ASTROART 5.0

Camera Interface
Version 5.43

TABLE OF CONTENTS

1 Manual Conventions

Welcome to the new Camera Interface of Astroart 5.0. This document contains information about control of CCD cameras, telescopes, focusers and filter wheels. The following conventions are used:

- The menu bar, sub-menu and floating menu items are highlighted in bold and embraced in square brackets: submenus commands are noted with an arrow (i.e. **[Tools] ⇒ [Star Atlas]**) this means that you should click on the **[Tools]** menu, then click **[Star Atlas]** on the submenu.
- The buttons to be pressed are highlighted in bold and enclosed in a box (i.e. **Setup** means to click the button labelled 'Setup').

2 Installation

The "CCD Camera Interface" is a plug-in for Astroart which provides control for your CCD cameras. To install this plug-in simply copy the file **piccdgui.dll** into the Astroart directory. To command a particular CCD camera you may also need a driver, available for free at Astroart web page:

www.msb-astroart.com

3 Let's Start

The Astroart Camera Interface contains several simulators for CCD cameras, Telescopes and Filter wheels, which let you test all functions indoor before using your real hardware. Take a look at Chapter 7 for a quick tutorial.

3.1 Hardware setup

At first check the camera position and linking: be sure to securely attach the camera to your focuser. Once you have safely connected the camera to the computer you can turn on all the hardware.
Now, start Astroart and select from the menu [Plug-In] ⇒ [CCD Camera]: this will display the *Setup Page* (see Fig.1)

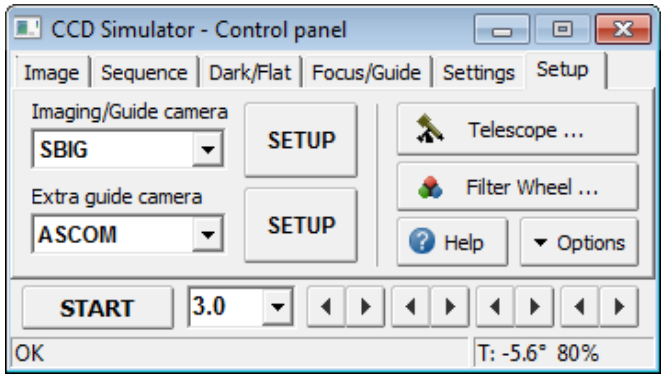
3.2 Software setup

The first option to select is the CCD model: select your CCD camera (you may need to download and install the appropriate CCD drivers[1]) then click on **Setup**. If the camera is not correctly detected, we suggest to consult the documentation of the CCD Driver, which is specific for every model. To make some indoor tests use the "Simulator" integrated in Astroart.

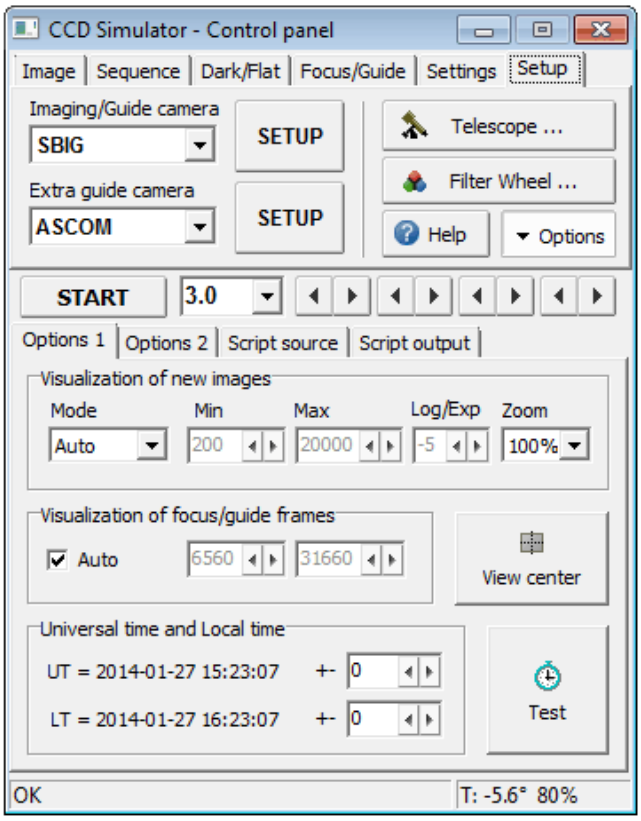
To command an auxiliary CCD camera click the second Setup button.

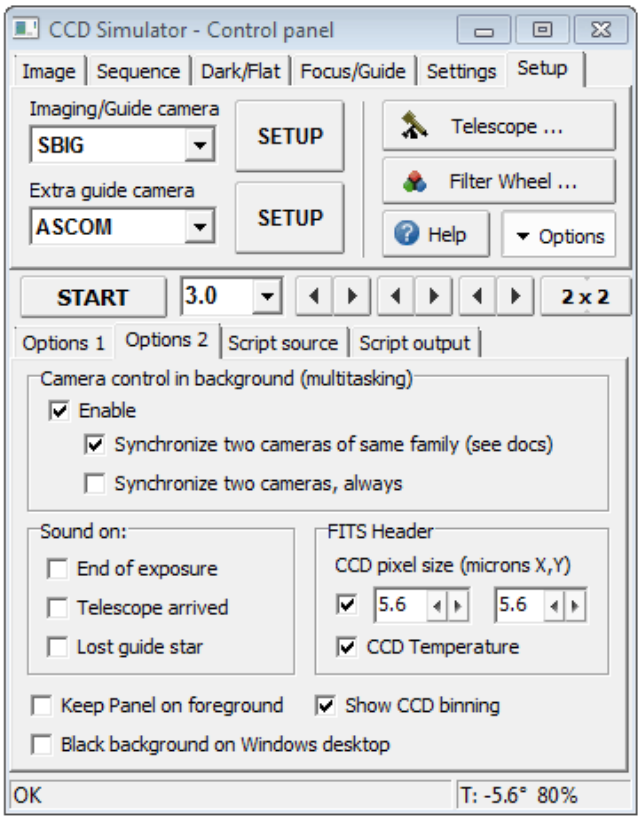
A further possibility to command two or more CCD cameras is to execute two sessions of Astroart. This can be very useful if you need to autoguide on the same subject for a long time (example: a variable star). The Astroart session which is guiding could be minimized to icon so it will not disturb the other Astroart session which remains free to take images and process them.

[1] Read carefully the instructions included with the CCD driver of the selected camera.



The Setup Page contains some hidden options which can be showed clicking the button **Options**.





Visualization (automatic and custom). The view mode for every new image. If ”Auto” is not selected then it’s possible to set the minimum and maximum visualization threshold and the transfer function for every new image, for more details see the Astroart Help The option “Auto (soft)” provides a natural look, “Auto (hard)” is useful for asteroid and supernova search.

UT and LT. Universal time and local time, as read from Windows. You may correct them if needed. The button “Check clock” displays for a few seconds the time of the PC to verify it with a precision better than one second.

Sound on. Plays a sound to alert the user about important events, likes the lost of the guide star.

Camera control in background. Enables multithreading for focus and autoguide. The options "Synchronize" are used when you are controlling two cameras. Synchronization is sometimes required when you are using two cameras of the same vendor AND their driver is not thread-safe. If you are using two ASCOM cameras from different vendors then disable this option. Synchronization reduces performance in one case: when you are focusing with the primary camera and autoguiding with the secondary camera at the same time.

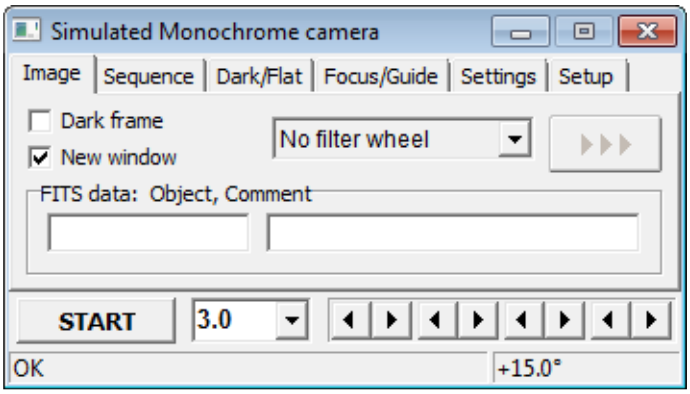
Black background. Displays a black frame below the Astroart desktop to hide the *Windows Desktop* and all other applications.

Keep panel on foreground. On some old PCs the CCD control window disappears behind the Astroart desktop, if this happens, select this option.

Show ccd binning. Displays a further button to quickly change the binning mode.

4 The Image Page

To start a new image select the *Image Page* and write with the keyboard an exposure time in seconds (example: “0.002” for two milliseconds) then click the Start button. The exposure time can be also set clicking the arrow buttons.



Dark frame. Select this option to close the shutter during the exposition. If your camera has not a built-in shutter you will have to cover the telescope. Click **Start** to integrate the dark frame, which will be displayed inside Astroart and stored in a private memory. If you select *”Enable dark frame correction”* option in the *Dark/Flat Page* every new image will be automatically corrected with this dark frame.

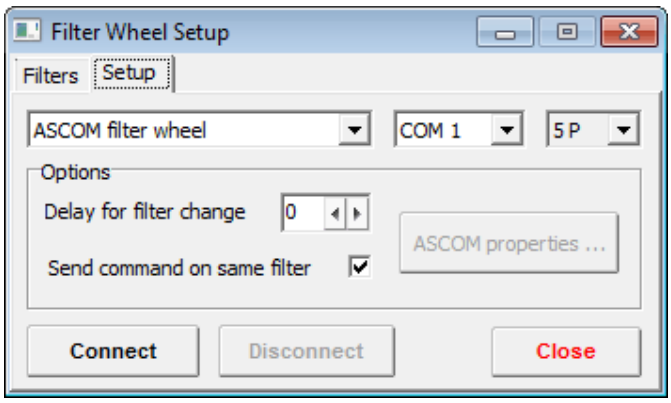
New window. Every new image is usually displayed into a new window. If this option is disabled then every new image will overwrite the previous one (if they have the same size). This is useful when doing sequences or test images.

Filter wheel. Select the filter name and click the arrow button to move the wheel to the desired position.

FITS Data. Here you can write the object name and a comment for the FITS header of the image. All other data like date, time, temperature (if supported by the driver) are added automatically.

5 Filter wheel

To activate the filter wheel select the *Setup Page*, click the **Filter wheel…** button, select the model, and click on **Connect**.



5.1 Setup page

Model. Select from the list your filter wheel. The list may change depending on the model of your CCD camera. To make indoor tests use the Simulator.

Com Port. If your filter wheel is controlled via the serial port, select here which port you are using.

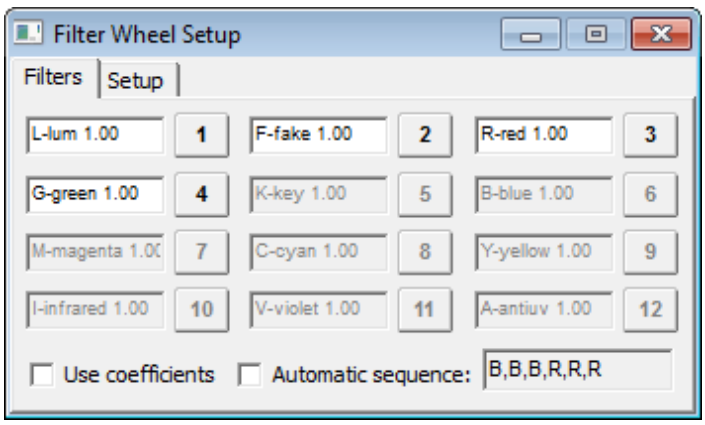
Positions. You may set how many filter positions are available, or just leave the default value for your model.

Delay for filter change. This number indicates how many extra seconds to wait for, after the wheel movements.

Send command on same filter. If enabled, Astroart sends the GOTO command to the filter wheel also when the correct filter is in position yet.

5.2 Filters page (sequences)

The following options are required during sequences, by the way they are not recommended since it's much easier to obtain the same result with a simple script.



Filter names. Write a capital letter which identifies the filter. Then write a minus sign followed by the complete name of the filter and a coefficient. The coefficient will be taken into account during a sequence as a multiplicative factor for the exposure time (set 1.00 to don't change the original exposure time).

Syntax: **`[Filter letter]-[Filter name] [Filter coefficient]`**
Example: `B-Blue 1.50`

Buttons. Click a button to verify that the wheel can turn to that position.

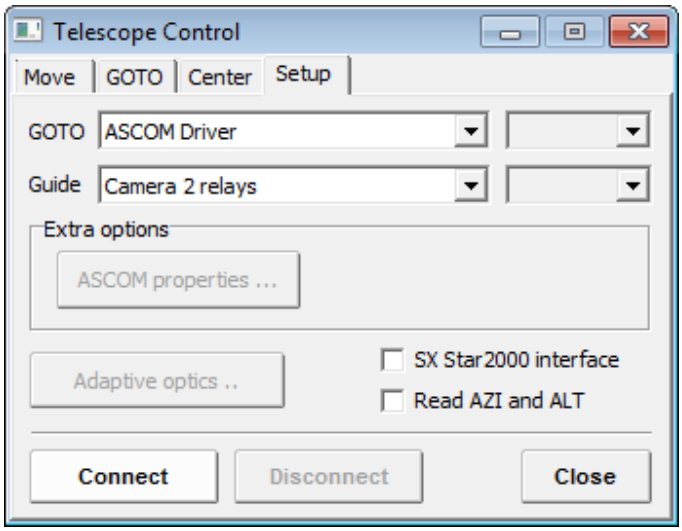
Use coefficients. If enabled, the exposure time of every image will be modified by the given coefficient. This may be useful to compensate the relative sensibility of the CCD with every filter.

Automatic sequence. If enabled, the filter wheel will be moved before every exposure of an automatic sequence and the first letter of the filter (example: R for Red) will be appended to the filename. Setting for example "R,G,B" these three filter will be used during a sequence.

6 Telescope Control

To control the telescope click the button **Telescope Setup…** in the *Setup Page (Fig.1)*.
Inside the Telescope Window select the appropriate protocol from the list and click on **Connect**. (Note that it's possible to also *connect* via the *Guide Window*, see chapter 7).
See also the Tutorial available in the next chapter.

Protocol/Interface. Select here the protocol used to command the telescope (many new mounts can emulate the GOTO protocol of the LX200, see the user manual of your hardware) and the secondary connection (if available) used to send guide pulses to the telescope.



- ***Camera Relays***. Relays integrated in some CCD cameras (SBIG, SXV, etc).
- ***Telescope Simulator***. A useful simulator to test indoor the Astroart telescope control.
- ***LX200 or GOTO compatible***. LX200 and GOTO mounts, STAR2000 in LX200 mode.

- ***LX200GPS new firmware***. Latest releases of the LX200GPS with RS232 autoguide commands.
- ***ETX or GOTO compatible***. Meade™ ETX telescopes and simple GOTO mounts.
- ***ASCOM driver***. Meade™, Celestron™, Astro-physics™, AstroOptiks™, Gemini™, SkySensor™ and many others. This option needs the Ascom libraries available at: http://ascom-standards.org/
- ***Celestron CGE/Nexstar***. Celestron CGE and Nexstar series.
- ***Starlight-Xpress Relay box, ST4 mode***. Autoguider box and STAR2000 relays output.
- ***Cookbook Relay Box 300 baud***. Relay boxes designed for the CB245 (300 baud).
- ***Cookbook Relay Box 9600 baud***. Relay boxes designed for the CB245 (9600 baud).
- ***Audine Relay Box***. Relay boxes originally designed for the Audine camera.
- ***VSSI direct cable***. Very Simple Serial Interface cable, a simple connection from the serial port to a telescope mount to correct the R.A. errors only. The signal DTR is the command RA+, the signal RTS is the command RA-. If the telescope is polar aligned this system is sufficient for every purpose.
- ***LX200 Shared port with SkyMap PRO***. An inter-process connection with the planetarium SkyMap™ PRO which allows both programs to share the same serial port.
- ***MTS-3***. PowerFlex MTS-3 (autoguide and CCD centering only).
- ***Interprocess Communication***. Interface with custom programs, see the Plug-in SDK.
- ***Verb Interface***. A simple and cheap interface ST-4 compatible, which is connected to the serial port of the PC (a USB->Serial converter may be used too). Does not require any power supply. Designed by P. Mergan, schematics available at the Astroart web site.
- ***StarlightXpress Adaptive Optics***. A great tool to guide your images without moving the telescope.
- ***Shoestring USB***. An interface which converts the USB port to a standard ST4 guide port (4 digital signals and ground). More details at: www.ShoestringAstronomy.com
- ***Shoestring Parallel port***. Shoestring GPINT parallel port adapter.

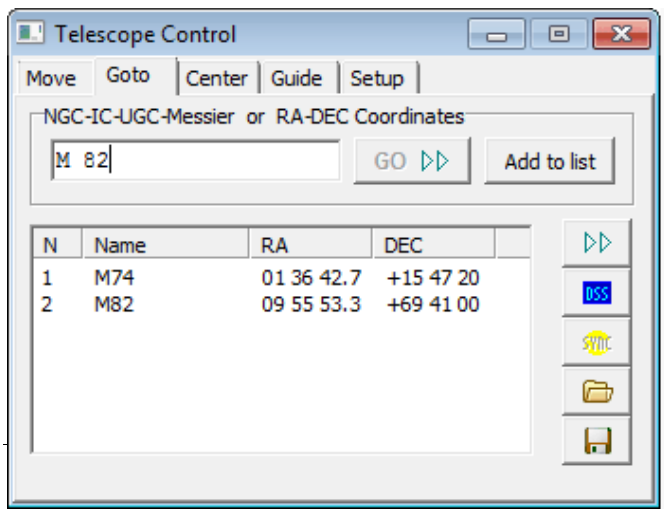
Adaptive Optics. Moves the AO mirror/prism to the middle position.

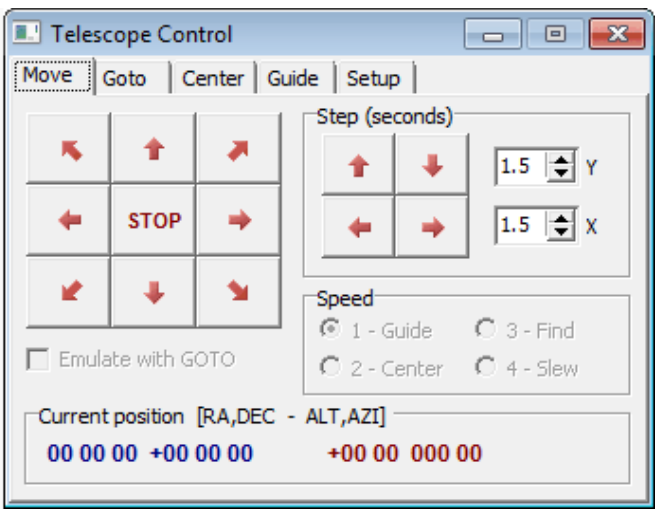
STAR 2000. Initializes the Starlight-Xpress™ STAR2000 interface.

Read ALT and AZI. If enabled, Astroart reads the horizontal coordinates from the telescope. This can be slow down the system on some mounts.

6.1 GOTO Control panel

This page contains some functions to control computerized mounts





The Goto and Move pages

NGC-IC-UGC-Messier. Type here the name of a deepsky object (example: N 4565, M 65, U 345 etc.) or some RA/DEC coordinates (example: 18 34.3 +34 56) then click **GO** to slew the telescope to that object. The blank space between the catalogue and the object number is always required ("N4565" won't be accepted). Enable the **JNow** option to convert the coordinates from J2000 to the current epoch.

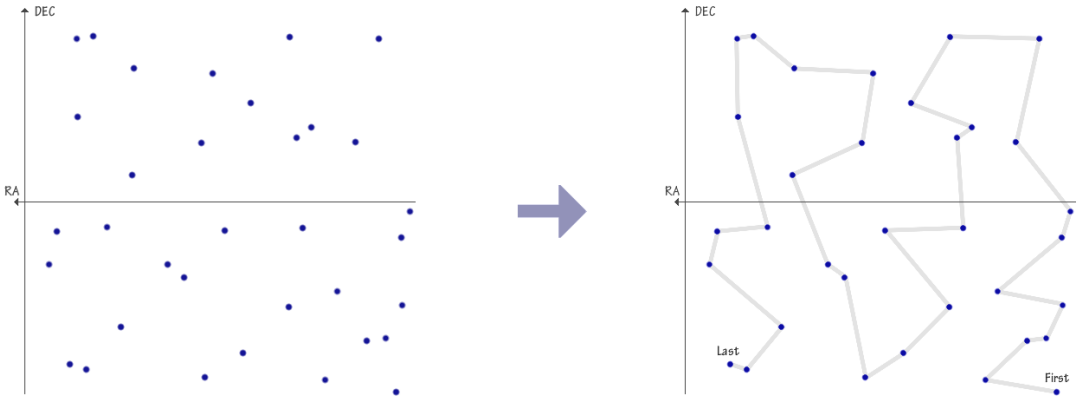
DSS. Opens a Digital Sky Survey image of the object. This function needs the DSS plug-in.

Sync. Sets the current position of the telescope to the last object coordinates.

Custom list. Manages lists of objects. To create a list write a text file (or a Excel file saved as .CSV) where every rows contains: *Name, RA, DEC*. The format of the coordinates is free, but the declination must contain the sign. The Name of the object cannot contain spaces, unless it's inclosed in quotation marks.
Example:

```
NGC4567 12 34.8 +78 45
"M 67" 12 34 56 +78 23.8
PK456+789 12 12.2 + 34 34 34
"UGC 3456", 12, 14, 16, +34, 54, 34      // Comments this way.
"UGC 4567", 12.23445, -23.23456
```

The last two rows are CSV [comma separated values] which can be exported and imported by Microsoft Excel.
To delete a row click the right mouse button. From the context menu you may also sort the list, it will be used a compromise between a shortest path algorithm and a increasing RA algorithm.



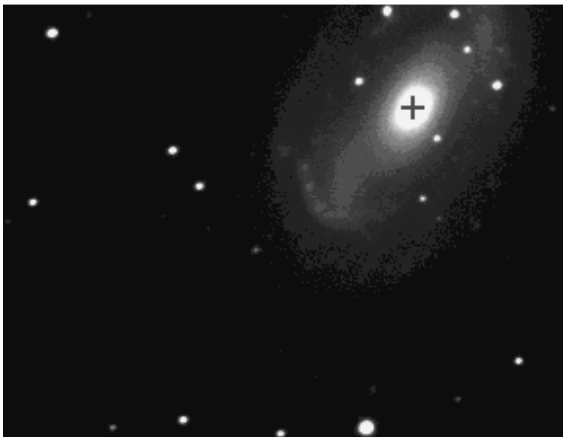
Arrow buttons. A "virtual keypad" to move the telescope, useful to center an object. The option "**Step**" will force the duration of every movement of the telescope to a given number of seconds, useful for mosaics and surveys. Please note that some ASCOM drivers do not support this function, in this case use the following mode.

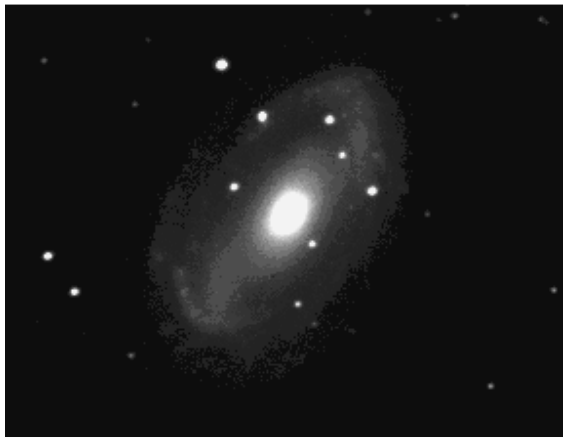
Emulate with GOTO. Select this option if your scope cannot be controlled for simple North, South, East, West commands, but it supports the GOTO protocol and the STOP-GOTO command. The speed depends on the telescope, probably 1 - 2 degrees per second.

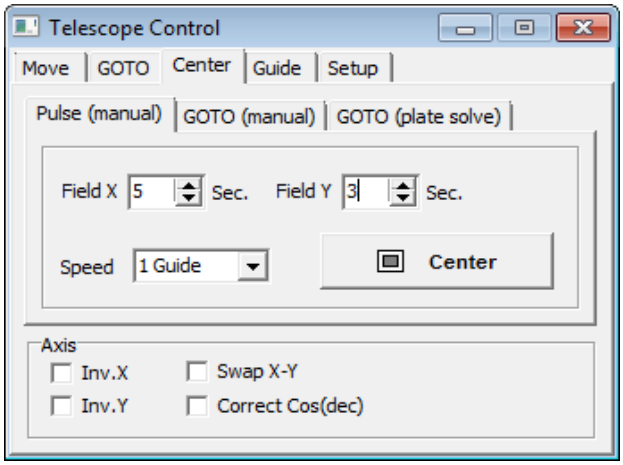
Speed. The telescope speed, this option is not supported by some telescope. Remember to select "guide" before autoguiding.

6.2 Autocenter

This function centers the object in the field of view of the CCD. You may obtain the same result using the script "Find coordinate" (plate solving) available in the next chapter.







Three methods are available to center the object. The first two methods are manual, you need to select a point over the object. The third method is automatic.

Pulse (manual).The telescope will be moved using the pulse commands (north, south, east, west) at the given speed. The field of view of the CCD image must be known in seconds (time), this parameter is simply the time necessary for the telescope to move along a full field. The field along the X axis can vary slightly with the declination, to correct this measure the field X at the equator and enable the option "Cos(declination)".

Goto (manual). The telescope will be moved with a GOTO command (Goto(ra+dx,dec+dy)). This method is less recommended since at the end of the GOTO it won't be possible to sync the position to the object. The field of view of the image must be known in arcminutes. The field X varies with the declination and it's possible to enable the option "Cos(declination)".

Goto (automatic). The telescope will be moved with a GOTO command, after having calibrated the image via plate solving. Before using this command you must to set the plate parameters in the window "Find Coordinates" from the Tools menu. See the Help file for more information.

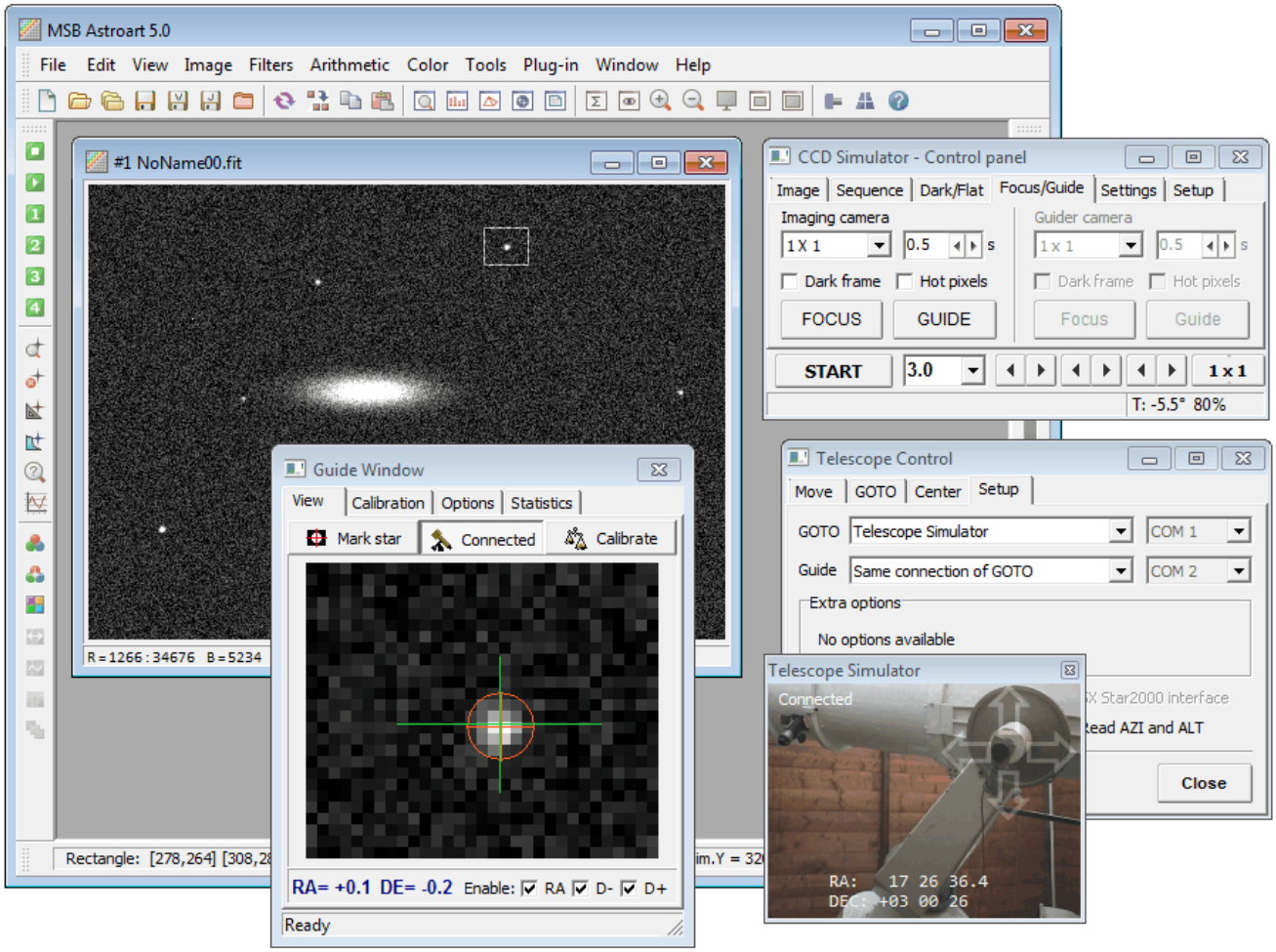
7 AutoGuiding

If you follow a star with your telescope at high magnification, you will notice that the position of the star changes. This movement is caused by three causes:

- A poor polar alignment which could cause a slow drift and a slow rotation of the field of view.
- The periodic error in the mount's tracking rate: this error results from gears that are slightly out of round. Some mounts have a built-in periodic error corrector called 'PEC'.
- The random errors due to many causes as dirt, dents and variations in the gears.

7.1 Tutorial

To quickly understand how the CCD and Telescope work together during an autoguide session, try this step by step tutorial:



1. Inside the Setup page of the camera control panel, Select "SIMULATOR" as CCD camera and click on "SETUP" to connect it.
2. Click on "Telescope ..", select "Telescope Simulator" as protocol and click "Connect".
3. Close the Telescope Window.
4. Click "Start" to start an exposure of 3 seconds.
5. Draw a small rectangle around a bright star and click "Guide" in the Focus/Guide page of the Camera Interface.
6. Click "Mark guide star" inside the Guide Window. The telescope simulator will now guide on the reference star.

7. To stop it, click the button "Connect Telescope" in the Guide Window. Verify that the star slowly drifts away. Click again the button to restart the guide.

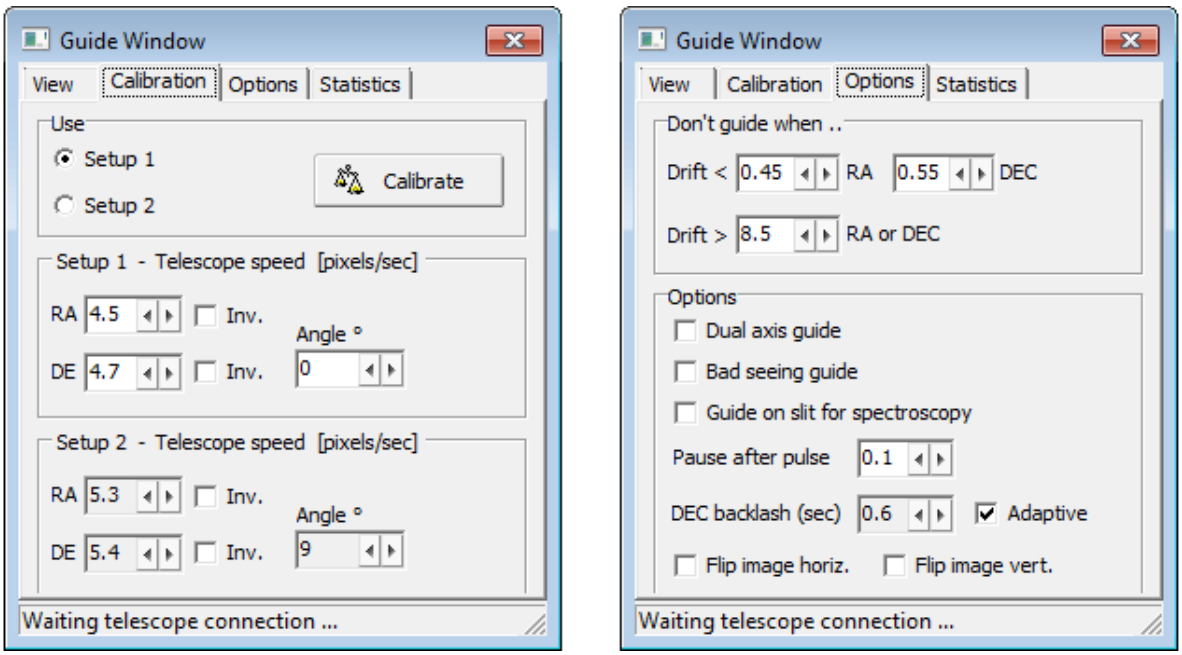
Repeat the procedure using the Guider simulator. During the guide, start a long exposure with the Camera simulator. Verify that both cameras work at the same time:

1. Close the Guide Window.

2. Click Setup (Extra guider camera) to connect the Guider Simulator.
3. Click "Guide" inside the "Guider camera" section of the Focus/Guide page to take a full exposure with the guider camera.
4. Select a rectangle around a star and repeat the procedure above.

7.2 Options

The most important settings are the telescope apparent speed in RA and DEC, and the rotation of the field of view. You may obtain them with the automatic calibration:



Telescope speed and Angle. The relative speed of the telescope in pixels per second. To measure this value simply move the telescope with the keypad for one second and measure how many pixels it moves. In Astroart this parameter is not critic, an error of ±50% will be automatically compensated, this means that there is no means to care about the *cos* δ factor.
The "Inv." options must be set if the autoguide frames are flipped horizontally or vertically. You may compensate this also with the "Flip Image…" options in the other panel.
The "Angle" parameter is the rotation of the camera relative to the equatorial axes, again an error of +- 30° will be compensated by the autoguide algorithm. Obviously it is possible to calculate all these values automatically with the Calibrate button.

Automatic calibration. Starts the automatic calibration to find out the telescope speed and the angle. It also detects inverted axes.

Don't guide when. A useful feature to prevent unwanted corrections for small drifts caused by bad seeing. 0.4 pixel is a good compromise between precision and rejection of noise. Values lower than 0.4 should be used only on a very short focal length (photo lens, for example).

Dual axis guide. If enabled, Astroart drives the telescope with two axes at a time. This feature is important for altazimuthal telescopes where both motors need to be controlled at the same time. For equatorial mount this option is not required because the DEC axis changes very slow depending the precision of the polar alignment, while fast DEC movements (wind, bad seeing) should be ignored.

Bad seeing guide. A special mode for extreme conditions (bad seeing, wind), it's not recommended for normal guide. If this option is not selected the guide will be adaptive in any case, but a different algorithm will be used.

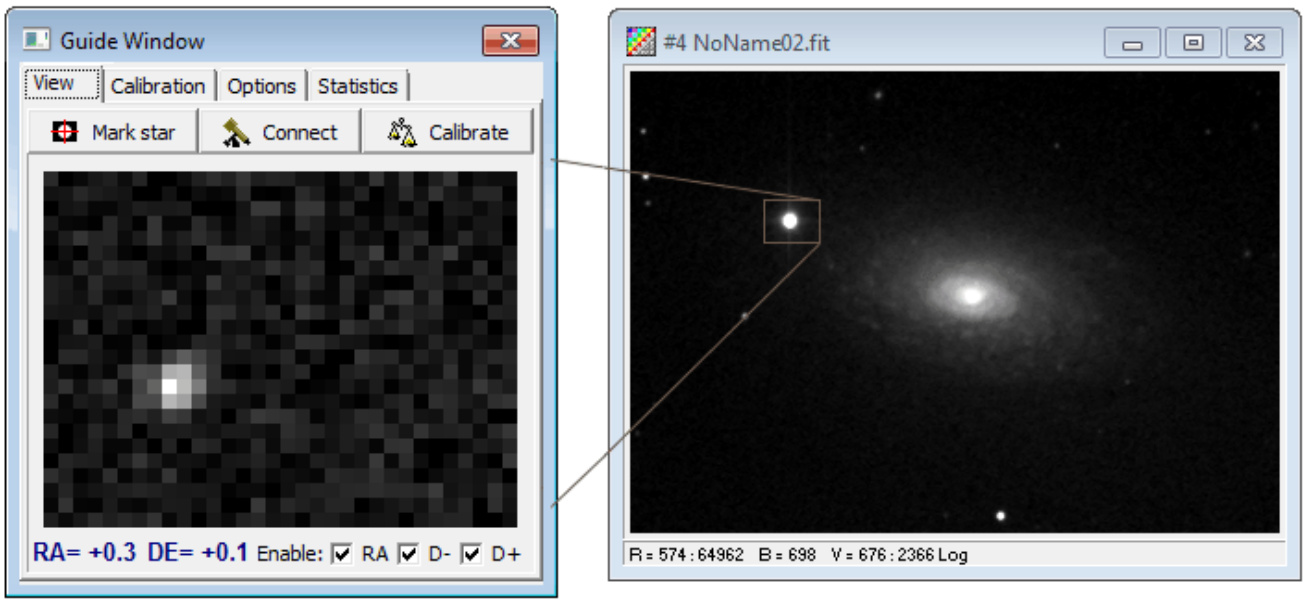
DEC backlash. Backlash may be a problem when the declination motor changes its direction. Usually Astroart corrects automatically the DEC backlash (the Automatic correction check box is enabled by default) unless you set a specific time compensation (in seconds). In this case be careful! Backlash corrections should be always under-compensated, an over compensation will cause an overshoot. In Right Ascension backlash is never a problem

7.3 Procedure

During an autoguided session, Astroart measures the position of a reference star (the guide star) on the tracking CCD, and sends the appropriate commands to both axes of the telescope to correct the tracking errors.

To start a session using one CCD camera follow this procedure:

1. Take an image with a short exposure time (example: one seconds): the exposure should be sufficiently long to identify a guide star in the field of view of the CCD, but short enough to let the guiding system works correctly with the same time: you should not exceed 1.5 - 2.0 seconds. To improve the sensitivity of your camera, change the binning from 1x1 to 2x2.
2. Draw a rectangle around the guide star. A brighter guide star will allow you to use a shorter *exposure guide time* but if your mount is accurate and very stable, you can use longer exposure and therefore dimmer guide stars.
3. Go to the Focus/Guide Page, select the same exposure time ("exposure guide time") and the same binning of your test image, then click the Guide button.
4. The guide star will appear in the *Guide Window.* Click on the **Mark guide star** button: a green cross will automatically mark the initial position of the guide star, while a red marker will follow the star while its shifts from the original position.



5. Click on the **Connect telescope** button to activate the telescope and to start the guiding session. Remember that the telescope parameters should have been previously set in the *Telescope Window*.
6. If needed, enable the Guide *RA* and Guide *DE* checkboxes. If the telescope is well polar-aligned you will obtain better results guiding only in RA. If the telescope in not polar-aligned you may enable only the DE+ or the DE- options (depending on the drift you see), this will prevent unwanted corrections caused by bad seeing.
7. If the corrections go in the wrong direction you'll probably need to execute the Autoguide Calibration from the Telescope Window.

The Autoguide Calibration calculates three parameters: SpeedX, SpeedY, Angle. In Astroart the autoguide is always adaptive so you don't need to calibrate at all, if the parameters are almost correct. You may calculate them by hand and reuse every night. By the way, if the system does not guide well, watch the behaviour of the star, this will reveal where the problem is, since only 3 scenarios are possible:

- **Overcorrection**. The scope is much faster than you have measured, so every correction brings the star beyond the central position. The solution is to increase the option Telescope speed or decrease the guide speed of the mount (if available); the best setting is 0.3 - 0.5 X (sideral), 1X is sometimes too fast.
- **Undercorrection**. The guide is "lazy" and the star is brought to the central position too slowly. This can also mean that the pulses are not reaching the telescope. To solve the problem measure again the *apparent telescope speed*.
- **Wrong direction**. after a few seconds the star is brought away from the center. The direction of one of the axis must be inverted. Enable the option "Inv. X" or the option "Inv. Y".

To discover the problem quickly it's strongly recommended to try to guide on one axis at a time, this will help to understand which problem affects the RA and DEC axis.

Tip: The *Guide Window* can be used as a recorder to measure the dX and dY errors of the mount (drifts). The dX and dY errors can be saved or copied into a text file. Select the Statistics page and click the Menu button. Rotate the mouse wheel to zoom in and out the graphs.

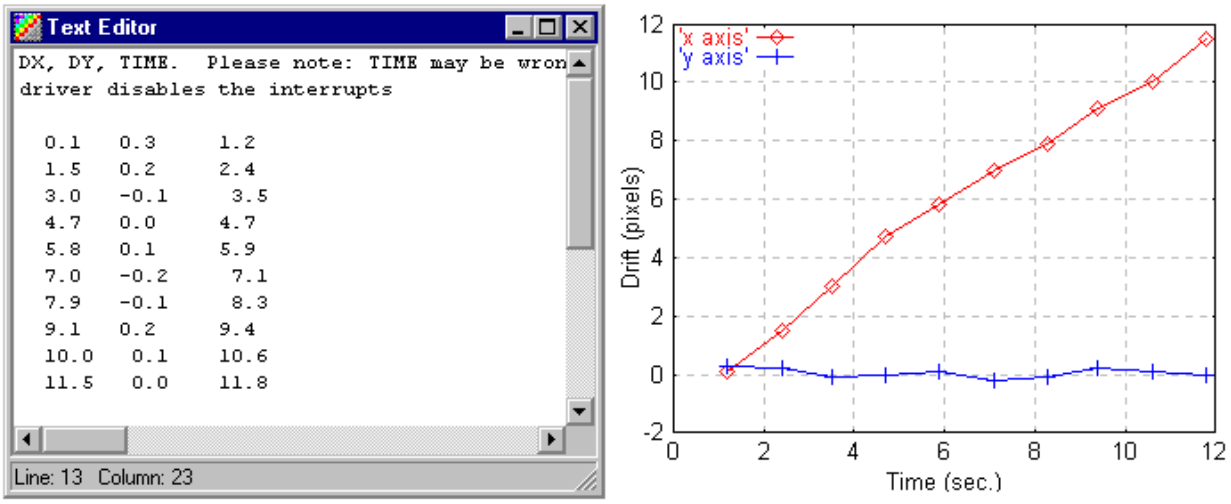
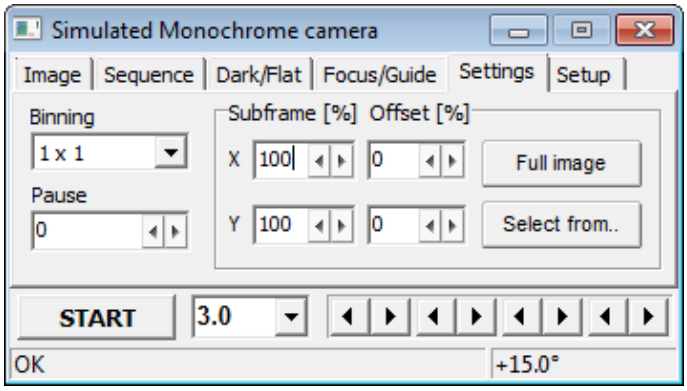


Fig. 1 – You can use the *Guide Window* as a recorder for guiding errors of your mount: in this simple example, within only 12 seconds we can see a large drift along the X axis (about 1pixel/sec)

8 The Settings Page

This page contains options about downloading images from the camera.



Binning. If 1x1 is selected then the CCD chip works at full resolution (example: 768x512 for a KAF400). If 2x2 is selected then four pixels will be grouped into one and the final image resolution is 384x256 (for a KAF 400). The advantage of binning is a faster download and better signal to noise ratio. 3x3 and 4x4 are useful to find or center an object.

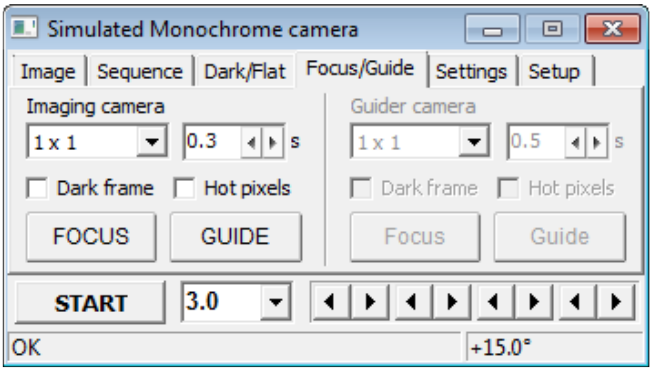
Delay. If this value is different from zero, then before each exposure there will be a pause of N seconds. This is useful if you need to change the setup before each image.

Sub-frame. This a useful feature for planetary imaging at high resolution. To speed-up the download and save space on disk it is possible to acquire only a part of the CCD array. To use this feature, download a full frame, select a rectangle on it, then click the **Select from...** button. The frame

boundaries will be written into the four edit boxes, (as percentage). You may also set these values by hand.

9 Focusing

Before focusing you need to integrate a full image and select a rectangle around a bright star, then click the “Focus” button.



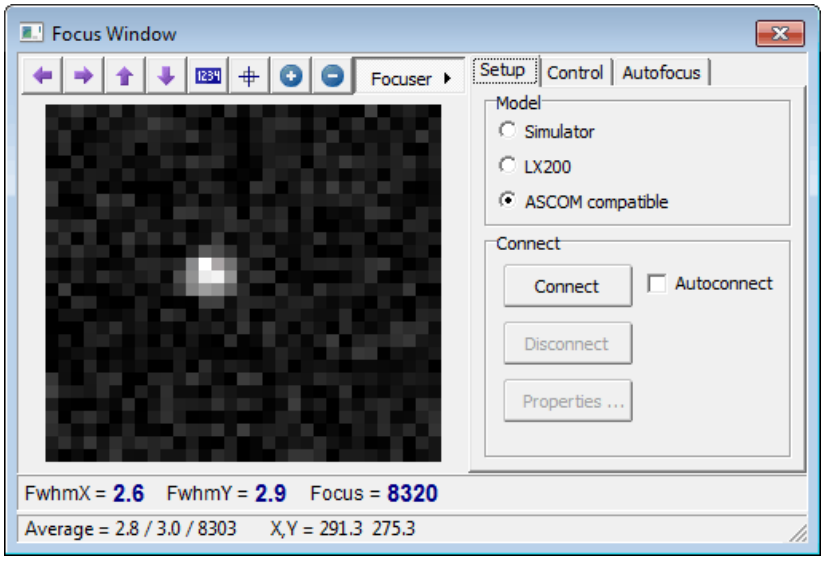
To take a full image with the main CCD camera click the **Start** button. To take a full image from the guider camera click the **Focus** or the **Guide** button.

Exposure. The exposure time for every frame (both for focusing and guiding).

Binning. The binning factor of every subframe.

Dark frame. If selected, a dark frame is acquired at the beginning of the focus session: (if the camera has not a shutter you should cover the scope before clicking the Focus button) Astroart will keep in memory the first frame as a dark frame and every subsequent image will be automatically corrected.

Hot pixels. Enables the automatic correction of hot pixels for every focus frame.



FwhmX, FwhmY. “Full width at half maximum” is a measure of the size of the star (in pixel) along the X and the Y axis.

Focus. A precise sharpness indicator: the higher the value, the better the focus.

X, Y. The coordinates of the star relative to the full image.

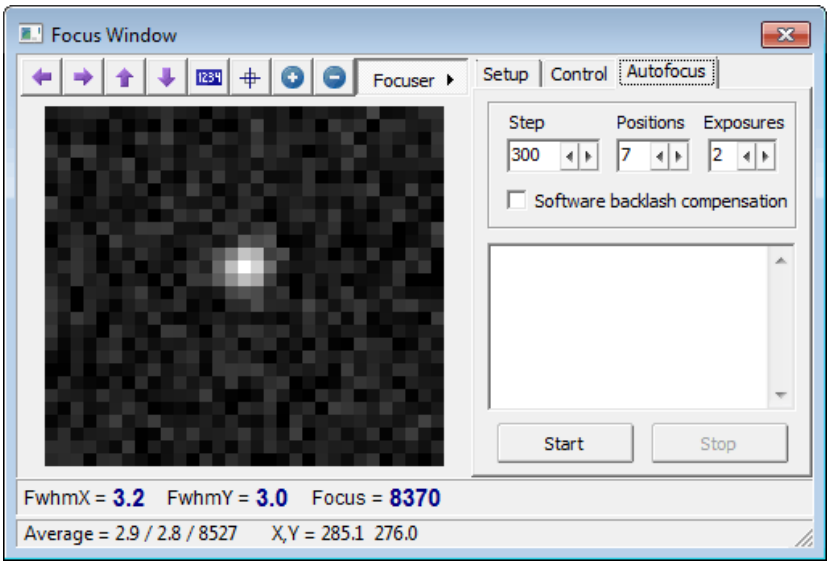
Average (FwhmX / Fwhmy / Focus). These three are the averaged values of Fwhm and the sharpness value, from the last subframes.

At the top of the Focus Window you can find some buttons: four arrow buttons to move the focus subframe across the CCD area. The [1234] button displays the sharpness parameter with a big font. The cross button displays a reference cross. The [+] and [-] buttons increase or decrease the exposure time.

Tip: the Focus Window can be used to center or find the object you are going to image: take a short exposure, select a rectangle as big as the whole image, go to the Focus/Guide Page (optional: select 4x4 binning for a fast download), click on the "Focus" button then move the telescope with its keypad.

9.1 Autofocus

It's possible to autofocus using any ASCOM-compatible focuser. Three parameters control the procedure:



Step. This depends by the range of your focuser, some have a range 0..5000 up to 0..50000. Usually 1/500 of the range is a good value.

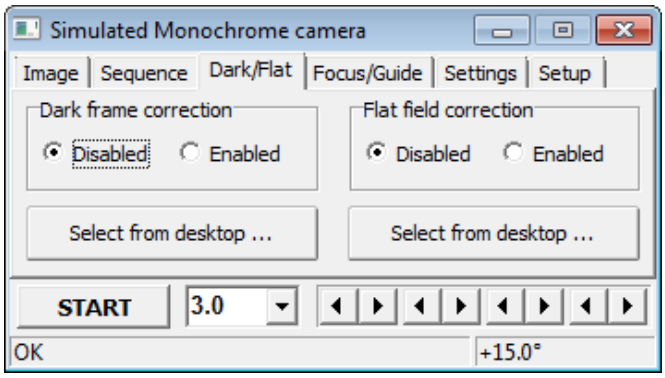
Positions. The number of positions to find the best focus. Usually it's an odd number. For example, if you set "7" the program will search: three positions back, the current one, and three ahead.

Exposures. How many exposures per position, this is useful when the seeing is not good. For every set of exposures the highest focus will be used. If the seeing is good, choose a value from 2 to 4, if it's bad, 4 to 6.

Software backlash. If enabled, the focuser always moves in the same direction (+). If your focuser has an integrated backlash correction (most focuser have) don't enable this option.

10 The Dark/Flat Page

This page enables the automatic correction of dark frames and flat fields during the acquisition of images.
This function is **not recommended**, since you will not be able to analyze the original raw images. It can be useful for live demonstrations.



Dark frame correction. Select this option to disable or enable the dark frame subtraction for every new image. If you change the binning factor the correction will be disabled.

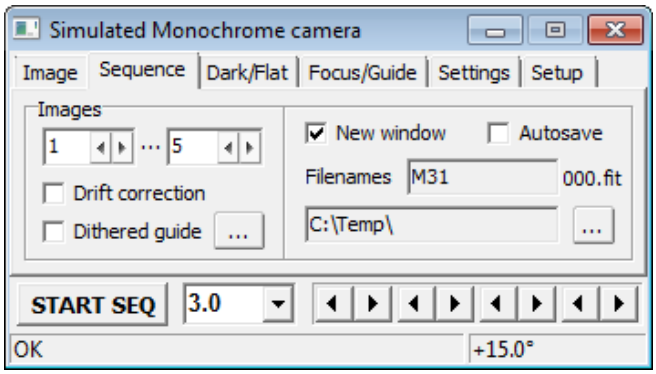
Flat field correction. Select this option to disable or enable the flat field correction for every new image

Select from desktop. Click this button to select a dark frame/flat field from the Astroart desktop, this is useful if you use your own sets of dark frames previously acquired.

For best results remember to take 5-10 dark frames and average them to reduce the random noise.

11 The Sequence Page

To improve the Signal-To-Noise ratio of CCD images it is often necessary to take a set of images and sum them. This can be done automatically from the Sequence Page.



Images. The number of exposures to be acquired, from 1 to 9999. It's also possible to set the initial index of the sequence.

Autosave. If selected every image will be saved with the filename specified in the edit box plus a ordinal number, into the chosen directory. Every image will be also opened into the Astroart desktop, if there is no need for this, disable *New Window* option.

New window. If not selected, all images will be displayed into a single window, to save memory. This is very useful if you need to take (and save) hundreds of images.

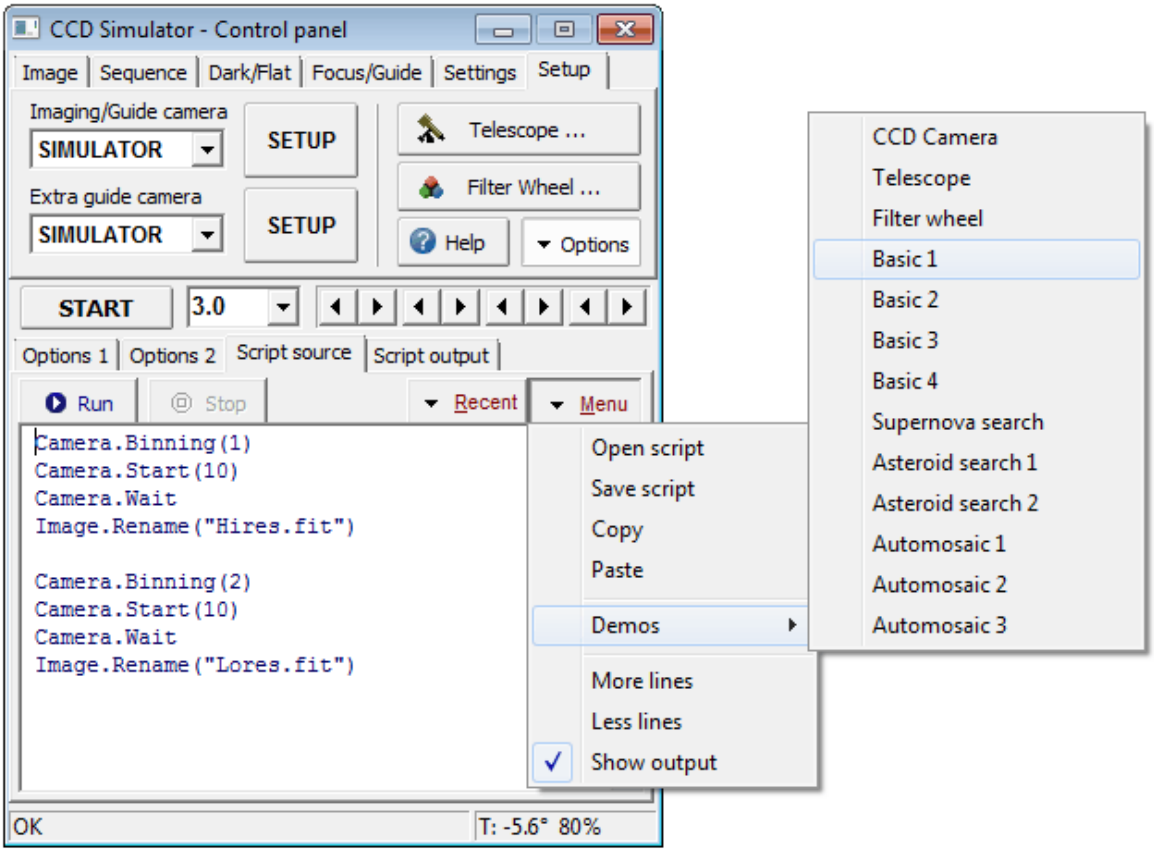
Dithered guide. This option can be useful during a sequence of autoguided exposures. Between each exposure the telescope will be move randomly by a given amount. This means that every image will not be aligned with the following one, but the signal to noise ratio will improve because different pixels will be used to collect data.

Drift correction. A useful feature for studying variable stars. After each exposure of the sequence the telescope will be moved to recenter the field. This allows long sequences (many hours) without the worry of drifts caused by bad polar alignment.

To stop a sequence click the button **Stop** once, and wait a few seconds.

12 Scripts

A script is a list of commands which are executed in sequence. Using scripts it's possible to perform very complex tasks, like automatic supernova and asteroid search, programming the telescope and the CCD camera.
The script language of Astroart is based on *Basic*. Its syntax is similar to QuickBasic, Visual Basic, VBScript, etc.



Example: (test it with the CCD simulator)

```
Camera.Start(10)
Camera.Wait
Image.Save("C:\sample.fit")
```

The first command starts a 10 seconds exposure, the second command waits until the end of the exposure, the last command saves the image. A further example: supernova search on 50 images.

```
for i = 1 to 50
 ra = Telescope.List.Ra(i)
 de = Telescope.List.Dec(i)
 name$ = Telescope.List.Name$(i)
 Telescope.Goto(ra,de)
 Telescope.Wait
 Camera.Start(60)
 Camera.Wait
 Image.Rename(name$ + ".fit")
next i
```

This simple script gets the coordinates and the names of the galaxies from the list in the Telescope Window. For every galaxy the script moves the telescope and starts the exposure.

To quickly learn the script features try all the demos available in the Menu. If required they will

automatically connect the CCD, Telescope and Filter wheel Simulators. This allows to make practice indoor.

If you enable the option "Show output" then the first Print() command will automatically switch to the "script output" panel.

12.1 Types, Variables and Functions

Two *types* are supported: numbers and strings. A numeric variable contains a number, a string variable contains a string.

Numeric variables.
They contain a number. The number is internally represented by a floating point value with double precision (64 bit, 15 digits).
Example:

```
x = 10.5
y = x + 1
```

String variables.
A string variable contains text. This text may be a single row or a multi-line text. The maximum size of a string variable is 64 MByte. String identifiers may be terminated by the dollar character, but this is not required.

Example:

```
a = "Hello"
b = a + "World"
```

The variable "b" now contains "HelloWorld"

You may use the dollar postfix to remember that a variable contains a string:

```
a$ = "Hello"
b$ = a$ + "World"
```

A single character of a string can be read using square brackets: a$[1] returns “H” and a$[2] returns “e” and so on. If the index exceeds the length of the string then it restarts from the beginning, so a$[6] returns “H” (a$ was “Hello”).

A single row of a multi-line string con be read using curly brackets.
Example, if a$ contains:

"This is a
text placed on
three rows"

Then a${2} returns “text placed on”. The function **count(**a$**)** returns how many lines are contained in a multi-line string.

Booleans.
Booleans are type-compatible with numbers. Just like in most languages, zero means "false" and every non-zero means "true".

Reserved words.

See the documentation available for VBScript, Visual Basic, QBasic or any other BASIC compiler.

IF THEN ELSE ENDIF OR AND NOT MOD REM FOR NEXT STEP BREAK CONTINUE WHILE ENDWHILE GOTO GOSUB PRINT INPUT END CLS SUB ENDSUB

Numeric functions.
See the documentation available for other BASIC compilers.

pi() sin(n) cos(n) tan(n) exp(n) ln(n) log10(n) log2(n) sqr(n) abs(n) rnd([n]) sgn(n) fix(n) int(n) round(n[,n]) frac(n) asin(n) acos(n) atan(n) atan2(n,n) sinh(n) cosh(n) tanh(n) asinh(n) acosh(n) atanh(n) degtorad(n) radtodeg(n) modulo(n,n) len(s) val(s) asc(s)

String functions.
See the demos for more information about these functions.

ucase$(s) lcase$(s) ltrim$(s) rtrim$(s) chr$(s) str$(n) mid$(s,n,n) hex$(n) left$(s,n) right$(s,n) ltab$(s,n) rtab$(s,n) format$(s,n) crlf$() opentext$(sfile) savetext$(s,sfile) appendtext$(s,sfile) copytext$(s) pastetext$()

Time and Coordinate functions.

JD() time$() date$() UTDateTime([jd]) SideralTime(lon,[jd]) EquatToAltaz(ra,de,lon,lat,[jd]) AltazToEquat(azi,alt,lon,lat,[jd]) PrecessionJ2000(ra,de) PrecessionJNow(ra,de) Precession(jdFrom,jdTo,ra,de) RA$(n) DEC$(n) DistanceFromMoon(ra,de) DistanceFromSun(ra,de) SunRaDec(jd) MoonRaDec(jd) ObjectCoordinates(name) ObjectMagnitude(name)

Other functions.

pause(n) speak(s) beep() playsound(s) message(s) createdir() finddir$(s,s) findfile$(s,s)

Camera and Telescope functions.

Function	Details	Example
Camera.Start(time[,shutter])	Starts an exposure of <time> seconds. Set <shutter> to zero to integrate a dark frame.	Camera.Start(60,0)
Camera.Wait	Waits until the end of the exposure.	
Camera.Exposing	Returns “1” if a exposure is in progress, otherwise “0”.	
Camera.Binning(mode)	Sets the binning mode. <mode> is a index to the binning list in the “Settings” page of the CCD panel.	Camera.Binning(2)

Camera.Subframe(w,h,x,y)	Sets a subframe for future exposures. All number are expressed as percentage. The following example sets a subframe of half size (50%) shifted by 25% so that it's centered on the CCD.	Camera.Subframe(50,50,25, 25)
Camera.SelectDarkFrame **Camera.EnableDarkFrame(enable)**	Selects the current image as dark frame and automatically enables the correction for the following images. Enables or disables the correction, 1 = enable, 0 = disable.	Camera.SelectDarkFrame() Camera.EnableDarkFrame(0)
Camera.Stop **Guider.Stop , Guider.Close**	Stops the current exposures or the guiding session.	
Camera.GetTemperature **Guider.GetTemperature**	Returns the current temperature of the CCD.	
Camera.SetTemperature(c[,p]) **Guider.SetTemperature(c[,p])**	Sets the temperature and starts cooling. It's also possible to set the power of the cooler (0..100%)	Camera.SetTemperature (-20)
Guider.MoveReference([dx,dy])	Changes the x and y coordinates of the reference star, to perform the "dithered guide". If dx and dy are not specified then the shift will be pseudo-random.	Guider.MoveReference() GuiderMoveReference(-0.3, 0.7
Guider.WaitStar([time])	Stops the script until the guide star is returned to the central position.	
Camera.Connect([driver]) **Guider.Connect([driver])** **Camera.Disconnect** **Camera.Connected** **Guider.Connected**	Connects or disconnects the CCD driver from Astroart.	Camera.Connect() Camera.Connect("Simulator")
Camera.StartAutoguide([r,x,y]) **Camera.StopAutoguide()**	Starts and autoguide session. It's possible to select the radius of the subframe and its coordinates, otherwise this command takes a sample image and selects automatically the best star.	r = Camera.StartAutoguide() if r <> 0 then [... autoguide is now active]
Focuser.Connect **Focuser.Disconnect** **Focuser.Connected**	Connects or disconnects the focuser.	Focuser.Connect Focuser.Autofocus Focuser.Disconnect
Focuser.Autofocus([x,y])	Starts an autofocus session. If x and y parameters (the coordinates of the focus	Focuser.Autofocus

	star) are not given then this command selects automatically the best star from the current image.	x = Image.GetPointX() y = Image.GetPointY() Focuser.Autofocus(x,y)
Focuser.GotoRelative(n)	Moves the focuser up or down by a specified amount.	Focuser.GotoRelative(-50)
Focuser.GotoAbsolute(n)	Move the focuser to a given coordinate.	Focuser.GotoAbsolute(1000)
Focuser.Position()	Returns the current position.	
Telescope.Connect **Telescope.Disconnect** **Telescope.Connected**	Connects or disconnect the telescope.	
Telescope.Goto(ra,dec)	Moves the telescope to the equatorial coordinates ra (0..24),dec. (-90..+90)	Telescope.Goto(23.4, 45.1)
Telescope.Wait	Waits until the telescope has completed a Goto.	
Telescope.Stop	Stops the telescope.	
Telescope.Ra **Telescope.Dec**	Returns the current position of the telescope.	x = Telescope.Ra y = Telescope.Dec
Telescope.Pulse(dir [,time])	Moves the telescope for <time> seconds towards the <dir$> direction (“N”,”S”, ”E”,”W”). If <time> is negative then the direction is inverted. If time is omitted, it moves until the Telescope.Stop command.	Telescope.Pulse(“N”, 0.5)
Telescope.Speed(n)	Sets the speed for Pulse motion. (1=guide, 2=center, 3=find, 4=slew)	Telescope.Speed(4)
Telescope.List.Open(file)	Opens a text file which contains objects and coordinates. See chapter 6.1.	Telescope.List.Open(“ c:\data\galaxies.txt”)
Telescope.List.Sort	Sorts the list, using a compromise between a shortest path algorithm and increasing RA algorithm.	
Telescope.List.Count **Telescope.List.Delete(n)** **Telescope.List.Clear**	Returns how many objects are listed in the Telescope Window. Deletes the N# element. Clears the list.	n = Telescope.List.Count
Telescope.List.Ra(index) **Telescope.List.Dec(index)**	Returns the coordinates of the <index>th object of the list. Returns -1 if index is not valid.	x = Telescope.List.Ra(25)

Telescope.List.Name$(index)	Returns the name of the <index>th object of the list.	a$ = Telescope.List. Name$(42)
Telescope.Send(string$) **Telescope.Receive$**	Sends o receives a string to the telescope via the serial port.	Telescope.Send("#Hc#") R$ = Telescope.Receive$
Telescope.Park **Telescope.Unpark**	Parks/Unparks the telescope. They work only with the GOTO or ASCOM protocol.	
Telescope.StartTracking **Telescope.StopTracking**	Starts/Stops the sideral tracking. They work only with ASCOM protocol.	
Telescope.AOCenter	Sets the AO mirror/prism to the central position.	
Wheel.Connect **Wheel.Disconnect** **Wheel.Connected** **Wheel.Filters**	Connects, disconnect and returns the number of filters of the filter wheel.	Wheel.Connect n = Wheel.Filters Wheel.Disconnect
Wheel.Goto(n) **Wheel.Goto($)**	Moves the filter wheel to the given filter.	Wheel.Goto(4) Wheel.Goto("R")
Image.Save(filename$) **Image.SaveView(filename$)**	Saves the current image as FITS. Saves the current image as jpg/png/bmp	Image.Save("C:\images\ saturn.fit")
Image.Rename(name$)	Renames the current image.	Image.Rename("jupiter.fit")
Image.Open(filename$)	Opens an image from disk	Image.Open("C:\moon.tif")
Image.GetKey$(key$) **Image.GetKey(key$)** **Image.SetKey(key$,value)**	Reads or Writes values from the FITS header.	a =Image.GetKey("NAXIS") Image.SetKey("COMMENT"," Bad seeing") Image.SetKey("JD",34234)
Image.FlipH **Image.FlipV** **Image.Resize(x,y)**	These functions modify the current image.	Image.Resize(320,240)
Image.BlinkAlign	Aligns the current image with the next one inside the Astroart Desktop and blinks them. Requires the Service Pack1.	Image.BlinkAlign
Image.Close	Closes the current image.	

Image.GetPointX() **Image.GetPointY()**	Returns the coordinate of the selected point (or star, or rectange) on the current image.	x = Image.GetPointX()
Image.DSS(ra,dec,name$)	Creates a new image from the Digital Sky Survey. Needs the DSS plugin.	Image.DSS(12.034,45.213,"asteroid.fit")
Image.Ra([x,y]) **Image.Dec([x,y])**	If the image is astrometrically calibrated, returns the ra/dec coordinates of the given point or the center plate (if the optional x,y parameters are not given)	ra = Image.Ra() de = Image.Dec() print "Center plate =",ra,de
Image.DistanceFrom(ra,de)	Returns the distance from the given coordinates to the center plate, in degrees.	See the Autocenter script.
Image.FindCoordinates (ra,de,stars,[side])	Finds the center plate via plate solving. The default side size is 2x2 degrees.	See Tools/Find coordinates and the Autocenter script.
Image.MaxValue **Image.MinValue** **Image.Average** **Image.StandardDeviation** **Image.Background**	Calculate statistics on the current image.	If Image.MaxValue > 65000 then ... < saturation is close >
Image.GetPixel(x,y) **Image.SetPixel(x,y,v)** **Image.Update**	Read and write single pixel over the image. After having written some pixels call the function Image.Update.	
Image.Macro(n)	Launches one of the four Macros defined in Astroart.	Image.Macro(2)
Output.Save(filename) **Output.Append(filename)** **Output.Copy** **Output.Text**	Saves the output panel to disk. Copies the output panel to the Clipboard. Returns the text of the output panel.	Output.Save("C:\Temp\Log.txt")
GuideLog.Text	Returns the log file of the Guide Window.	

Serial.Connect(port) **Serial.Disconnect()** **Serial.Send(string)** **Serial.Receive()**	Manages the serial port, to send direct commands to custom hardware. Connect and Send returns 0 or 1 in case of failure/success. Disconnection is automatic when a script terminates.	Serial.Connect(1) Serial.Send("#GH#) Pause(0.1) r$ = Serial.Receive$()
System.Execute(filename)	Executes a new program.	System.Execute("C:\Windows\Notepad.exe myfile.txt")
System.Broadcast(message$, wparam,lparam)	Sends a Windows Message to all windows. This can be used to control other programs. The function is equivalent to the Windows API: h = RegisterWindowMessage(message$) SendNotifyMessage(HWND_BROADCAST,h,wparam,lparam)	

12.2 Loop instructions

Two loop instructions are supported: **For** and **While**.
The easiest way to understand a loop is to look at an example.
The following program prints the numbers from 1 to 10, "a" is the control variable:

```
for a = 1 to 10
 print a
next a
```

The BREAK instruction.
Exits from a loop. In this following example the loop stops when "a" becomes greater than 5.

```
for a = 1 to 10
 print a
 if a>5 then break
next a
```

The CONTINUE instruction.
Acts as a NEXT instruction. A new iteration starts immediately.

```
for a = 1 to 10
 print a
 if a > 5 then continue
 print "Test"
next a
```

FOR-NEXT instruction.
The complete syntax for this command is:

```
FOR <variable> = <expression> TO <expression> [STEP <constant>]
...
```

```
...
NEXT <variable>
```

Examples:

```
for angle = 1+asin(0.4) to 1+asin(0.75) step 0.1
 print angle
next angle

s = 0
for y = 1 to 10
 for x = 1 to 20
  z = x*y : print z : s = s+z
 next x
next y
print s
```

WHILE-ENDWHILE instruction.
This instruction evaluates a condition at the beginning of the loop. If the condition is false then the cycle stops and execution continues after the ENDWHILE instruction.

Example:

```
a = 1
while a <= 10
 print a
 a = a+1
endwhile
```

Since the While command evaluates the condition at the beginning of the loop, the instructions inside the loop may be never executed. The *Break* and *Continue* instructions can be used in WHILE-ENDWHILE cycles just like in the FOR-NEXT cycles.

12.3 Conditional instructions

The IF-THEN instructions evaluates a logical expression and determines the flow of the program based on the result of that expression.
Example of logical expression:

```
a > 5 and b$ = ".fits"

a >= 3 or not (b <> 5 and b+3 = c)
```

Precedence of operators:

Higher precedence.
( )
< > <= >= <> =
NOT
AND
OR

Lower precedence.

Syntax.

```
IF <logical expression> THEN
...
...
[ELSE]
...
...
ENDIF
```

Compact Syntax.

```
IF <logical expression> THEN <instructions> ELSE <instructions>
```

IF instructions can be nested without limits. The ELSE part is optional.
Example:

```
for a = 1 to 10
 if a < 6 then print "-" else print "+"
next a
```

Example:

```
for a = 1 to 10
 if a < 6 then
   print "-"
   if a = 5 then print "Half work"
 else
   print "+"
   if a = 10 then print "The end"
 endif
next a
```

12.4 User subroutines

Subroutines can be defined and called within a script:

```
sub hello(x,y)
  print x,y
end sub
```

User subroutines can return one or two values, although this is not standard in many Basic dialects:

```
sub mySum(a,b,c)
  return a+b+c
end sub

s = mySum(4,6,2)
print "the sum is: "; s
```

Subroutines can be defined everywhere in the source code. The recommended place is at the end or at the beginning of the script. Another example:

```
sub PolToRect(r,a)
  return r*cos(a),r*sin(a)
end sub

x,y = PolToRect(10,pi/4)

myLon = -12: myLat = +45
ra,de = AltazToEquat(180,+5,myLon,myLat)   'calculates ra and dec at south, for parking.
```

12.5 Input-Output functions

PRINT function
Prints a text on the “Script output” panel of the CCD window.

```
PRINT [ expression [, expression] [; expression] ]
```

Every expression must be separated by a semicolon or comma; if semicolon is used then a space is added between the writings. If comma is used then a TAB separator is added between the writings.
Example:

```
print "Some math" ; 3+5 ; 4*4*4 ; sin(2.5)
```

INPUT function
Shows a dialog window where the user can enter data.

```
INPUT [ <Question> , ] variable
```

Use this function to ask the user for input values. If the user doesn't click the OK button then zero or an empty string is returned.
Example:

```
input "How old are you ?", a
input "Your name ?", n$
print n$ ; "is"; a
```

MESSAGE function
Shows a dialog window and wait until the user press “OK”.
Example:

```
message("Ready to start")
```

PAUSE function
This command pauses the execution of a script for a given amount of seconds. It can be useful after a telescope goto to wait the stabilization of the mount before starting the exposure.

The command PauseResume() resumes a pause inside another Astroart session which is running in the same PC. It can be useful for synchronization.

COMMENTS
To write a comment into a script use the symbol " ' "
example:

```
' This is a comment
' placed on two rows
```

12.6 Telescope and camera script

A script for automatic research controls the telescope, CCD camera and filter wheel. Usually it consists of three tasks.

1. Open a list of objects(example: Variable stars, Galaxies, etc.)
2. Setup a loop.
3. For every cycle, move the telescope, select the filter, take an image and save it.

For step (1) two methods are possible:

1. The function "Telescope.List.Open(filename$)" which loads a compatible list (see chapter 6.1). In the script use the commands "Telescope.List..."
2. The function OpenText$(filename$) which loads any text file into a string variable. This multi-line string can be parsed using the operators "{}" or the functions "Mid$", "Val", etc.

Example, using lists: (The object list was opened by hand).

```
n = Telescope.List.Count
for i = 1 to n
 ra = Telescope.List.Ra(i)
 de = Telescope.List.Dec(i)
 name$ = Telescope.List.Name$(i)
 print n; name$, ra; de
 Telescope.Goto(ra,de)
 Telescope.Wait
 Pause(4)
 Camera.Start(120)
 Camera.Wait
 Image.Save("c:\images\2004\" + name$ + ".fit")
 Image.Close
next i
```

See all the demo scripts inside the Camera Interface for more information.

12.7 Automatic align and blink

The function "blinkalign" automatically aligns two images and blinks them, to help you in comparing the images. This is useful for search of asteroids, supernovas and comets.
Example: open one image from a recent session, let's say "M67.fit", then execute the following script:

```
r$ = "D:\Astroimages\OldReferences\"
Image.Open(r$ + Image.Filename$)
Image.BlinkAlign
Image.Close
Image.Close
```

Where r$ is a folder of old images of the same field. Your reference image will be opened, compared to the new one, then both images will be closed. To compare another pair of images, open the next one from the recent folder and execute the script again.

The whole procedure could be make fully automatic iterating through all images of the recent folder, here is a sample script:

```
n$ = "D:\Astroimages\NewImages\"
r$ = "D:\Astroimages\OldReferences\"
im$ = FindFile$(n$,"*.fit")
n = Count(im$)
for i = 1 to n
  Image.Open(n$ + im${i})
  Image.Open(r$ + im${i})
  Image.BlinkAlign
  Image.Close
  Image.Close
Next i
```

12.8 Autocenter script

The autocenter script centers the object after a GOTO, using the “Find coordinate” (plate solving) feature of Astroart 5. There are two versions of the script, the first one uses the “resync” command of the telescope:

```
' < Get ra,de from your object list >
' < nstars = 4, or 5 if you set an high error >
' <                  in Find coordinates parameters >
....
Image.FindCoordinates(ra,de,nstars)
dist = Image.DistanceFrom(ra,de)
if dist > 0.1 then
   print "Centering telescope..."
   Telescope.SyncTo(Image.RA,Image.DEC)
   Telescope.Goto(ra,de)
   Telescope.Wait
Endif
```

The function “Image.FindCoordinates()” calibrates the image via plate solving. The result is “1” on success or “0” for failure. In the example above we don’t check the return value since the check is performed by the next instruction:

The function “Image.DistanceFrom()” calculates the distance in degrees between the center plate and the given coordinates. If the image is not calibrated, it returns “0”. As you can see, if the distance is

higher then 0.1 degrees (6 arcminutes) the telescope is synced to the corrected coordinates and a small GOTO is executed to the object.

This script is safe since a failure in FindCoordinates() does not move the telescope to a wrong position. Depending your system you may reduce the maximum distance to 0.05 degrees (3 arcminutes). Before using this command it's mandatory to calibrate the Find Coordinates parameters, as explained in the Astroart manual.

Another script is the following one: it doesn't need a centering GOTO, nor a resync, but the first displaced image is not corrected so the maximum "dist" must be smaller.

```
raOff = 0
deOff = 0
...
' <start of cycle>
' <raObj,deObj = read from the object list>
 ra = raObj - raOff
 de = deObj - deOff
' <Telescope GOTO towards ra,de>
 Image.FindCoordinates(ra,de,nstars)
 dist = Image.DistanceFrom(ra,de)
 if dist > 0.05 then
    raOff = Image.RA - raObj
    deOff = Image.DEC - deObj
 endif
 ....
```

This script simply keeps track of the difference between the real coordinates (as measured by FindCoordinates) and the object coordinates. This difference is then stored in raOff and deOff to correct the next GOTO. This make since the error increases with time, so this script is useful on a list of many objects (e.g. supernova search).
Measuring the offset in RA and DEC may be used in the first example too. If your telescope does not support the Resync() command, then you may emulate it with the sequence:

```
....
if dist > 0.05 then
    raOff = Image.RA - raObj
    deOff = Image.DEC - deObj
    Telescope.Goto(raObj - raOff, deObj - deOff)
    Telescope.Wait
 endif
....
```

13 History

2015/08/11 – V 5.43. Sorting of telescope lists with shortest path. Improved precision of deepsky coordinates. Plate-solved telescope centering with JNow. Script FindCoordinates with custom field size.

2015/02/12 – V 5.42. New script commands: Image.MinValue, MaxValue, Average, Background, GetPixel, SetPixel, Macro. Catalog of star names for telescope GOTO.

2014/10/28 – V 5.41. New script commands: PlaySound, ObjectCoordinates, Guider.WaitStar. The script editor now automatically scrolls to the current line during execution.

2014/05/12 – V 5.40. New options for autoguide, improved support for Windows 125% and 150% scaling, script commands: Camera.GetTemperature, Telescope.Tracking, SunRaDec, MoonRaDec.

2014/01/31 – V 5.30. Compatible with the new SDK. Goto in JNow coordinates. Script commands: SideralTime, Output.Append, GuideLog.Text, AppendString. Minor improvements in the user interface.

2013/11/10 – V 5.21. New script commands: Telescope.AOCenter and Guider.WaitStar. Fixed minor bugs in the user interface.

2013/09/10 – V 5.20. Integrated focuser control, improved autoguided sequences. Script commands for Altaz-Equat conversions and JNow.

2013/06/27 – V 5.10. Autoguide and focus in background (multithreading). Dedicated user interface for main and guider camera.

2013/05/31 – V 5.04. New scripts commands: Telescope.UnPark, Telescope.StartTracking, Telescope.StopTracking, fixed a bug in Input command. Improved SynScan support.

2013/05/25 – V 5.03. Improved scripts: user subroutines, new command Image.SaveView, menu for recent files.

2012/11/19 – V 5.02. Automatic center via plate solve, new options for dithered sequences, new options for the script editor. Script commands Focuser.Read() and JD() for julian date.

2012/05/09 – V 5.01. Option “Hot pixels” for focusing and autoguide. Script commands DistanceFromMoon, DistanceFromSun. Script commands for serial port control.

14 Contact Information

MSB Software.
Via Goetz 93, Classe
48124 Ravenna RA - Italy
Tel/Fax +39 0544 527265
WEB: http://www.msb-astroart.com
WEB: http://www.msbsoftware.it
E-mail: info@msbsoftware.it