:

https://github.com/mworion/MountWizzard4/blob/master/support/2.0/Mate_Scripts.zip

StellarMate ARM64 (64bit):

Version 2.x:

https://github.com/mworion/MountWizzard4/blob/master/support/2.0/StellarMate64_Scripts.zip

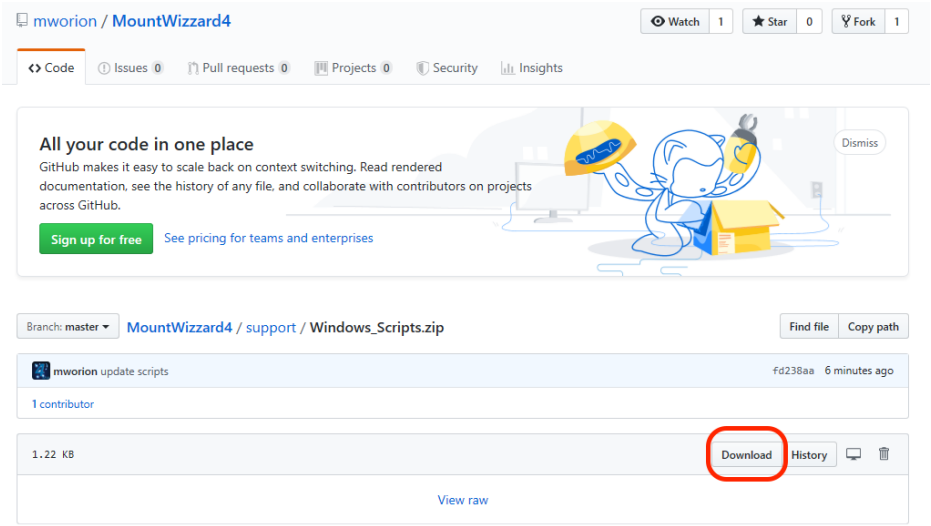
UbuntuMate Astroberry ARM7 (32bit):

Version 2.x:

https://github.com/mworion/MountWizzard4/blob/master/support/2.0/Astroberry_Scripts.zip

Downloading the zip files

Please click the link and press download from the page:



There is a video on youtube with the install process for Mac: https://youtu.be/xJxpx_SmrVc.

Short videos for installation

For a better impression of how MW4 could be installed, there are some special videos showing a installation on different platforms.

- Windows10: https://youtu.be/q9WbiHhW5NU
- Mac OS Catalina: https://youtu.be/bbZ9_yLm1TU
- Ubuntu 18.04: https://youtu.be/kNfLrtJtkq8

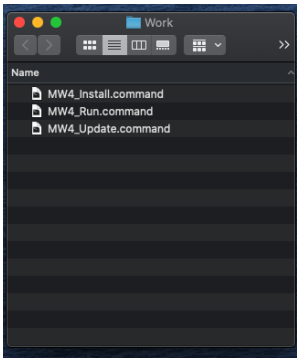
Step 1

Please create a working directory of your choice and location. For MacOSx I would recommend not using a location on the desktop as it might cause troubles with execution right in newer OSx installations. The directory can be renamed later on, it also can also be moved to any other location. Copy the scripts for your platform into this directory.

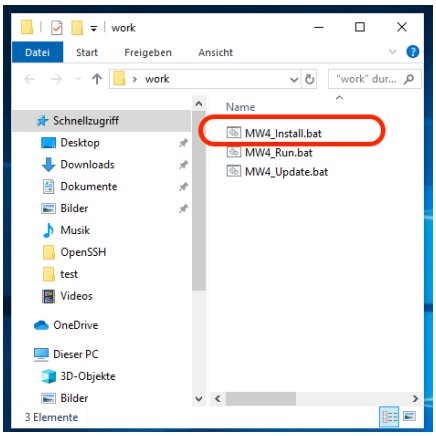
Hint: Over time, there might be some improvements also made for these scripts. So if you had installed MW4 some time ago and will install new setups, it might be helpful to check if some new scripts are available for better handling.

---

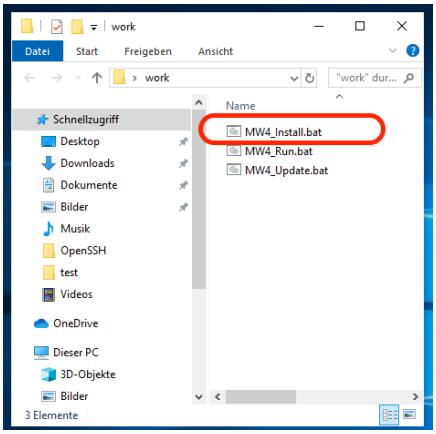
the directory should than for OSx look like:



In Windows10 it looks like:



Warning: Please closely check if your working directory is writable. Otherwise MW4 could not work properly!



Windows10 might as you the first time of execution the following question:

and you could accept that by clicking “addition information” and than execute:

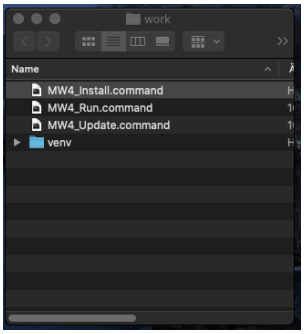
Step 2

Run one of the scripts following script. During installation a terminal window might and shows the progress of installation.

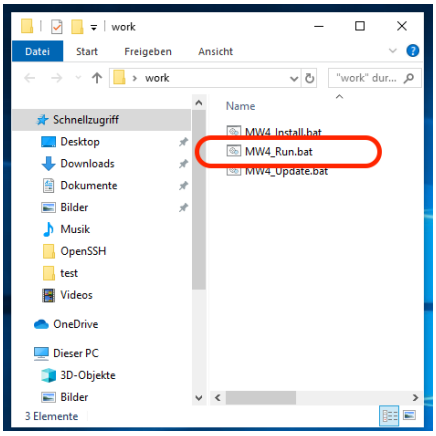
```
MW4_Install.bat          # Windows
MW4_Install.sh           # Ubuntu
MW4_Install.command      # OSx
```

With the script a virtual environment for python is installed in your working dir under the name "venv". After that it installs all necessary libraries and MW4 itself into this virtual environment. So any other installation of python applications is not influenced by MW4 install.

After running the install script the directory should for OSx look like:



In Windows10 it looks like:



In Windows10 for the first time you might be asked again for permission (see above).

Please use for the following step the install marked in red.

MW4 is already installed inside the virtual environment venv in your work dir.

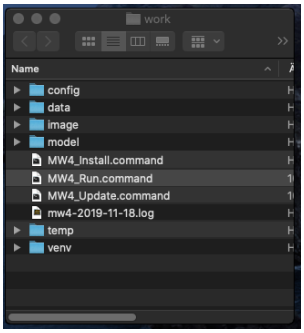
Warning: Please check if an online connection is available on your computer during installation as the libraries and MW4 is installed from online sources.

Step 3

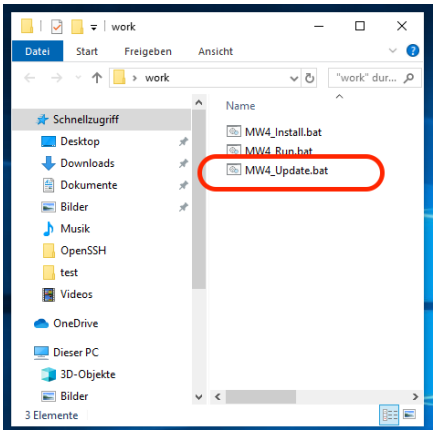
Run one of the scripts

```
MW4_Run.bat          # Windows
MW4_Run.sh           # Ubuntu
MW4_Run.command      # OSx
```

This script will start MW4 for the first time and it will create some subdirectories in your working folder. When starting, a splash screen show the progress of it's initialization. After first start the directory should for OSx look like:

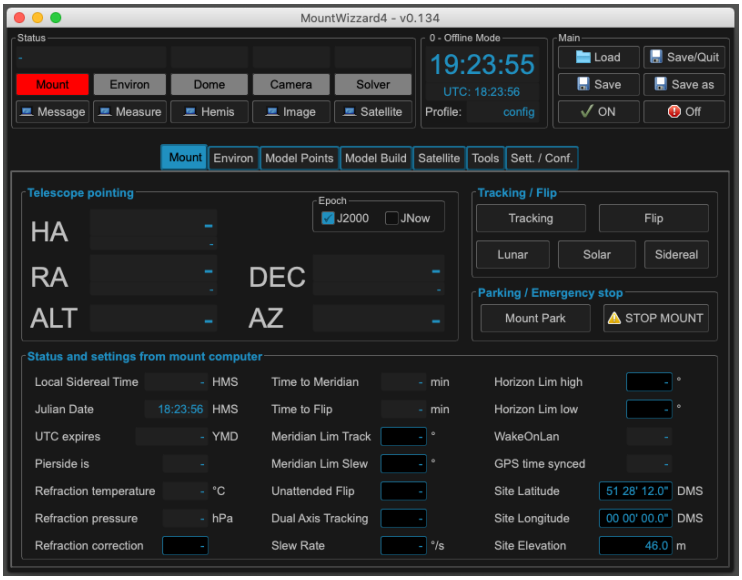


In Windows10 it looks like:



In Windows10 for the first time you might be asked again for permission (see above).

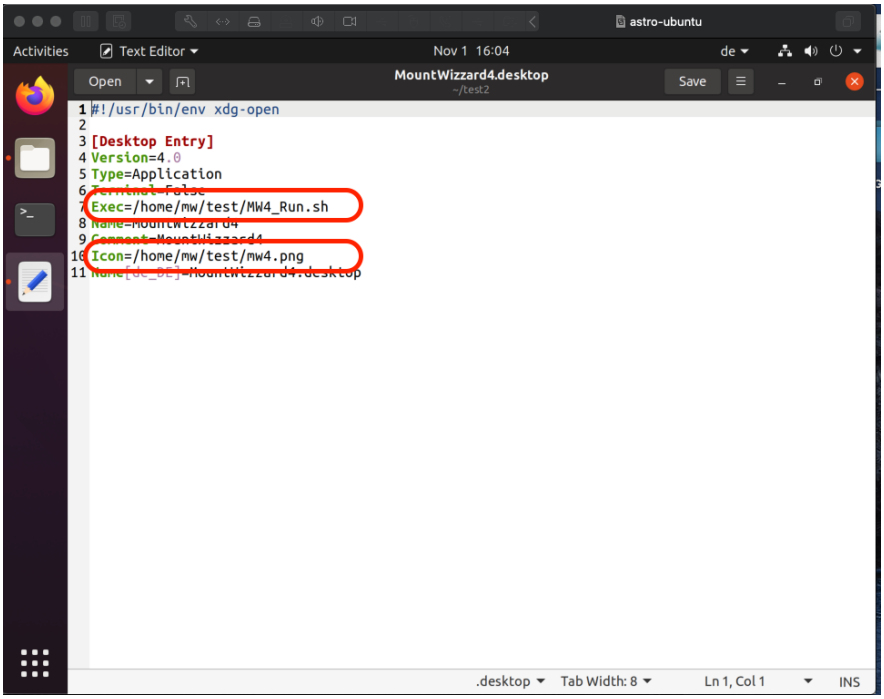
With the first run you will see a log file written and you should have a first window from MW4 open. Please notice that there will be no visible terminal window, but a minimized power shell in the menu. This might take some seconds before MW4 comes up with the splash screen:



If you see the upper window, you succeed and from now on you are able to customize your setup of MW4 and it's features.

Setting up Ubuntu

For Ubuntu the scripts also include an icon file (mw4.png) as well as a desktop description file (MountWizzard4.desktop). In order to use this add-on, please adjust the directories used in this file:



Unfortunately this is broken un Ubuntu 20.04LTS, see (including the workaround):

https://askubuntu.com/questions/1231413/basic-desktop-actions-are-not-available-on-ubuntu-20-04

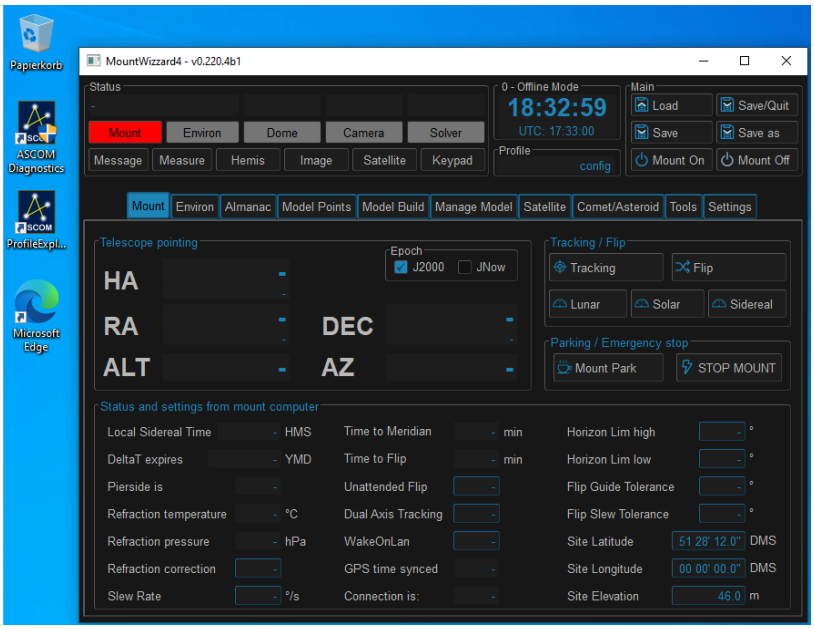
If you install nemo (hint as workaround) as file manager, the desktop icons will work.

DPI scaling on Windows

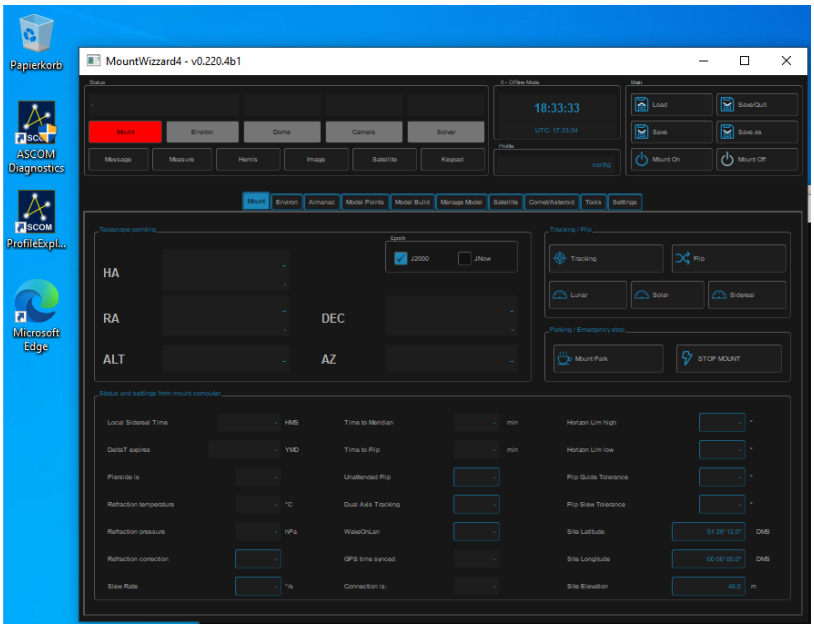
If you are running a windows machine with setting the zoom factor for you display settings different to 100%, you might notice inadequate font sizes etc. Unfortunately this could not be worked around within MW4 itself, but you could change some environment variables to omit this problem. The actual script already contain some setting to keep the resolution to 100% even if you choose to increase this value for other applications. You want to play with these settings to make the appearance correct:

```
SET QT_SCALE_FACTOR=1
SET QT_FONT_DPI=96
```

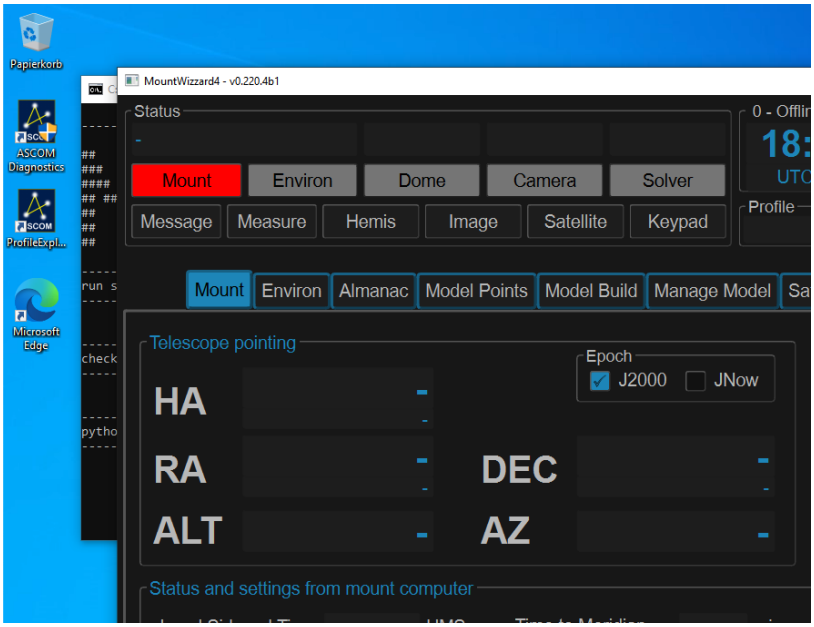
Here some examples of the settings: Normal scaling (scale = 1, dpi = 96)



Small fonts (scale = 1, dpi = 48)



Bigger scale (scale = 1.5, dpi = 96)



If you would like to have MW4 displayed bigger than 100%, please increase the QT_SCALE_FACTOR to the value desired. A value of 1 means 100%, so 2 means 200%. You will experience to set the font adequately.

DPI scaling on Ubuntu

This is quite similar to windows. You have to set the environment variables QT_SCALE_FACTOR and QT_FONT_DPI accordingly. They are already part of the MW4_Run.sh scripts.

Installation on Apple Silicon

For software that is not yet updated, Apple has built in translation software called Rosetta 2. Rosetta 2 will interpret traditional Intel-based code and make it look like ARM-based code. And it does this pretty well. Generally speaking as a user it is very difficult to distinguish between apps that have 'native M1 support' to traditional Intel-based apps.

But for any apps that are run from the command-line in Terminal, this standard Rosetta 2 translation does not happen. Within Astrophotography it is not uncommon to have apps that run from the command-line. Please hav a look to: https://www.astroworldcreations.com/blog/apple-silicon-and-legacy-command-line-software

Update manually

If you plan to upgrade MW4 to the newest release, MW4 has it's own internal updater and using the script is not necessary. In some circumstances this might be necessary. In these cases you could use on of the

```
MW4_Update.bat          # Windows
MW4_Update.sh           # Ubuntu
MW4_Update.command      # OSx
```

scripts. The command script updates to the latest release.

Note: You only could update to official releases. Beta's are not supported.

4.2.3 Install on Astroberry RaspberryPi

I strongly recommend using the actual astroberry server https://www.astroberry.io if you would like to use KSTars/EKOS and MountWizzard4 on a Raspi (3 or 4).

Under https://github.com/mworion/MountWizzard4/tree/master/support you will find the necessary scripts to install MW4 directly under astroberry. Please download the corresponding ZIP archive (Astroberry_Scripts.zip), make a working directory on astroberry server, extract the archive there and run the install script.

For starting MW4, please start the run script. Basically that's all to do.

Hint: Actually astroberry is supported from MW4 version 2.1.3 on. It has some limitations in features: no support of 3D mount simulator.

Warning: The installation for is dedicated for a 32bit system on Raspi4. If you plan to use a 64 bis system, please look to StellarMate64

4.2.4 Install on StellarMate1.7++ (64bit)

Another way of using MW4 on Raspi4 64 Bit is to download bullseye 64bit image or an Stellarmate1.7++ image. On the support page you will find the corresponding install and run script as ZIP archive. Please download them in your work folder. The scripts will use precompiled wheels for aarch64 as much as possible to improved the installation speed e.g. on your RPi4.

```
./MW4_Install.sh
```

After a short while MW4 is installed and should be ready to run like in ubuntu installation.

Note: The install scripts only support python 3.8-3.9 versions"

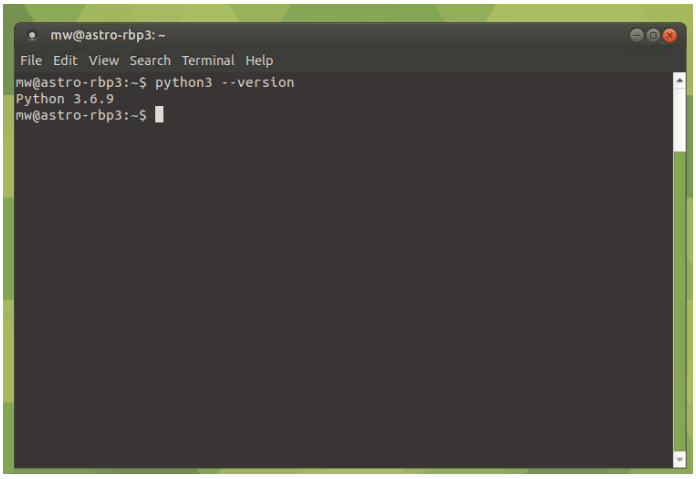
4.2.5 Install on RaspberryPi 3

Hint: The simplest raspi installation for rpi3 works with astroberry

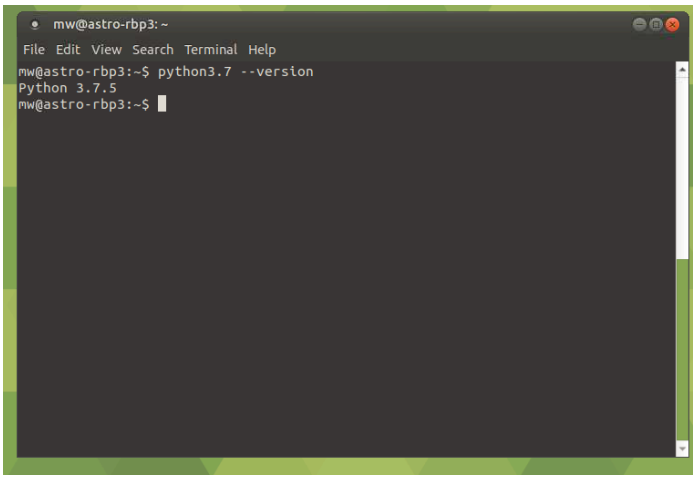
---

Installing Python on RPi3

To get MW4 installed on RPi3 you will follow the instructions of Robert Lancaste (many thanks to him fore this work!) on https://github.com/rlancaste/AstroPi3 with installing AstroPi3 scripts. The installation procedure I describe is based on Raspbian Buster with desktop. should give you the following result:



In addition you have to take care, that python3.8 is installed. The actual Ubuntu mate 18.04.2 distribution comes with python 3.6, so we need to update this. Please follow the description: *Ubuntu* (page 13). After that you should get an python3.8 or newer available on your system:



If everything went fine, we can proceed to the next step.

Installing PyQt5 on RPi3

As on arm the installation of PyQt5 could not be done through pip, the actual tested path is to install Qt directly via apt-get on your RBP3. As result, you cannot install MW4 easily in a virtual environment as apt-get will install all libraries in a system path.

As there were no compiled binaries for actual Qt version available, you have to compile it yourself.

```
sudo apt-get update
sudo apt-get install python3.8-dev
sudo apt-get install qt5-default
sudo apt-get install sip-dev

cd /usr/src
sudo wget https://www.riverbankcomputing.com/static/Downloads/sip/4.19.19/sip-4.19.19.
↪tar.gz
sudo wget https://www.riverbankcomputing.com/static/Downloads/PyQt5/5.13.2/PyQt5-5.13.
↪2.tar.gz

sudo tar xzf sip-4.19.19.tar.gz
sudo tar xzf PyQt5-5.13.2.tar.gz
```

(continues on next page)

(continued from previous page)

```
cd sip-4.19.19
sudo python3.7 configure.py --sip-module PyQt5.sip
sudo make -j4
sudo make install

cd PyQt5_gpl-5.13.2
sudo python3.7 configure.py
sudo make -j4
sudo make install
```

There are in different packages to be downloaded and installed. They build on each other, so keep the order of compiling and install. This procedure take about 2 hours or more, depending on the system.

Warning: So far PyQtWebEngine does not build on RPi3! So I removed for the build from 0 .138 on the capabilities, who need the PyQtWebEngine package. This is basically the Keypad. So you will have limited features!

So before you could actually run MW4 you need to install some mor libraries:

```
sudo apt-get install libgfortran5
sudo apt-get install libjpeg-dev zlib1g-dev
python3.7 -m pip install -U Pillow
```

Once you are set, make a work directory, cd to this directory and install MW4 by

```
python3.8 -m pip install mountwizzard4
```

and run MW4 with the command

```
python3.8 ~/.local/lib/python3.8/site-packages/mw4/loader.py
```

If everything went fine, you should see MW4 on RPi3:

4.2.6 Install on RaspberryPi 4

Hint: The simplest raspi installation for rpi3 works with astroberry

We are installing MW4 on an ubuntu 20.04.1LTS 64Bit system. In relation to the RPi3 it seems to be much simpler to do. Nevertheless some of the big packages will be compiled on your system during installation, which means this will take some time (hours). There is the opportunity to use precompiled packages out of the install scripts provided.

Another big step forward is that you could use now a virtual environment for installing MW4.

Installing Python on RPi4

To get MW4 installed on RPi4 you will follow the instructions of Dustin Casto:

https://homenetworkguy.com/how-to/install-ubuntu-mate-20-04-lts-on-raspberry-pi-4/

to get Ubuntu Mate 20.04.1 LTS on your RPi4.

Hint: Some users experience problems with KStars/EKOS on original ubuntu-mate desktop. So the recommendation is to use a KDE bases desktop like kubuntu. The easiest way to install a desktop on top of the server installation is using: https://github.com/wimpysworld/desktopify

After you have finished the setup and got the desktop up and running, the command

```
python3 --version
```

should give you the following result 3.8.5: Please take care, that a python version 3.8.5 or later is installed.

The actual Ubuntu mate 20.04.1LTS distribution comes with python 3.8.5, so everything should be OK. Next we have to do is to install a virtual environment capability, the packet manager pip and the development headers for python to be able to compile necessary packages:

```
sudo apt-get install python3.8-venv
sudo apt-get install python3-pip
sudo apt-get install qt5-default
```

Note: You need to have both packages installed as otherwise the install script or later does MW4 not run.

Using the precompiled wheels

Please choose the script fitting to you ubuntu version (18.04.x or 20.04.x) The scripts will use precompiled wheels for aarch64 as much as possible to improved the installation speed e.g. on your RPi4.

```
./MW4_Install_aarch64_18_04.sh

or

./MW4_Install_aarch64_20_04.sh
```

After a short while MW4 is installed and should be ready to run like in ubuntu installation.

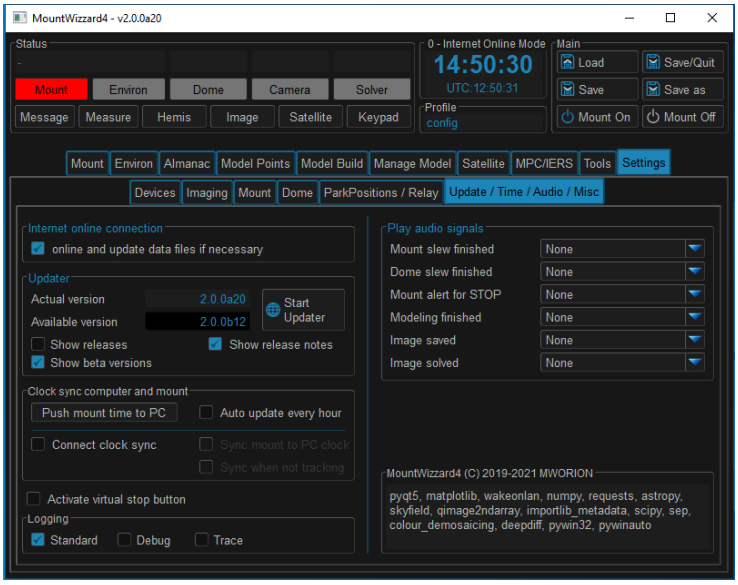
Note: The install scripts only support python 3.8-3.9 versions"

4.2.7 Using the internal updater

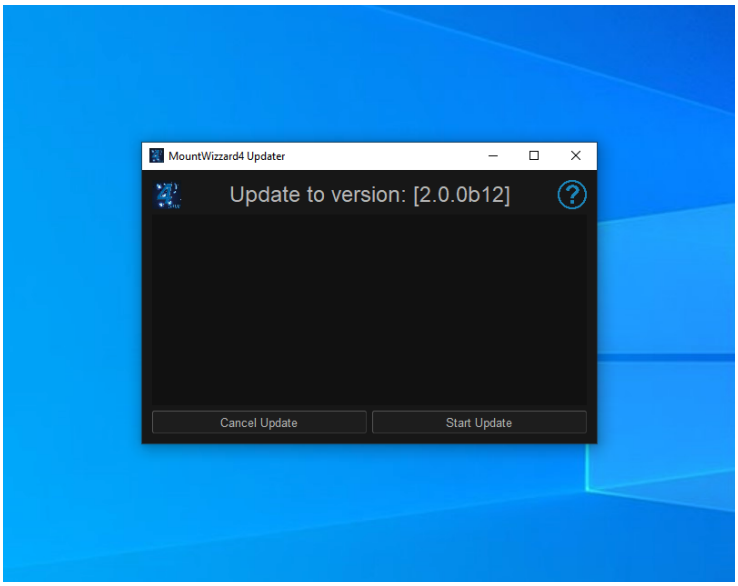
From version 2 on MW4 brings an improved updater concept. Basically you choose to update within MW4 ans select the version of choice. After that a check if version exists will be processed and if successful, MW4 will be closed and the updater program will be started.

Hint: See also on youtube: https://youtu.be/TxxnMizbU1g

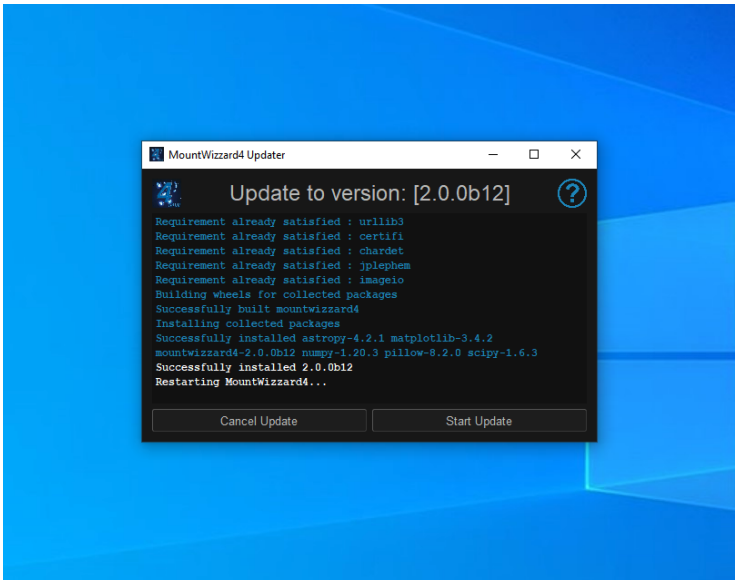
Note: MW4 does not store your actual settings. If you would like to do so, please **save** your configuration before updating.



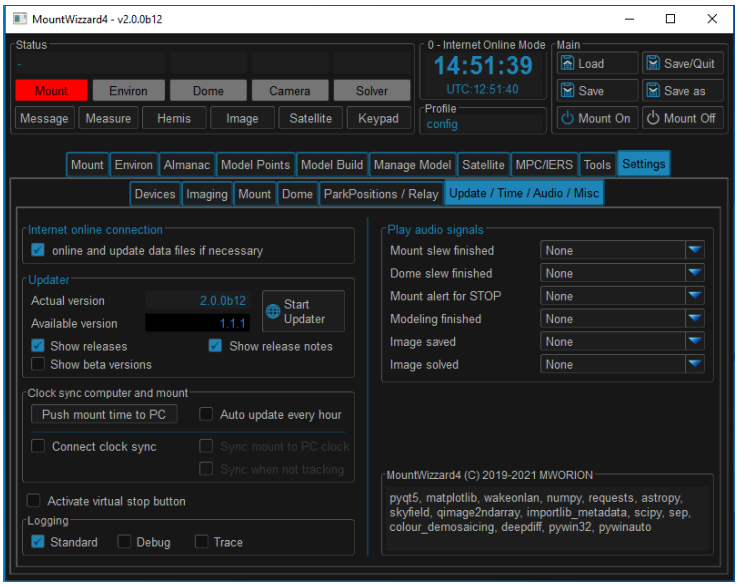
Once the updater program has started you see which version will be installed. Now you could start the update or just cancel it.



If started, the updater will download and install all necessary libraries.



The updater finishes automatically and reloads MW4. This will happen in both cases **Cancel Update** and **Start Update**. If you close the updater window directly, MW4 will be not reloaded.



MW4 new version starts.

4.2.8 Install Plate Solvers

Supported platesolvers are astrometry.net, ASTAP and Watney. All solvers should be installed locally. There is no support for astrometry.net online. If you install a plate solver, please be reminded that you have to install their index files as well. Unfortunately all are using different index files and methods, also depending on your optical setup. MW4 helps you in finding the necessary index files for your setup:

Based on optical and Sensor data. MW4 will make a prognosis, which index selection might be good. You also have direct internet links to the sources.

Astrometry.net

Astrometry.net is useful on Linux and Mac installations. On Windows there is no good setup possible. There are many solutions available (e.g. ANSRV), but these could not be used through MW4. You will find astrometry.net here:

http://astrometry.net

ASTAP

ASTAP is an application available for all platforms from Han. Great software! It is available as application with GUI and as pure command line interface solution (CLI). If you don't use ASTAP for other things as well, I would recommend installing the CLI version locally in a folder of your choice. You find all information on:

https://www.hnsky.org/astap.htm

Watney

Watney is a new solver from Jusas, based in the algorithm Han published for ASTAP. It is available as API and command line interface (CLI) version for all platforms. For the use with MW4, please install the CLI version in a folder of your choice. You will find all information on:

https://github.com/Jusas/WatneyAstrometry

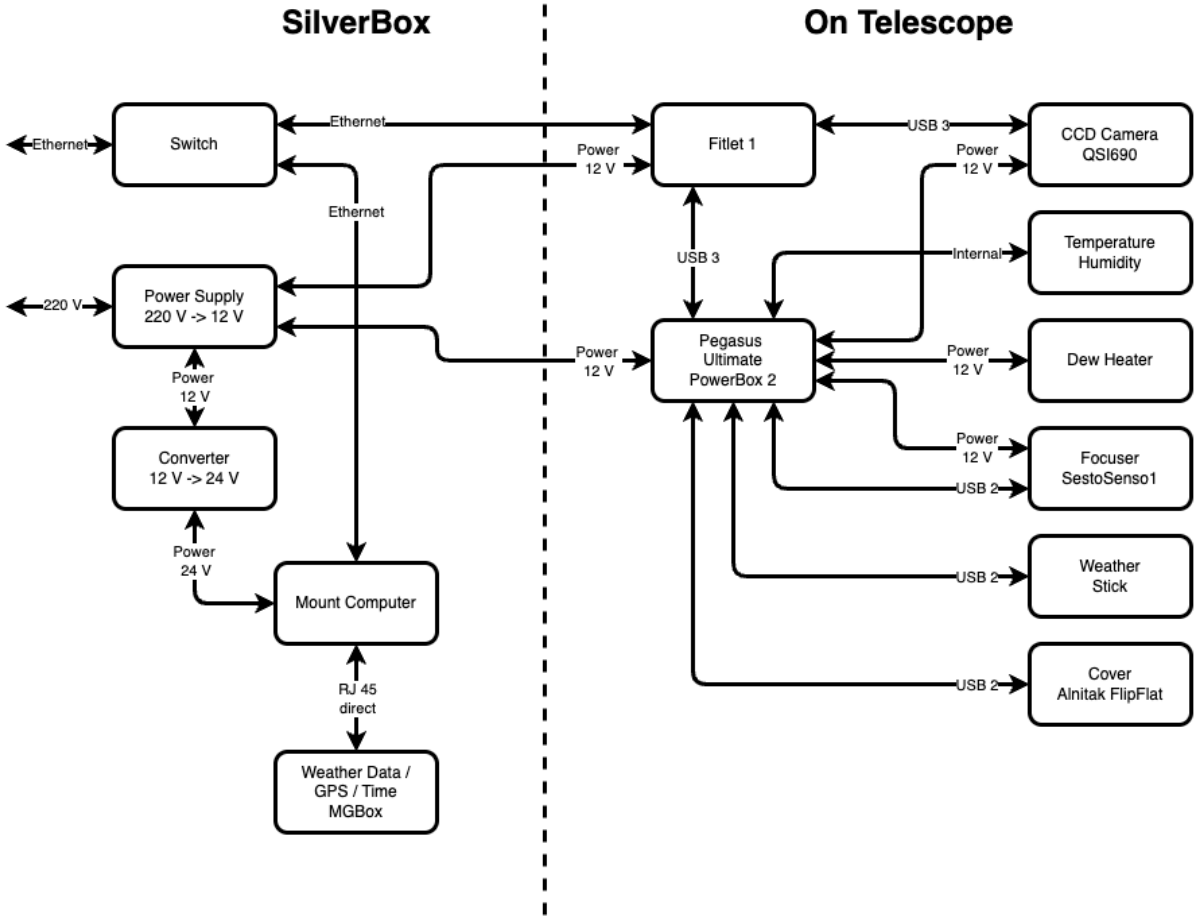
and

https://watney-astrometry.net

4.2.9 Example setup for rig

To give a realistic example I will explain my own setup. I build a portable silver box, which is the housing for the DC / DC converters, the power supply, the ethernet switch, the mount computer and all the switches and connectors needed for attaching the silver box to the mount.

All other components are located directly on the telescope, so the is only an ethernet connection and two 12 V power supply wires to the telescope.



4.3 Configuring

4.3.1 Mount

4.3.2 Dome

4.4 Using functions

4.4.1 Use Almanac

4.4.2 Use Environment

4.4.3 Generate Build Points

4.4.4 Update IERS data

4.4.5 Use Minor Planet data

4.4.6 Run Model Task

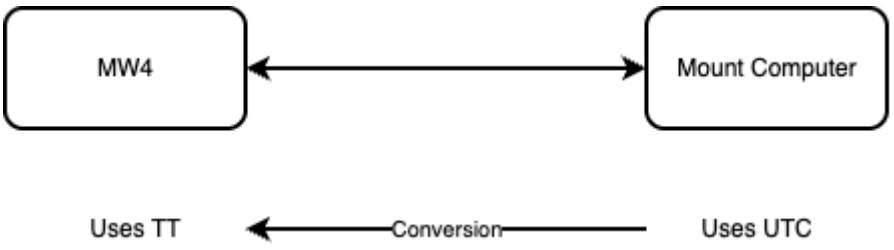
4.4.7 Use Satellite

4.5 Architectural topics and math

Within these pages I would explain how and why I made the architecture decisions for linking it with the 10micron mount computer. This might help for setting up or just explain the behavior you experience when using MW4. I do this also as my development documentation. There might be some faults and error in it. If you find one, please let me know. I would like to get MW4 from it's technical base as clean as possible.

4.5.1 Handling time

One basic definition is that MW4 will use at any time the clock of the mount computer. Therefore MW4 polls julian date, difference utc - ut1, time sidereal. This allows full sync for any calculation to be made. No time from computer to mount is necessary, but could be done at any time (except during model build run). The mount mostly use the julian date representation except for model build where a local sidereal time (LST) is used. In this case MW4 just stores the value and feed it back when the model is programmed. That's the reason why you should not change time during model run.



One important difference between MW4 and Mount exists. As I use skyfield as on of the frameworks with it's units for Angle, Coords, Time etc. I have to take the time definition of skyfield into account. Skyfield chooses TT (Terrestrial Time) as it's basic concept, whereas the mount uses UTC (Coordinated Universal Time) as reference. TT is a modern astronomical time standard defined by the International Astronomical Union. TT is distinct from the time scale often used as a basis for civil purposes, UTC. TT is indirectly the basis of UTC, via International Atomic Time (TAI).

4.5.2 Precision of internal calculations

MW4 is using for all calculations the skyfield (https://rhodesmill.org/skyfield/) from Brandon Rhodes. As for the new command set offered with 10microns FW3.x it needs to calculate the alt/az coordinates for a satellite track each second for the entire track. As you would like to follow the as precise as possible I made some comparisons between the internal calculations done in 10micron mount and the results provided by skyfield.

In skyfield there is a chapter about satellite calculations and precision: https://rhodesmill.org/skyfield/earth-satellites.html#avoid-calling-the-observe-method Despite the fact that the observe method is expensive the difference in calulation time for a 900 step track is on my computer 120ms (using more precise observe method) to 7ms (using the less precise difference).

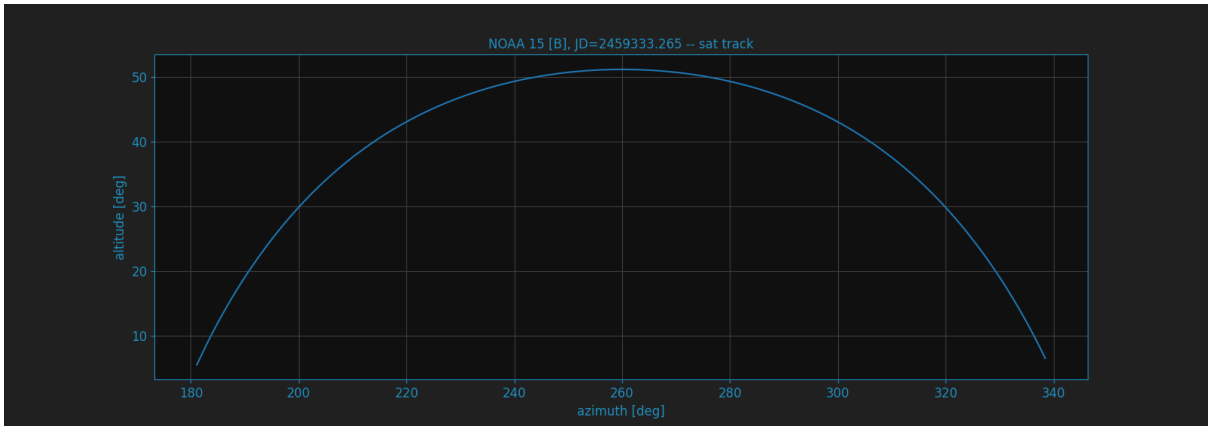
Brandon writes about it:

> While satellite positions are only accurate to about a kilometer anyway, accounting for light travel time only affected the position in this case by less than an additional tenth of a kilometer. This difference is not meaningful when compared to the uncertainty that is inherent in satellite positions to begin with, so you should neglect it and simply subtract GCRS-centered vectors instead as detailed above.

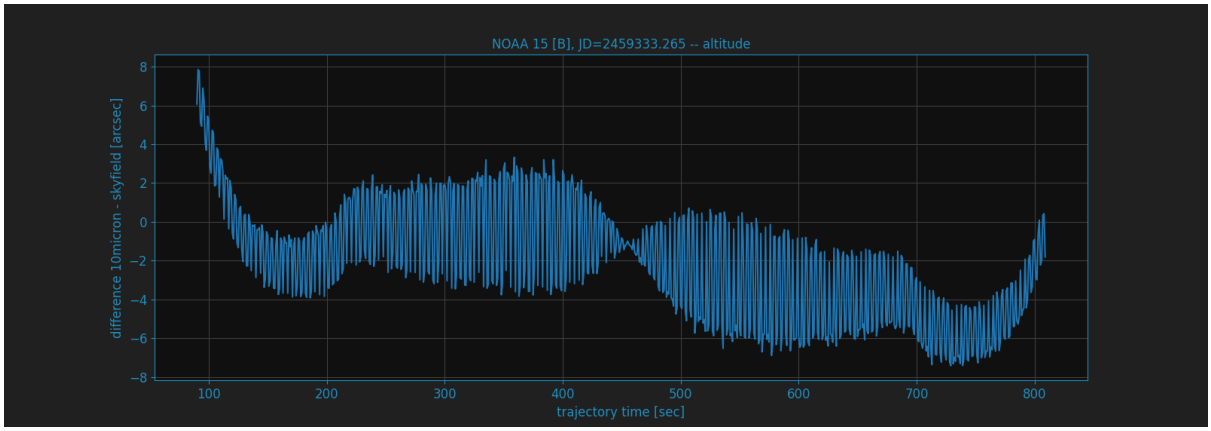
Here the charts for NOAA 15 [B] at julian date JD=2459333.26498 for the transit happening. The used TLE data was:

```
NOAA 15 [B]
1 25338U 98030A   21104.44658620  .00000027  00000-0  29723-4 0  9990
2 25338  98.6888 133.5239 0011555 106.3612 253.8839 14.26021970192127
```

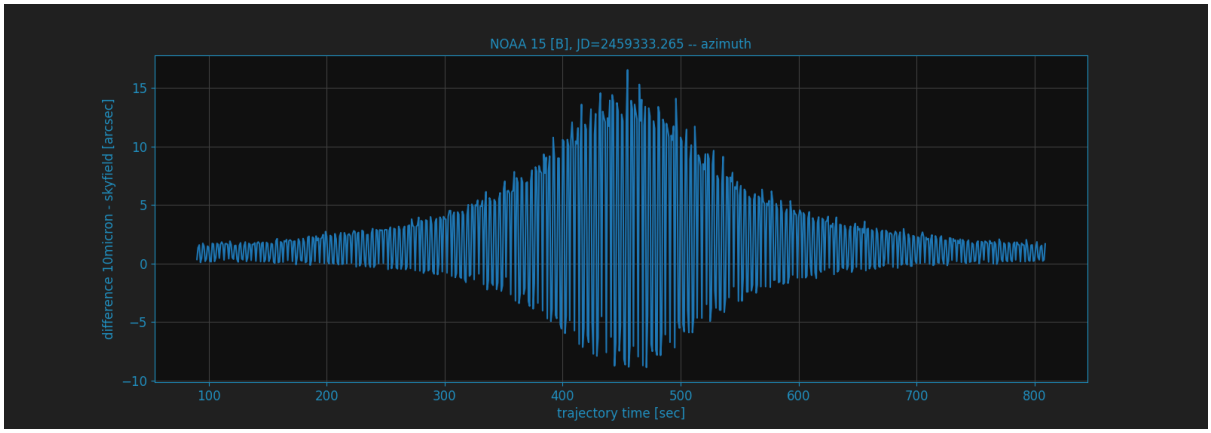
You could see the alt/az of the sat track.



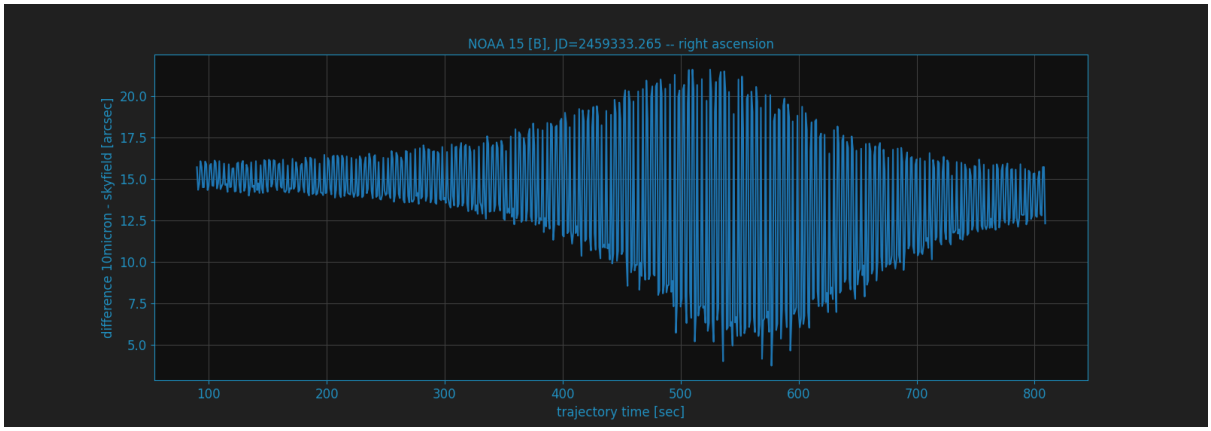
the difference for altitude between 10micron and skyfield



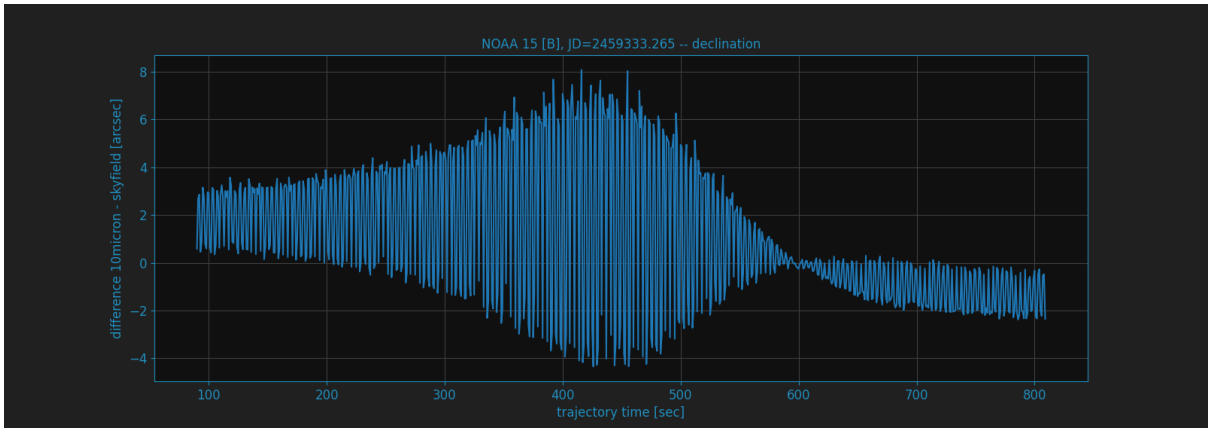
the difference for azimuth between 10micron and skyfield



the difference for right ascension between 10micron and skyfield



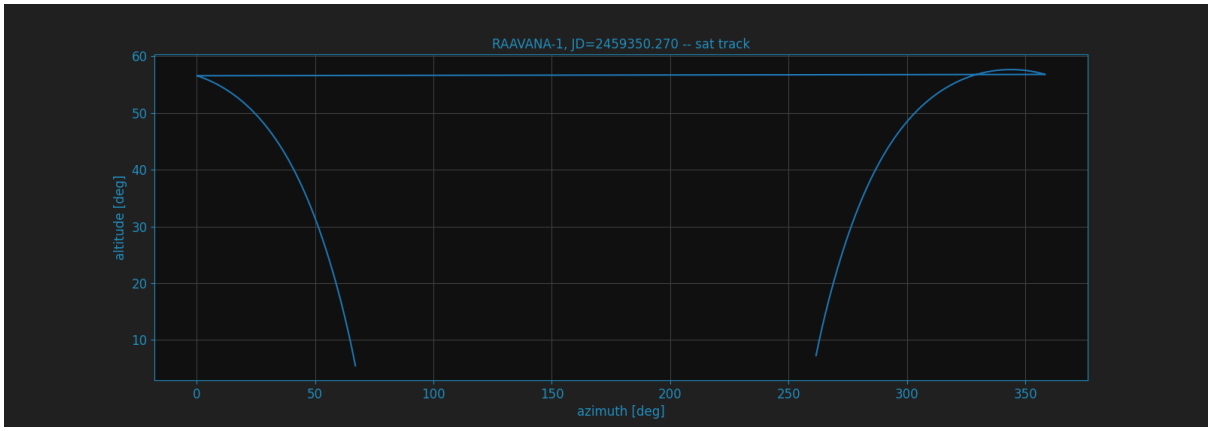
the difference for declination between 10micron and skyfield



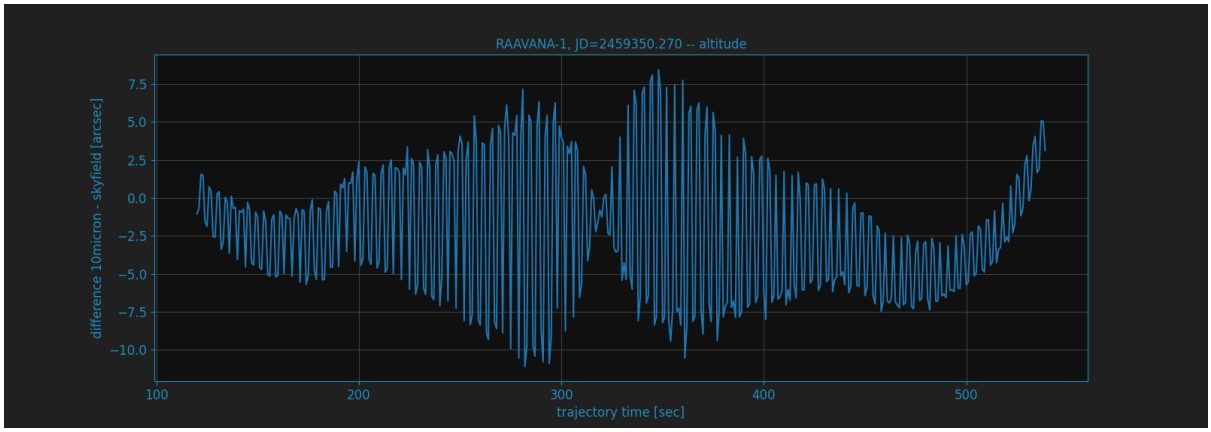
There is a set of plots for another satellite, which shows the same behavior. The used TLE data was:

```
RAAVANA-1
1 44329U 98067QE  21134.29933328  .00044698  00000-0  30736-3 0  9995
2 44329  51.6342 100.9674 0004554 122.3279 237.8162 15.74179130108776
```

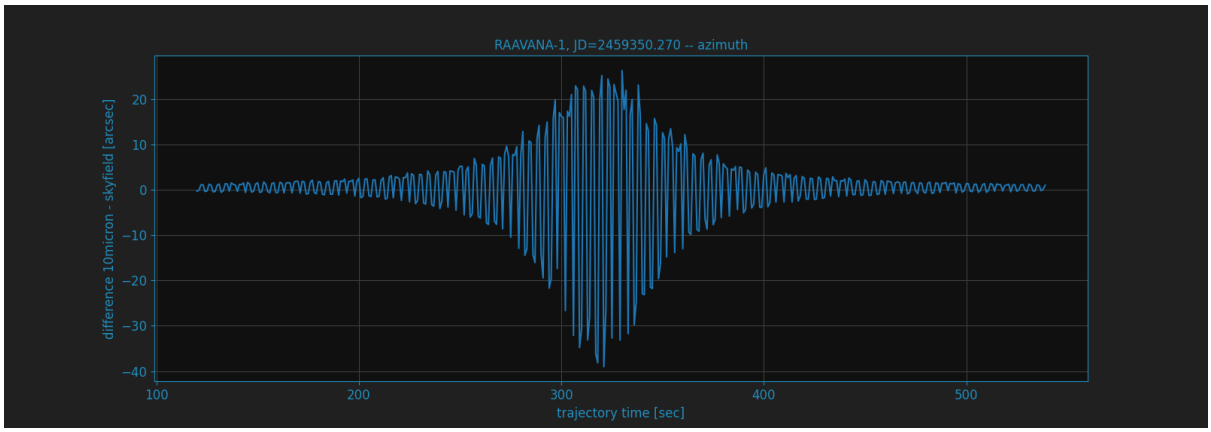
You could see the alt/az of the sat track.



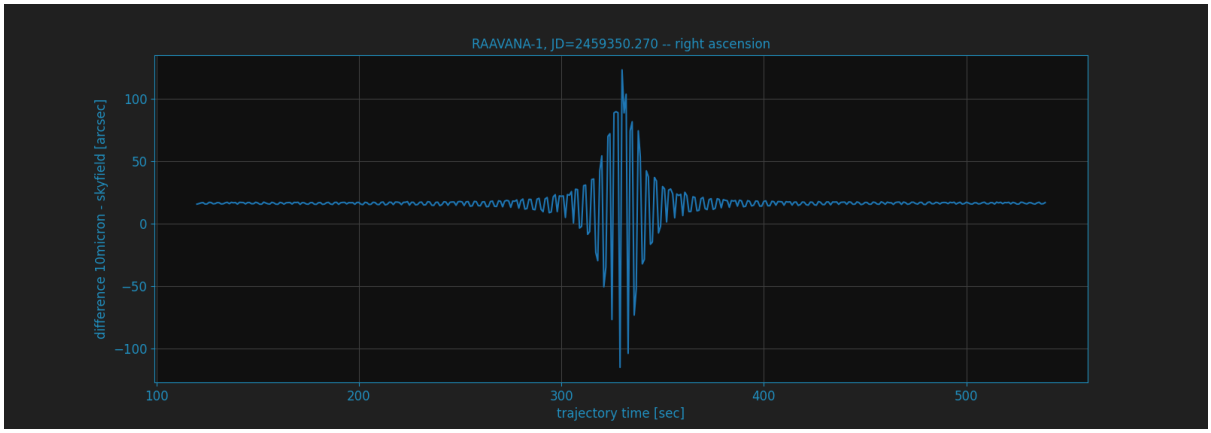
the difference for altitude between 10micron and skyfield



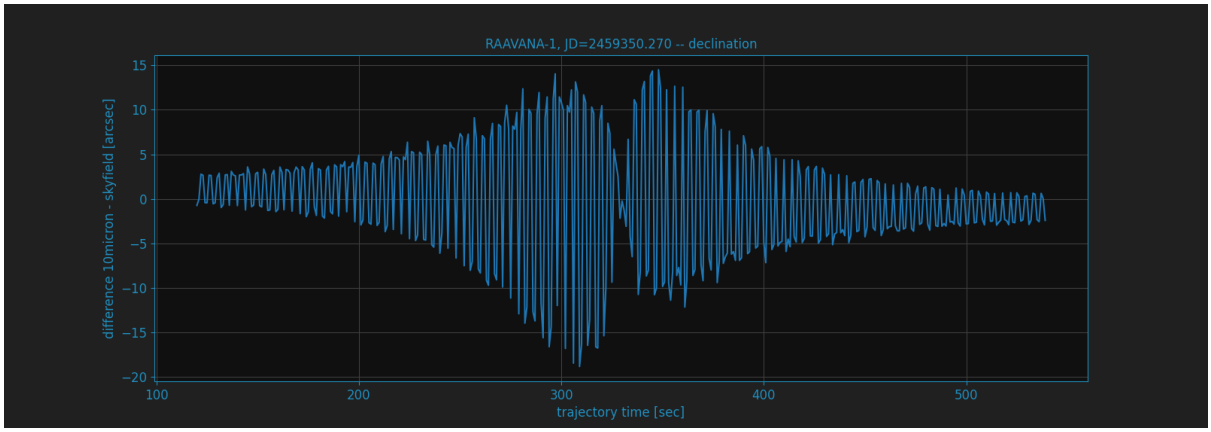
the difference for azimuth between 10micron and skyfield



the difference for right ascension between 10micron and skyfield



the difference for declination between 10micron and skyfield



For all calculations is valid:

- they are using refraction correction with the same values.
- the coordinates from 10micron are gathered with :TLEGEQJD#, :TLEGAZJD# commands
- julian date is in UTC time system
- 10micron firmware 3.0.4
- skyfield version 1.39

4.6 Changelogs

The changelogs contains the user related function or environment updates. For a detailed changes list, please refer to the commit list on GitHub.

4.6.1 Ideas for the future

- implementing Baader dome control e.g. 10micron dome interface
- receive video indi for sat tracking and horizon setup
- data exchange to EKOS SQLite for Location and horizon through SQLite interface
- satellite scheduler
- adding simple model feature for rainbow 135

4.6.2 Beta versions of MW4

- none so far

4.6.3 Released versions of MW4

Version 3.0

3.0.0

Version 3.0 is a major release! Please update with care! No ARM7 support / ARM64 only Python 3.8 - 3.9

- add: GUI: all charts could be zoomed and panned
- add: GUI: all tab menu entries could be customized in order and stored /reset
- add: GUI: all open windows could be collected to visual area
- add: GUI: separate window with big buttons are available
- add: GUI: reduced GUI configurable for a simpler user interface
- add: video: support for up to 4 external RTSP streams or local cameras
- add: video: adding authentication to video streams
- add: video: adding support for HTTP and HTTPS streams
- add: almanac: now supports UTC / local time
- add: almanac: support set/rise times moon
- add: environment: integrate meteoblue.com seeing conditions
- add: analyse: charts could show horizon and values for each point
- add: analyse: alt / az charts with iso 2d contour error curves
- add: audio: sound for connection lost and sat start tracking
- add: model points: multiple variants for edit and move points
- add: model points: set dither on celestial paths
- add: model points: generate from actual used mount model
- add: model points: existing model files could be loaded
- add: model points: golden spiral with exact number of points
- add: polar align: adding hint how to use the knobs measures right
- add: plate solve: new watney astrometry solver for all platforms
- add: hemisphere: selection of terrain file
- add: hemisphere: show actual model error in background
- add: hemisphere: edit horizon model much more efficient
- add: hemisphere: show 2d contour error curve from actual model
- add: hemisphere: move point with mouse around
- add: dome: control azimuth move CW / CCW for INDI
- add: satellites: all time values could be UTC or local time now
- add: MPC / IERS: adding alternative server for download
- add: measure: window has max 5 charts now (from 3)
- add: measure: more values (time delta, focus, cooler power, etc.)

- add: image: photometry functions (aberration, roundness, etc.)
- add: image: tilt estimation like ASTAP does as rectangle and triangle
- add: image: add flip H and flip V
- add: image: show RA/DEC coordinates in image if image was solved
- add: image: center mount pointing g to any point in image by mouse double click
- add: image: center mount pointing to image center
- add: image: support for reading XISF files (simple versions)
- add: imaging: separate page for imaging stats now
- add: imaging: stats: calcs for plate solvers (index files etc.)
- add: imaging: stats: calcs for critical focus zones
- add: drivers: polling timing for drivers could be set
- add: drivers: game controller interface for mount and dome
- add: system: support for python 3.10
- add: help: local install of documentation in PDF format
- add: profiles: automatic translation from v2.2.x to 3.x
- improve: GUI: layout for main window optimized and consistent and wording updates
- improve: GUI: complete rework of charting: performance and functions
- improve: GUI: clean up and optimize IERS download messages
- improve: GUI: get more interaction bullet prove for invalid cross use cases
- improve: GUI: moved on / off mount to their settings: avoid undesired shutoff
- improve: GUI: show twilight and moon illumination in main window
- improve: INDI: correcting setting parameters on startup
- improve: model points: optimized DSO path generation (always fit, less params)
- improve: model run: refactoring
- improve: model run: better information about status and result
- improve: hemisphere: improve solved point presentation (white, red)
- improve: plate solve: compatibility checks
- improve: system: all log files will be stored in a separate folder /log
- improve: system: enable usage of python 3.10
- improve: system: use latest PyQt5 version
- improve: system: adjust window sizes to be able to make mosaic layout on desktop
- improve: system: moved to actual jpl kernel de440.bsp for ephemeris calcs
- remove: system: matplotlib package and replace with more performant pyqtgraph
- remove: system: PIL package and replace with more powerful cv2
- remove: system: move from deprecated distutils to packaging
- remove: system: support for python 3.7 as some libraries stopped support
- remove: imageW: stacking in imageW as it was never used
- remove: testing support for OSx Mojave and OSx Catalina (still should work)
- fix: drivers: device selection tab was not properly positioned in device popup

Version 2.2

2.2.9

- fix: internal updater shows only alpha versions instead of betas

2.2.8

- fix: updates for supporting newer ASTAP versions
- fix: model run will cancel if solving fails
- fix: workaround ASTAP FITS outputs which are not readable via astropy
- update ephemeris file

2.2.7

- fix: text labels
- fix: getting min / max values from indi devices
- fix: updates for supporting newer ASTAP versions
- fix: model run will cancel if solving fails

2.2.6

- fix: reduce load in debug trace mode
- fix: model process stalls in some cases in normal mode
- fix: text labels
- fix: getting min / max values from indi devices

2.2.5

- fix: reduce load in debug trace mode
- fix: model process stalls in some cases in normal mode

2.2.4

- fix: remove race condition for large image file causing solve error in ASTAP
- fix: reduce load in debug trace mode

2.2.3

- fix: mount orientation in southern hemisphere

2.2.2

- fix: almanac moon phase drawing error

2.2.1

- update: builtin data for finals200.all
- fix: download iers data: fix file not found feedback

2.2.0

- add: support SGPro camera as device
- add: support N.I.N.A. camera as device
- add: two modes for SGPro and N.I.N.A.: App or MW4 controlled
- add: debayer (4 modes) all platforms (armv7, StellarMate, Astroberry)
- add: filter satellites for twilight visibility settings
- add: setting performance for windows automation (slow / normal / fast)
- add: auto abort imaging when camera device is disconnected

- add: missing cursor in virtual keypad window
- add: support for keyboard usage in virtual keypad window
- add: screenshot as PNG save for actual window with key F5
- add: screenshots as PNG save for all open windows with key F6
- add: query DSO objects for DSO path setting in build model
- improved: flexible satellite handling when mount not connected
- improved: show selected satellite name in satellite windows title
- improved: 3D simulator drawing
- improved: updater now avoids installation into system package
- improved: GUI for imaging tab - disable all invalid interfaces
- improved: redesign analyse window to get more space for further charts
- improved: Tools: move mount: better UI, tooltips, multi steps in alt/az
- improved: gui in image window when displaying different types
- improved: reduced memory consumption if display raw images
- improved: defining park positions with digit and improve gui for buttons
- improved: when pushbutton shows running, invert icons as well
- improve: moon phases in different color schemes
- upgrade: pywin32 library to version 303 (windows)
- upgrade: skyfield library to 1.41
- upgrade: numpy library to 1.21.4
- upgrade: matplotlib to 3.5.1
- upgrade: scipy library to 1.7.3
- upgrade requests library to 2.27.2
- upgrade importlib_metadata library to 4.10.0
- upgrade deepdiff library to 5.7.0
- upgrade wakeonlan library to 2.1.0
- upgrade pybase64 library to 1.2.1
- upgrade websocket-client library to 1.2.3
- fix: simulator in southern hemisphere

Version 2.1

2.1.7

- add: 12 build point option for model generation
- add: grouping updater windows upper left corner
- add: support for languages other than english in automation
- add: minimize cmd window once MW4 is started
- fix: KMTronic Relay messages

2.1.6 - add: explicit logging of automation windows strings for debug - add: showing now detected updater path and app - revert: fixes for german as they do not work

2.1.5

- fix: checking windows python version for automation

2.1.4

- add: enabled internal updater for astroberry and stellarmate
- add: temperature measurement for camera
- improved: logging for ASCOM threading
- improved: image handling
- fix: DSLR camera devices

2.1.3

- add: config adjustments for astroberry and stellarmate devices (no debayer)
- improved: logging for UI events

2.1.2

- fix: non connected mount influences camera on ASCOM / ALPACA
- fix: logging string formatting

2.1.1

- fix: for arm64 only: corrected import for virtual keypad
- fix: arrow keys on keypad did accept long mouse press

2.1.0

- add: hemisphere window: help for choosing the right star for polar alignment
- add: hemisphere terrain adjust for altitude of image beside azimuth
- add: angular error ra / dec axis in measurement
- add: device connection similar for ASCOM and ALPACA devices
- add: extended satellite search and filter capabilities (spreadsheet style)
- add: estimation of satellite apparent magnitude
- add: extended satellite tracking and tuning capabilities
- add: enabling loading a custom satellite TLE data file
- add: command window for manual mount commands
- add: sorting for minimal dome slew in build point selection
- add: setting prediction time of almanac (shorter reduces cpu load)
- add: providing 3 different color schemes
- add: virtual keypad available for RPi 3/4 users now
- improve: check if satellite data is valid (avoid error messages)
- improve: better hints when using 10micron updater
- improve: simplified signals generation
- improve: analyse window plots
- improve: rewrite alpaca / ascom interface
- improve: gui for running functions

- improve: test coverage
- remove: push time from mount to computer: in reliable and unstable
- fix: segfault in qt5lib on ubuntu

Version 2.0

2.0.6

- fixes

2.0.5

- fix: bug when running "stop exposure" in ASCOM

2.0.4

- improvement: GUI for earth rotation data update, now downloads
- improvement: performance for threads.
- improvement: added FITS header entries for ALPACA and ASCOM
- fix: removed stopping DAT when starting model

2.0.3

- improvement: GUI for earth rotation data update, now downloads
- improvement: performance for threads.

2.0.2

- fix: robustness against errors in ALPACA server due to memory faults #174
- fix: robustness against filter names / numbers from ALPACA server #174
- fix: cleanup import for pywinauto timings import #175
- improvement: avoid meridian flip #177
- improvement: retry numbers as int #178

2.0.1

- fix: MW4 not shutting down when dome configured, but not connected
- fix mirrored display of points in polar hemisphere view

2.0.0

- add new updater concept
- add mount clock sync feature
- add simulator feature
- add terrain image feature
- add dome following when mount is in satellite tracking mode
- add dome dynamic following feature: reduction of slews for dome
- add setting label support for UPB dew entries
- add auto dew control support for Pegasus UPB
- add switch support for ASCOM/ALPACA Pegasus UPB
- add observation condition support for ASCOM/ALPACA Pegasus UPB
- add feature for RA/DEC FITS writing for INDI server without snooping
- add completely revised satellite tracking menu gui

- add partially satellite tracking before / after possible flip
- add satellite track respect horizon line and meridian limits
- add tracking simulator feature to test without waiting for satellite
- add alt/az pointer to satellite view
- add reverse order for failed build point retry
- add automatic enable webinterface for keypad use
- add broadcast address and port for WOL
- add new IERS and lead second download
- add more functions are available without mount connected
- add change mouse pointer in hemisphere
- add offset and gain setting to imaging
- add disable model point edit during model build run
- update debug standard moved from WARN to INFO
- update underlying libraries
- update GUI improvements
- fix for INDI cameras sending two times busy and exposure=0
- fix slewing message dome when disconnected
- fix retry mechanism for failed build points
- fix using builtins for skyfield and rotation update
- fix plate solve sync function

Version 1.1

1.1.1

- adding fix for INDI cameras sending two times BUSY, EXP=0

1.1.0

- adding release notes showing new capabilities in message window
- adding cover light on / off
- adding cover light intensity settings
- reversing E/W for polar diagram in hemisphere window
- adding push mount time to computer manual / hourly
- adding contour HFD plot to image windows
- adding virtual emergency stop key on time group
- update build-in files if newer ones are shipped
- auto restart MW4 after update
- adding OBJCTRA / OBJCTDEC keywords when reading FITs
- upgrade various libraries

4.6.4 Released versions of scripts

The changelog contains the user related function or environment updates. For a detailed changes list, please refer to the commit list on GitHub.

Unreleased versions of scripts

3.0beta10

- enable all feature of scripts to one single startup.pyz file with parameters
- new script is limited to windows, mac and linux only. No support of Raspi or aarch64

Released versions of scripts

2.4

- windows only: improved robustness against local installs
- adding a clean system utility

2.3

- windows only: improved robustness against local installs

2.2

- adding support for astroberry and stellar mate (based on astroberry?)
- adding workaround on windows for removed python2 support in setuptools

2.1

- fixes

2.0

- update to requirements v2.0beta release of MW4

1.1

- update to requirements v1.1 release of MW4

0.4

- support for python 3.7 - 3.9, removing support 3.6
- added prebuild wheels for ubuntu mate 18.04 aarch64 for python 3.7 - 3.9
- added prebuild wheels for ubuntu mate 20.04 aarch64 for python 3.7 - 3.9

0.3

- adding more error log support
- improved ui

0.2

- added support functions for better error handling while using

0.1

- adding support for raspberryPi3 with ubuntu mate
- adding support for raspberryPi4 with ubuntu mate server / kubuntu

0.0

- adding support for automatic installation on windows
- adding support for automatic installation on ubuntu

- adding support for automatic installation on osx