Astroart 6.0

Camera Interface 6.01

User's Manual

Table of contents

1 Camera control

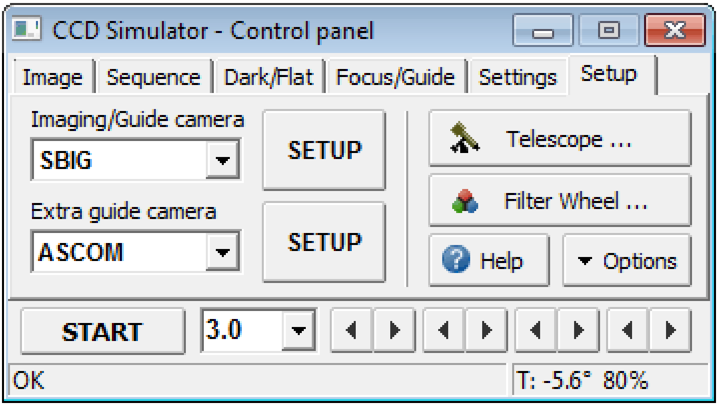
Camera control in Astroart is implemented as a "plug-in", this solution allow us to support new cameras very quickly.

1.1 Installation

To install a plug-in, unzip it into the Astroart installation folder (usually: C :\ Program Files x86 \ Astroart). The documentation is included in the same zip archive.

1.2 Setup page

The first option to be selected is the camera model, then click on "Setup". If the camera is not correctly detected, we suggest to consult the documentation of the plugin, which is specific for every model. To make tests, you may also select the Simulator.



To command an auxiliary camera click the second Setup button.

A further possibility to command two or more cameras it to execute two sessions of Astroart. This can be very useful if you need to autoguide on the same subject for many hours (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.

Options

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. The option “Auto (soft)” provides a natural look, “Auto (hard)” is useful for asteroid and supernova search. If ”Auto” is not selected then it’s possible to set the minimum and maximum visualization threshold and the transfer function.

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 at the same time. 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.

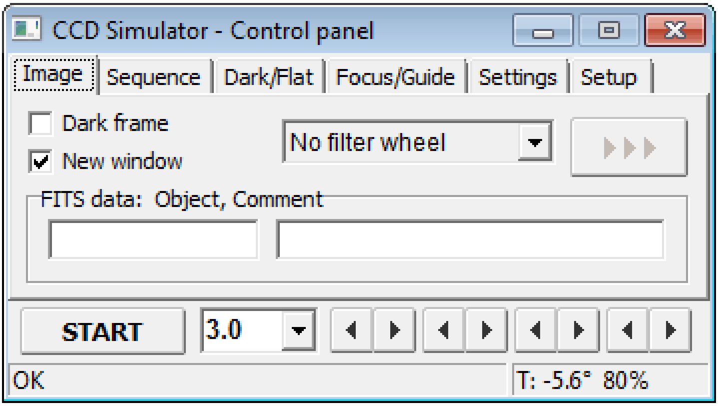
Black background. Displays a black frame below the Astroart desktop to hide the Windows Desktop and all the other applications.

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

Restore panel on top. On some old PCs the Camera control window may disappear behind the Astroart desktop, if this happens, select this option.

1.3 Image page

To start a new exposure select the *Image Page,* write with the keyboard an exposure time in seconds (example: "0.002" for two milliseconds) then click the "Start" button. The exposure time can also be 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.

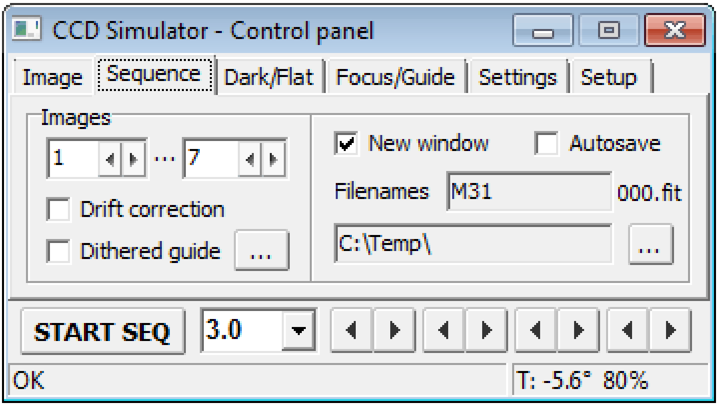
New window. Every new image is usually displayed in a new window. If this option is disabled then every new image will overwrite the previous one (if they have the same size). This is useful when doing sequences or test images.

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 FITS data like date, time, temperature (if supported by the driver) are added automatically.

1.4 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 an ordinal number, into the chosen directory.

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.

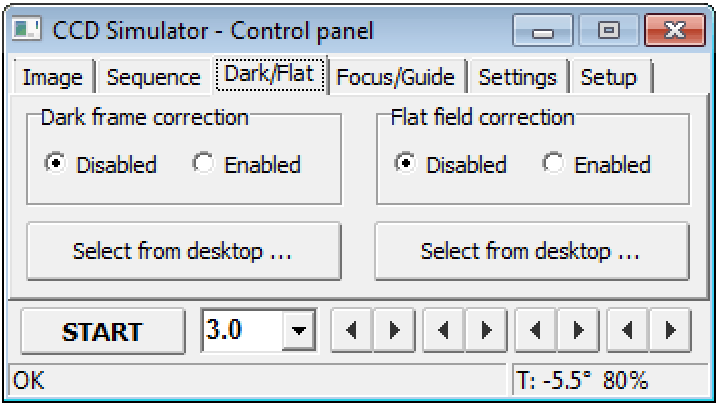
Dithered guide. This option can be useful during a sequence autoguided exposures. Between each exposure the telescope will be move randomly by a given amount. This means that every image will not be aligned with the following one, but the signal to noise ratio will improve because different pixels will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

1.5 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.

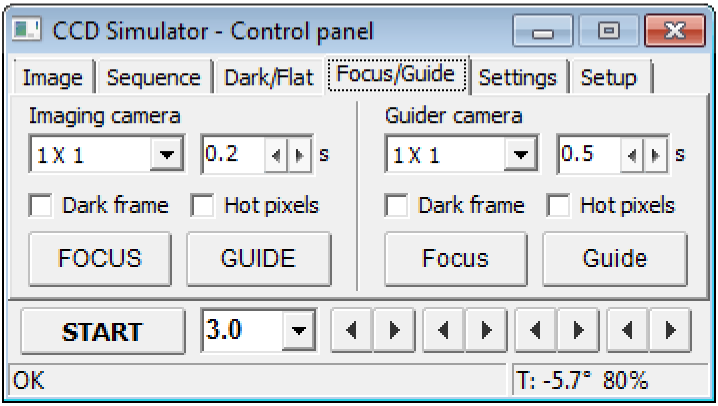
Flat field correction. Select this option to disable or enable the flat field correction for every new image.

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

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

1.6 Focus/Guide page

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



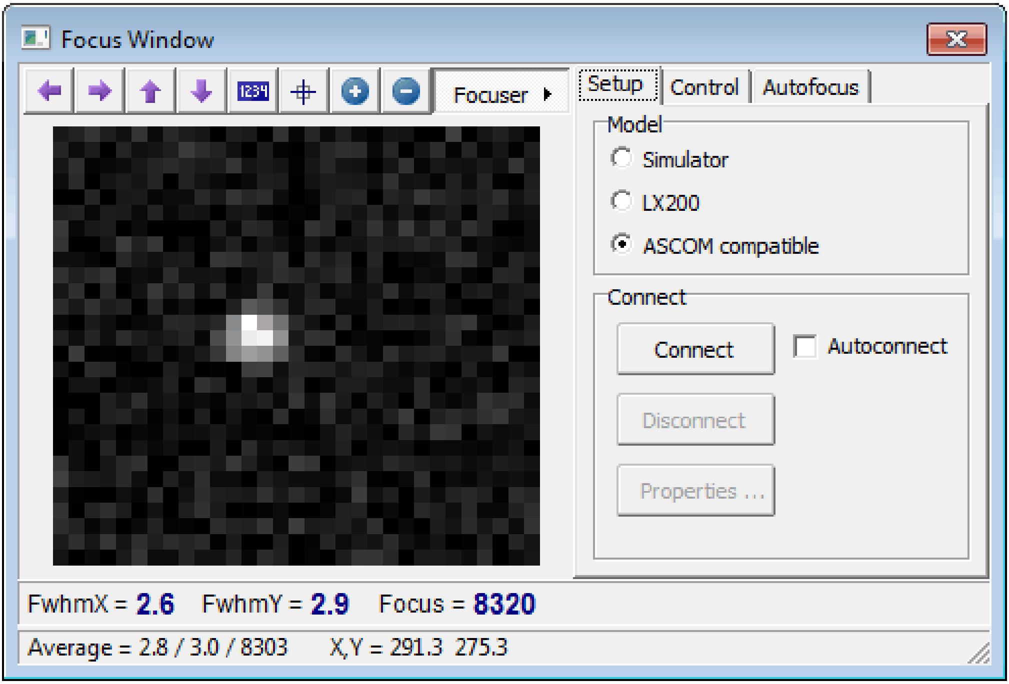
To take a full image with the main CCD camera just click the “Start” button. To take a full image from the guider camera click the “Focus” or the "Guide" button.

Exposure. The exposure time for every frame (both for focusing and guiding). Can be changed in realtime.

Binning. The binning factor of every exposure.

Correct 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.

Average (FwhmX / Fwhmy / Focus). These three are the averaged values from the last few exposures.

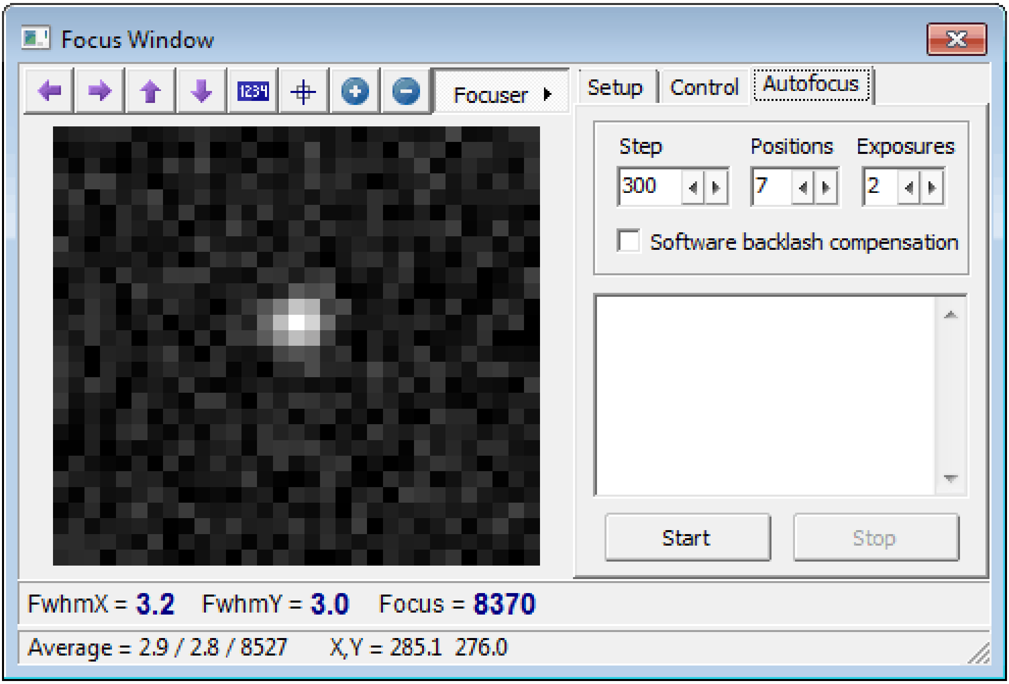
X, Y. The absolute coordinates of the star, inside the full image.

At the top of the *Focus Window* you can find some buttons: The four **arrow** buttons 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.

Autofocus

All ASCOM compatible focusers with digital position are supported.



Step. This value 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.

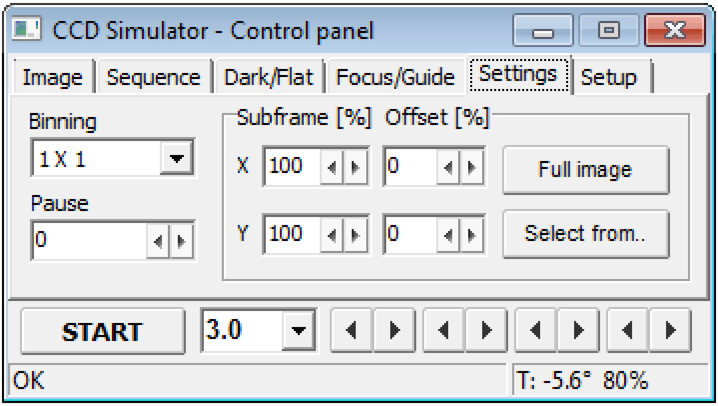
Exposures. How many exposures per position. This is useful when the seeing is not good, for every set of exposures the highest focus will be used. If the seeing is good, choose a value from 2 to 4, if it's bad, 4 to 6.

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.

1.7 Settings page

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

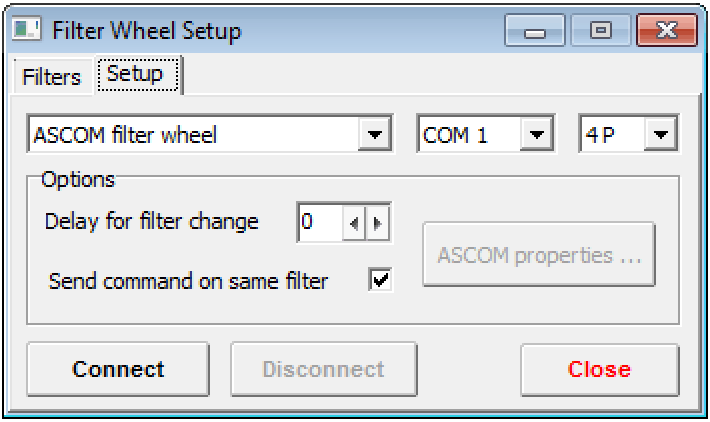
Pause. Before each exposure there will be a pause of N seconds.



Sub-frame. This is 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.

1.8 Filter wheel

To activate the filter wheel select the Setup page, click the "Filter Wheel..." button, select the model, and click on "Connect".



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.

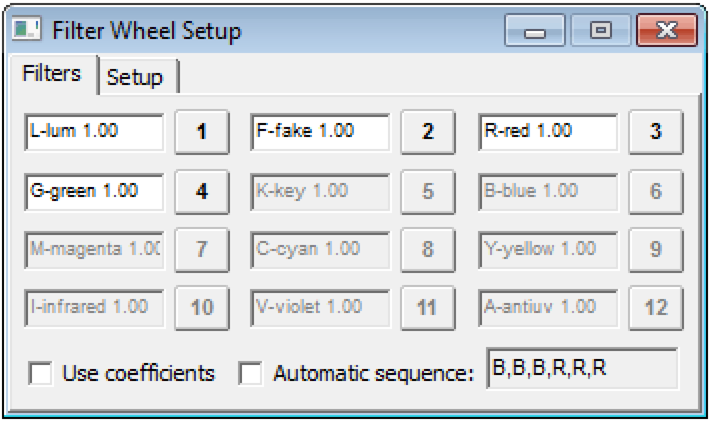
Positions. You may set how many filter positions are available, or just leave the default value for your model.

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.

Sequences

The following options are useful during sequences, by the way they are no longer 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 filtered image will be modified by the given coefficient. This may be useful to compensate the relative sensibility of the CCD with every filter.

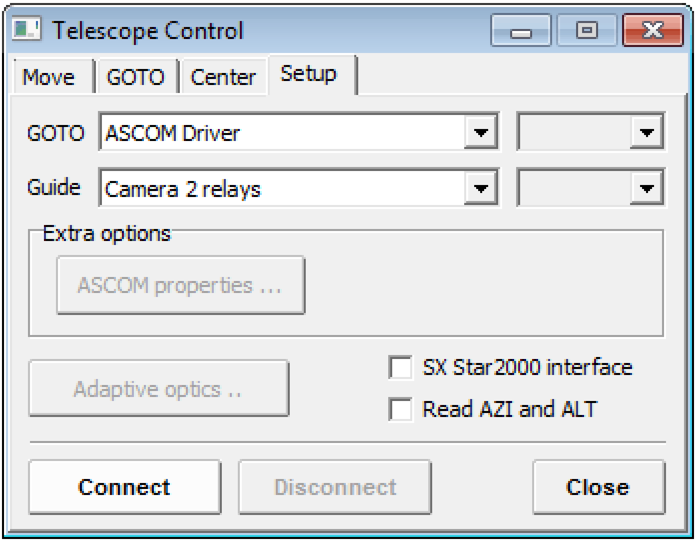
Automatic sequence. If enabled, the filter wheel will be moved before every exposure of an automatic sequence and the first letter of the filter (example: R for Red) will be appended to the filename. Setting for example "R,G,B" these three filter will be used during a sequence.

2 Telescope Control

To configure and control the telescope select the Setup Page and click the "Telescope setup ..." button.

2.1 Setup Page

This page is used to connect the telescope. Select here the protocol used to command the telescope, and the secondary connection (if available) used to send guide pulses.



Options

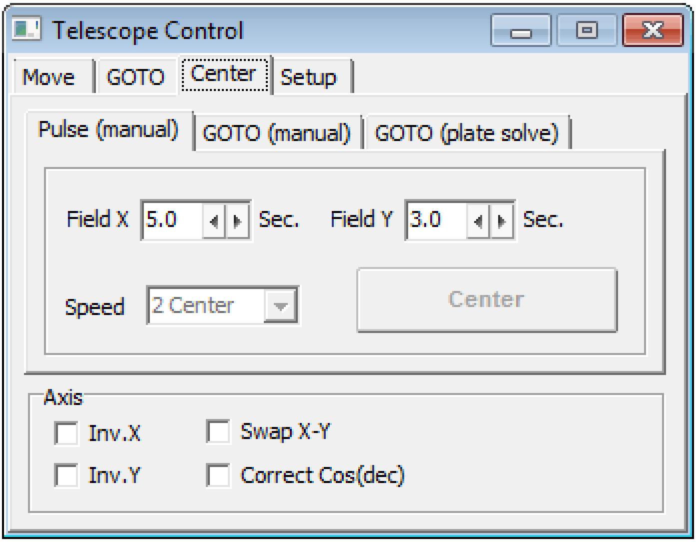
Adaptive Optics. Moves the AO mirror or prism to the middle position.

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

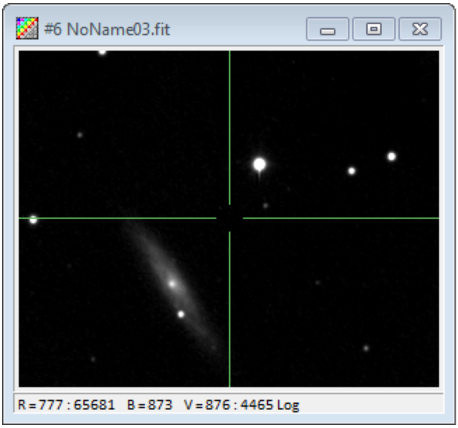
Read ALT and AZI. If enabled, Astroart reads the horizontal coordinates from the telescope. This option could slow down the system on some mounts.

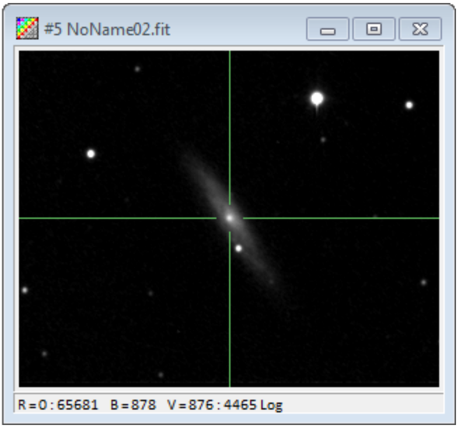
2.2 Center Page

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



Three methods are available to center the object. The first two methods are manual, you will need to select a point over the object then click the [Center] button. The third method is automatic:



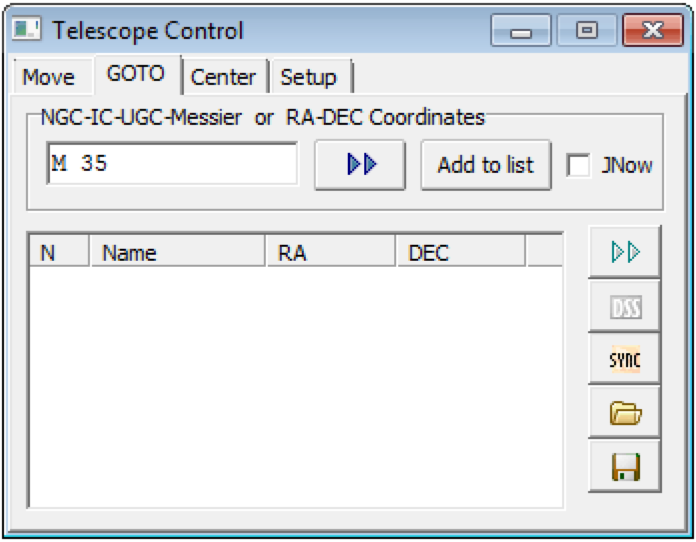


- **Pulse (manual)**. The telescope will be moved using the pulse commands (North, South, East, West) at the given speed. The field of view of the CCD image must be known in seconds (time), this parameter is simply the time necessary for the telescope to move along a full field. The field along the X 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 not so 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. Be sure that the plate parameters in the Find Coordinates window are correctly set.

2.3 Goto Page

This page is used to move the telescope to a given object and to manage lists of objects.



NGC-IC-Messier or RA-DEC Coordinates. 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 on the *GO* button to slew the telescope to that object. The blank space between the catalogue and the object number is mandatory ("N4565" won't be accepted). Enable the **JNow** option to convert the coordinates from J2000 to the current epoch.

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

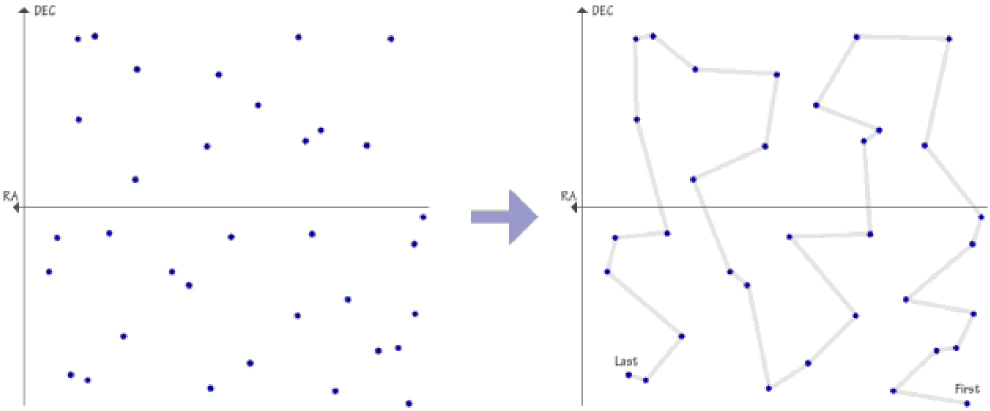
Sync button. Sets the current position of the telescope to the last object coordinates.

Open, Save, Object list. To create a list click the button "Add to list" or just write a text file 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
```

In this example the last two rows are CSV [comma separated values] which can be exported and imported by Microsoft Excel.

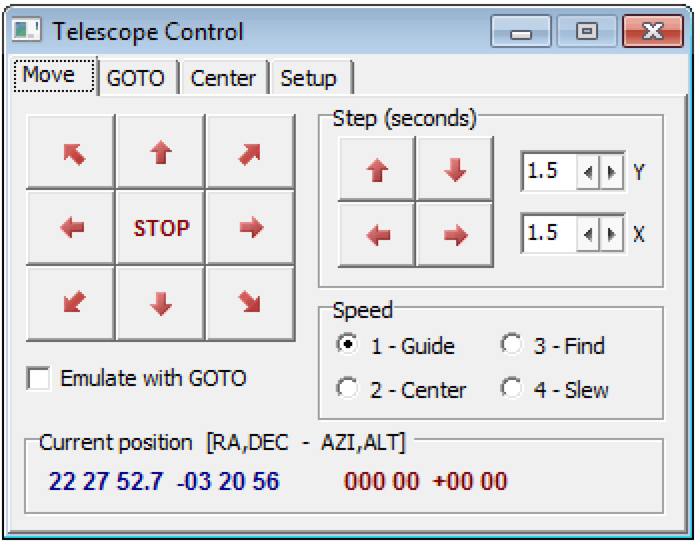
Right click to display a context menu which contains commands to delete rows and to sort the list:



The object list is sorted using a compromise between a shortest path algorithm and a increasing RA algorithm.

2.4 Move Page

This is a "virtual keypad" to move the telescope, useful to center an object. Please note that some ASCOM drivers do not support this function, in this case you may try the option "Emulate with GOTO".



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, usually 1 - 2 degrees per second.

Speed. The telescope speed, this option is not supported by some telescopes. Remember to select “Guide” before autoguiding.

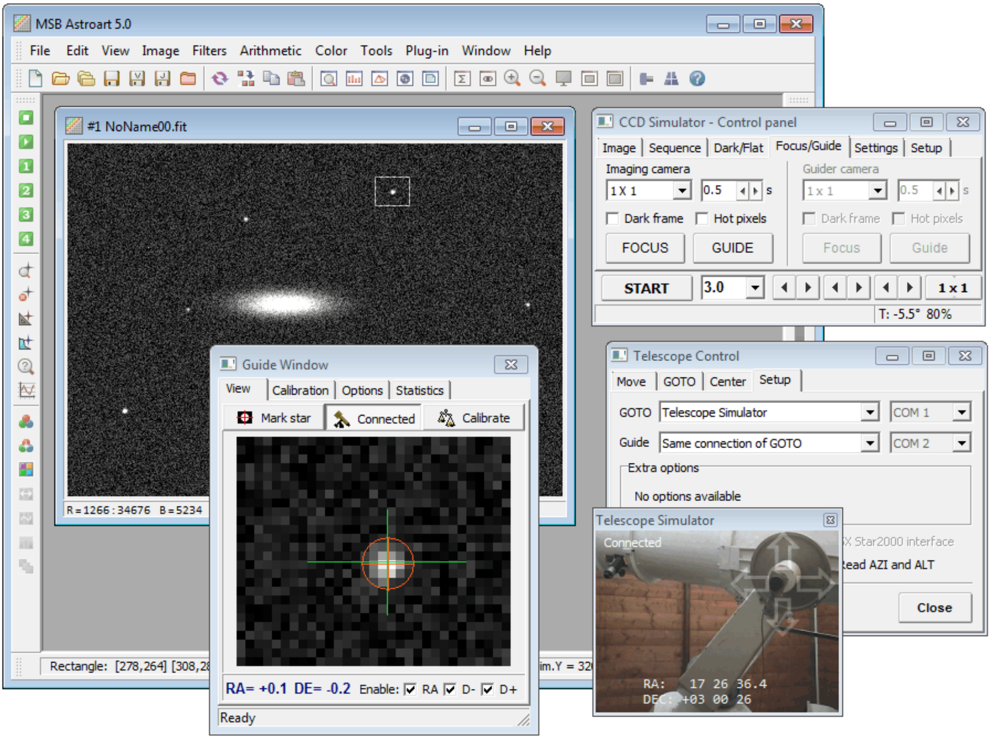
2.5 Guiding overview

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.

Autoguide 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 "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 CCD Interface.
6. Click "Mark guide star" inside the Guide Window. The telescope simulator will now guide on the reference star.
7. To stop the autoguide, click the button "Connect Telescope" in the Guide Window. Verify that the star slowly drifts away. Click again the button to restart the guide.

Let's repeat the procedure using the *Guider Simulator* (the secondary camera). During the guide, we will start a long exposure with the Camera simulator and verify that both virtual 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" (second camera) in 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 from point (6). During the guide, start a long exposure with the main camera.

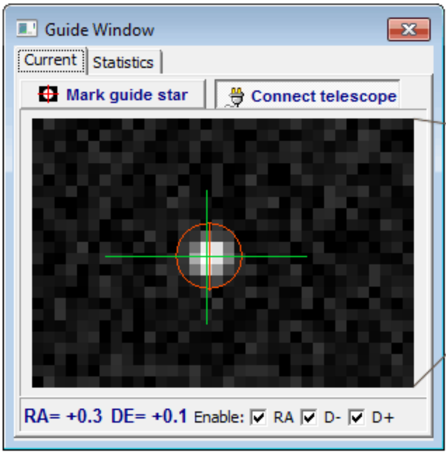
2.6 Autoguiding

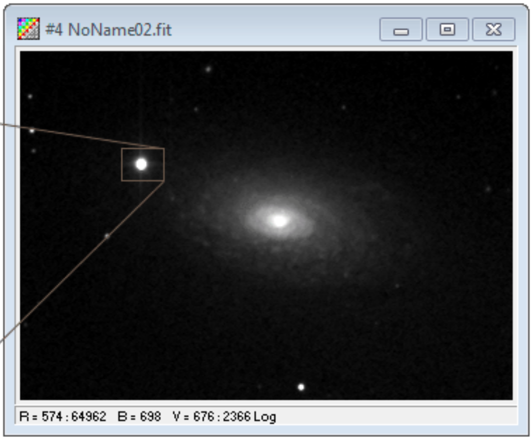
Autoguiding means automatic guiding of a telescope with the aid of a software. Astroart measures the position of a reference star (the guide star) on the tracking CCD, and sends the appropriate commands to both the axes of the telescope mount to correct the tracking errors.

To start a guided session using one CCD camera follow this procedure.

1. Take an image with a short exposure time. The exposure should be sufficiently long to identify a bright (not saturated) 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 guide times and therefore dimmer guide stars.
3. Go to the Focus/Guide Page, select the exposure time (0.2 – 1.5 seconds) 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 recalibrate 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 center. 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 center 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".

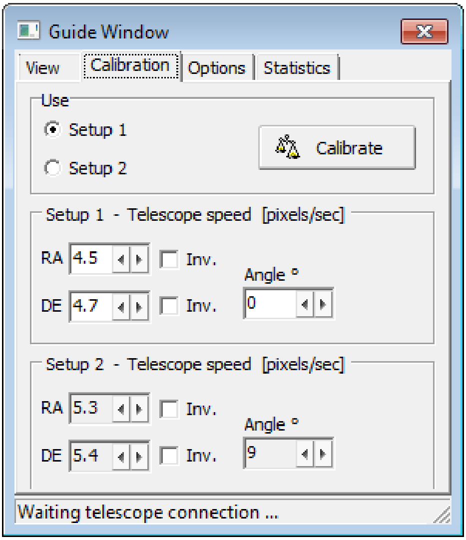
To quickly find the problem it is strongly recommended to try to guide on **one axis at a time**, this will help to understand which problem affects the RA and DEC axes.

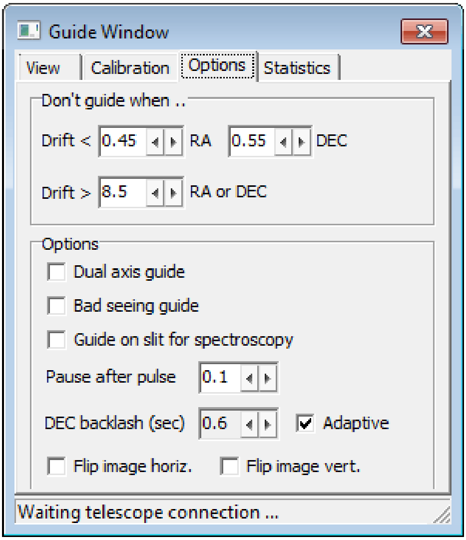
Options

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 need to care about the cos(delta) factor.

- The “Inv.” options must be set if the autoguide frames are flipped horizontally or vertically. You may also compensate this 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 much better to find out all these values with the automatic calibration.

The **Calibrate** button starts the automatic calibration to find out the telescope speed and the angle. It also detects inverted axes. You may save two different setups.





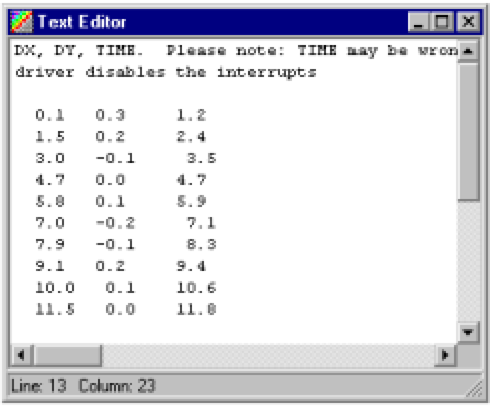
Don't guide when. A useful feature to prevent unwanted corrections for small drifts caused by bad seeing. 0.4 pixel is a good compromise between precision and rejection of noise. Values lower than 0.4 should be used only on a very short focal length (photo lens, for example).

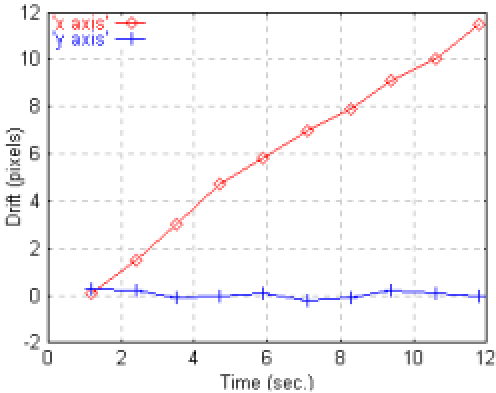
Dual axis guide. If enabled, Astroart drives the telescope with two axes at a time. This feature is important for altazimuthal mounts where both motors need to be controlled at the same time. For equatorial mount this option is not required because the DEC axis changes very slow depending the precision 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

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.



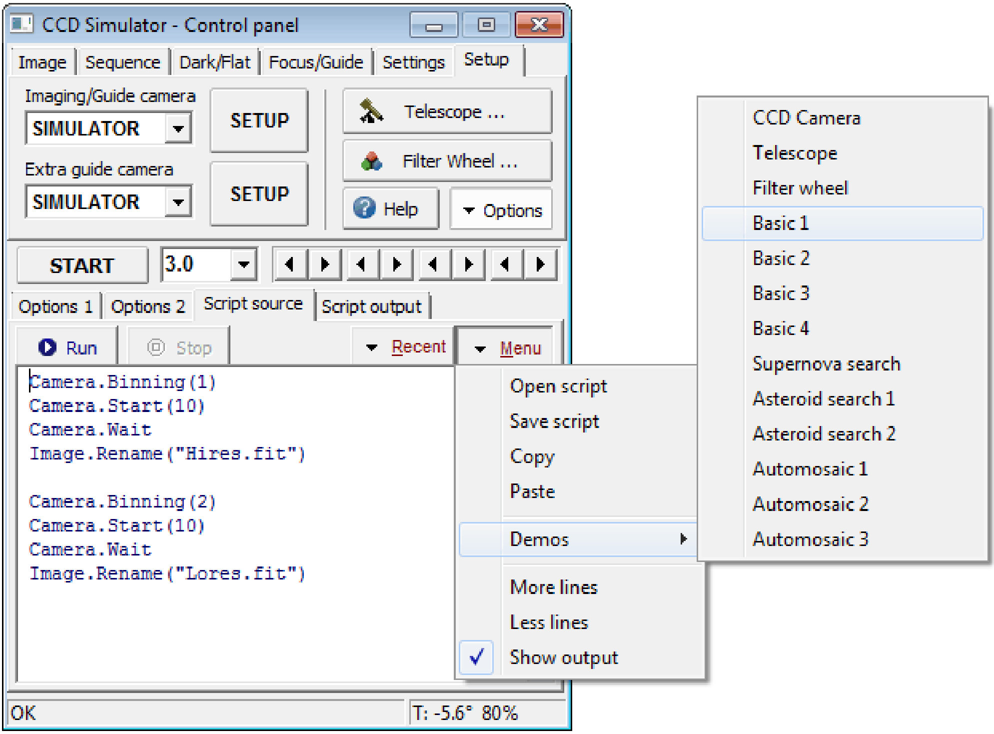


To display the guide statistics select the "Statistics" page of the guide window. Click the "Menu" button for a list of all the commands available. Rotate the mouse wheel to zoom in and out the graphs.

3 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.

The script language of Astroart is based on BASIC, see: en.wikipedia.org/wiki/Basic



Example: (test it with CCD simulator)

```
Camera.Start(10)
Camera.Wait
Image.Save("C:\Temp\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 more complex 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 a list saved in the *Telescope window*. For every galaxy the script moves the telescope, takes the exposure and saves the image.

To quickly learn the script features, try all the demos available in the menu: all simulators will be connected automatically.

3.1 Script Types and Variables

Three types are supported: **numbers** , **strings** and **booleans**. A numeric variable contains a number, a string variable contains a text string, a boolean is "true" or "false".

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 is 64 MByte. You may append the dollar sign ($) to the variable name to remember that it contains a string.

```
a = "Hello"

b = a + "World"
```

The variable **b** now contains the string "HelloWorld".

A single character of a string can be read using square brackets: referring to the previous example **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 single row of a multi-line string can 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 (3 in our example).

Boolean variables

b = 5 > 2 ; b now contains "true", e.g. it's one.

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

Reserved words

The following words are reserved and cannot be used as variables names (see the documentation available for VBScript, Visual Basic 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 SUB CLS`

3.2 Script Functions

See the demo scripts in the menu of the *Script Source* and the documentation available for other BASIC compilers.

Numeric functions

`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

`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() 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.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.SelectDarkFrame **Camera.EnableDarkFrame(**enable**)**	Select the current image as dark frame and enables the correction for the following images.	
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.WaitStar	Stops the script until the guide star is returned to the central position.	
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.	
Camera. StartAutoguide() **Camera. StartAutoguide(**r, relative S/N**)** **Camera. StartAutoguide(**r,x,y**)** **Camera. StopAutoguide()**	Starts an autoguided session. If no coordinates are given then a full image will be taken and the best star will be used. It's possible to set the radius of the subframe and the relative S/N (default is 1.0, increase it to accept only brighter stars). Otherwise it's possible to set the coordinates of the center of the subframe.	**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 star) are not given then this command selects automatically the best star from the current image.	**Focuser.Autofocus** **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)	Moves the focuser to a given position.	**Focuser. GotoAbsolute (1000)**
Focuser.Position()	Returns the current position.	
Image.Save(filename**)** **Image.SaveView(**filename**)**	Saves the current image.	**Image.Save("C:\images\saturn.fit")**
Image.Close	Closes the current image.	
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.DSS(ra,dec,name**)**	Creates a new image from the Digital Sky Survey. Needs the DSS Viewer plugin.	**Image.DSS(12.034,45.213,"asteroid.fit")**
Image.GetKey(key**)**	Reads number values from the FITS keywords.	**a =Image.GetKey("NAXIS")**
Image.SetKey(key, value**)**	Writes values into the FITS keywords.	**Image.SetKey("COMMENT","Bad seeing")** **Image.SetKey("JD",2453709.222792)**
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.	**Image.BlinkAlign**
Image.GetPointX() **Image.GetPointY()**	Returns the coordinate of the selected point (or star, or rectangle) on the current image.	**x = Image.GetPointX()**
Image.Ra ([x,y]**)** **Image.Dec (**[x,y]**)** **Image.DistanceFrom(ra,de)**	Returns the coordinates of the center plate, if the image is astrometrically calibrated. Returns the distance from the given coordinates to the center plate, in degrees.	**ra = Image.Ra()** **de = Image.Dec()** **print "Center plate = ",ra,de**
Image.FindCoordinates(ra,de,stars,[side])	Finds the center plate via plate solving.	See 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,**	Read and write single pixel over the image. After having written	

v) **Image.Update**	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 output panel as a string.	**Output.Save("C:\Log.txt")**
GuideLog.Text	Returns the log file of the guide window as a string.	
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.Broadcast(message, wparam, lparam**)**	Sends a *Windows Message* to all windows. This can be used to control other programs. h = RegisterWindowMessage(message) SendNotifyMessage(HWND_BROADCAST,h, wparam, lparam)	
System.Execute(filename**)**	Starts a program.	**System.Execute("C:\Windows\Notepad.exe myfile.txt")**
Telescope.Connect **Telescope.Disconnect** **Telescope.Connected**	Connects or disconnect the telescope.	
Telescope.Ra **Telescope.Dec**	Returns the current coordinates of the telescope.	**y = Telescope.Dec**
Telescope.Goto(ra, dec**)**	Moves the telescope to the equatorial coordinates ra (0..24),	**Telescope.Goto(23.4, 45.1)**

	dec. (-90..+90)	
`Telescope.Wait`	Waits until the telescope has completed a Goto.	

`Telescope.List.Count`
`Telescope.List.Delete(n)`
`Telescope.List.Clear`
	Returns how many objects are listed in the Telescope Goto page.
	Deletes objects from the list.
		`n = Telescope.List.Count`

`Telescope.List.Name(`index`)`	Returns the name of the <index>th object of the list.	`a = Telescope.List. Name(42)`
`Telescope.List.Ra(`index`)` `Telescope.List.Dec(`index`)`	Return the coordinates of the <index>th object of the list.	`x = Telescope.List.Ra(25)`
`Telescope.List.Open(`file`)`	Opens a text file which contains objects and coordinates. See the Telescope Goto page.	`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.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.Stop`	Stops the telescope.	
`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**	Connects or disconnects the filter wheel.	
Wheel.Filters	Returns the number of filters of the filter wheel.	**n = Wheel.Filters**
Wheel.Goto(n**)** **Wheel.Goto(filter)**	Moves the filter wheel to the given filter.	**Wheel.Goto(4)** **Wheel.Goto("R")**

Input-output functions

PRINT function

Prints a text on the "Script output" panel.

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 ; "years old"
```

MESSAGE function

Shows a dialog window and wait until the user press "OK". Example:

```
MESSAGE("Ready to start")
```

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, so actually they are "functions":

```
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. Another example:

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

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

myLon = 12.5: myLat = +45
ra,de = AltazToEquat(180,+5,myLon,myLat)   'calculates ra and dec at south, for parking.
```

Comments

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

```
' This is a comment
```

3.3 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 is used to exit from a loop. In this 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

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
END WHILE
```

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

3.4 Conditional Instructions

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

```
a > 5 AND exts = ".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]
...
...
END IF
```

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

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
```

3.5 Automatic research script

A script for automatic research may control the telescope, the camera and the filter wheel.

Telescope and camera script

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:

A) The function "**Telescope.List.Open(**filename**)**" which loads a list of objects (see Telescope Goto page). Eventually inside the script it is possible to use the commands "**Telescope.List.Name(index)**", "**Telescope.List.Ra(index)**" and "**Telescope.List.Ra(index)**".

B) The function **OpenText(**filename**)** which loads any text file into a string variable. This multi-line string can be parsed using the functions "**Mid**", "**Val**", etc.

Example (the object list was already opened):

```
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\2015\" + name + ".fit")
  Image.Close
next i
```

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 than 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 Reference chapter.

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 can also be used to in the first example, 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
....
```

Blink and align script

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, then execute the following script:

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

Where mydir is a folder of old images of the same field. This 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:

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

See all the demos (click the "menu" button) for more examples.

4 Glossary

4.1 ADU

A.D.U. or "Analog to Digital Unit" is the unit of measurement of pixel values.

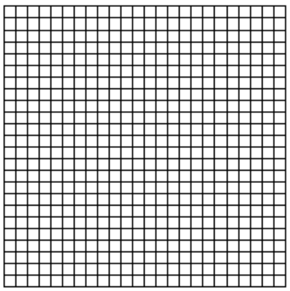
Electrons captured during the exposure are converted to ADU by the ADC chip (analog to digital converter). ADU represents the voltage of the pixel as read from the CCD compared to the maximum voltage of the ADC. This maximum value defines the dynamic range of the ADC (12 bit = 2^12th = 4096 values, 16 bit = 2^16th = 65536 values, etc.).

To read the ADU value of a pixel with Astroart just move the mouse pointer over that pixel and look the *Status bar* of the Astroart desktop. To modify a pixel's ADU see: *Edit Pixels*.

4.2 Binning

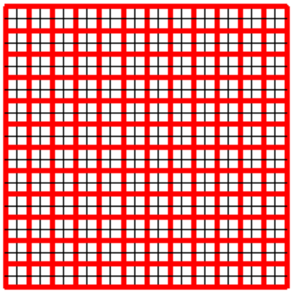
Binning means the union of two or more pixels.

A CCD chip is an array of light detecting regions called pixels (for picture elements). A CCD picture can be acquired joining some adjacent pixels and making them one effective superpixel. The figures below represent a detector of 144 pixels, the thick lines indicate the current binning mode.



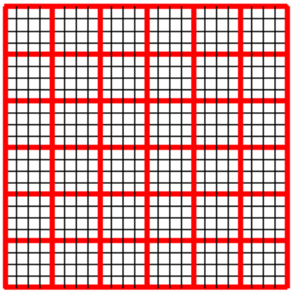
Binning 1x1

Number of pixels: 576



Binning 2x2

Number of pixels: 144



Binning 4x4

Number of pixels: 36

The advantage of binning is the reduction in noise, so the Signal to Noise ratio increases.

Imagine a 2x2 array of pixels. Each pixel has signal and noise. If you add the values of the four pixels the new S/N doubles, because it becomes 4*Signal / sqrt (4*Noise). Noise adds as the square root because it is a random process.

The drawback of binning is the loss of resolution. Smaller pixels detect more portions of an object. However with sufficiently small pixels binning has not a large impact since it's generally the sky (seeing) that limits the resolution.

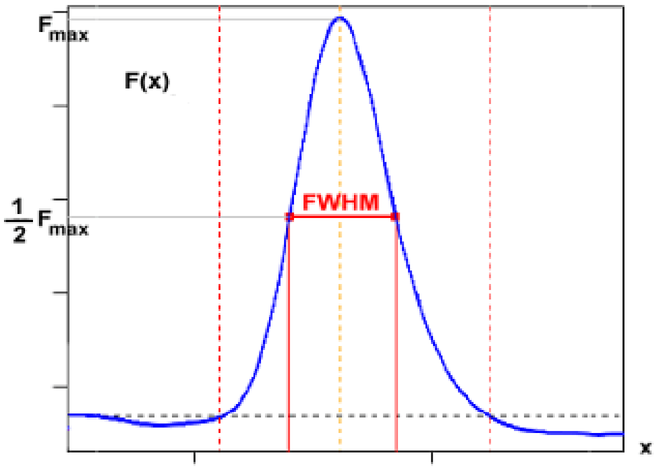
4.3 CCD

CCD chips consist of an array of light-sensitive capacitors. When light falls on the chip during the exposure, electrons released by the semiconductor are collected by these capacitors. The number of electrons is proportional to the amount of light.

At the end of the exposure the charge of each capacitor is measured, the resulting electric signal is converted by a component called ADC (analog-to-digital converter) into a number which is transmitted to the PC. This number is called "ADU", the value of the pixel.

4.4 FWHM

FWHM or *Full Width at Half Maximum* is a measure of star size. To calculate it from a profile of a star simply measure the distance between the two points where the signal is 50% of the peak value.



The FWHM can be measured in pixels or arcseconds. Inside the Stars window the FWHM is measured in pixels.

4.5 Gain

The gain of a CCD camera, or the system gain, express how many electrons are represented by one ADU. A gain of 2.0 electrons/ADU means that each ADU represents 2 electrons. This implies that the total well depth of a Kodak KAF-0400 CCD (85,000 electrons) could be represented by 85000 / 2.0 = 42500 counts.

As long as the total well depth of a sensor can be represented, a lower gain is better to minimize the noise and give better resolution. Gains which are unnecessarily high can result in more digitization noise, while gains which are too low will minimize noise at the expense of well depth. For example, a 16 bit CCD camera with a gain of 1.0 would allow only 65,536/1.0 = 65,536 electrons of the 85,000 to be digitized. System gains are designed as a balance between digitization counts, digitization noise, and total well depth.

4.6 LAA

Astroart LAA is a special version of Astroart compatible with the Microsoft "LARGE ADDRESS AWARE" technology on Windows 64 bit.

Astroart LAA can therefore use 4 Gb of RAM instead of 2 Gb. This version is kept separate because some older camera and telescope drivers are not compatible with it. In this case, you may use this version for image processing only.

4.7 Point Spread Function

The Point Spread Function (PSF) is a function which defines how a point source would appear if imaged with the instrument.

Ideally a point source would produce a clear signal in just one pixel on the image but due to the optical effects this is not the case. Optical diffraction and atmospheric distortion are the factors causing light from point sources to be smeared into many pixels. This means that the PSF defines the resolution of the instrument.

4.8 Outside constant

This is a constant used when considering points "outside the image".

In the Preferences window it's possible to set a value for it (example: zero) or to force to make it variable and equal to the sky background.This constant is used by the commands: Rotate, Shift, Align, other filters calculates a more precise value computed near every edge of the image.

4.9 Readout Noise

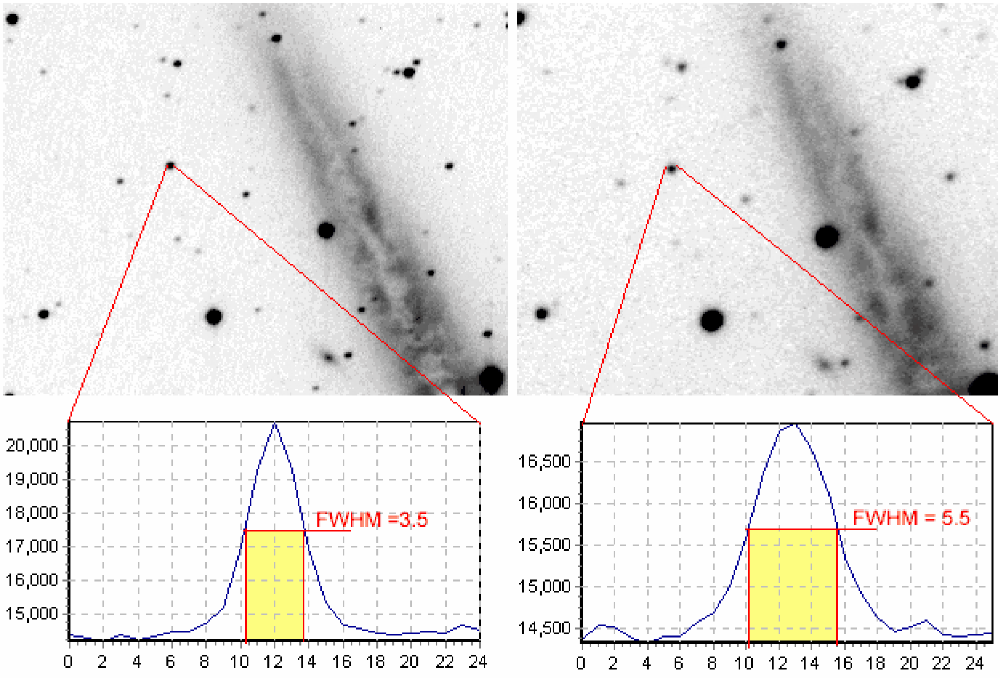
Each time the charge is dumped out of the CCD there is an uncertainty in the measure which is called readout noise and expressed in electrons (e-).

Readout noise is specified both for the CCD sensor and the total system. For example, a good noise figure for the Kodak KAF-0402E and KAF-1602E is 13e- typical, while a good number for the entire system is 15e- typical. Most applications are not limited by readout noise, since the noise caused by the sky background and dark current is much higher.

4.10 Seeing

Without an atmosphere, a small star would appear as a diffraction figure determined by the optical system, and would be inversely proportional in size to the diameter of the telescope. However when light enters the Earth's atmosphere, the different temperature layers and different wind speeds distort the light and the image of the star.

This distortion changes at a high rate, usually more than 100 times a second. In a typical astronomical image with an exposure of several seconds, the distortion blurs the star into a disc called the point spread function or "seeing disc". The diameter of the seeing disc is the *Full Width at Half Maximum* (FWHM) which is a common measure of the seeing conditions.



On the left, an image with a seeing of 2.9 arcseconds. On the right, the same image on a worse night, with a seeing of 4.6 arcseconds. Courtesy Cavezzo Observatory.

The FWHM of the seeing disc (or just Seeing) is usually measured in arcseconds, abbreviated with the symbol ("). A 1.0" seeing is a good one for average astronomical sites. The seeing of an urban environment is usually much worse. At the best high-altitude mountaintop observatories the air is stable, since it's not in contact with the ground during the day, so sometimes the seeing is as good as 0.4".

4.11 Selected point

Every image can have one more selected points, a cross marks their position.

To select a point simply click on it with the left button. To select a point over a star press [CTRL] while clicking. You may also select a point in the *Edit Pixel* window pressing [SPACE], or launching the command *Edit* / *Select*.

To delete a point click on it. To delete all points draw a rectangle. To restore all points or copy them from another image, click the left mouse button keeping [SHIFT] pressed.

4.12 Selected rectangle

An image can have a selected rectangular region. To select a region press the left mouse button and draw a rectangle, or use the command *Edit* / *Select*. This rectangle is not a mask for filters, when a filter is applied and the image has a rectangle selected, the filter is always applied to the whole image.

4.13 Signal to noise ratio

Signal-to-noise ratio (SNR) is the ratio between the signal and the noise of an image (or a part of it). The higher the SNR, the better the image. Noise in CCD images is caused by sky background, dark current, electronic sources and statistical randomness of the signal. For a correct measure of SNR of stars, you must set the camera parameters (gain, etc.) in the Preferences.

What is the noise with a ideal CCD camera (i.e. with zero noise) on a absolutely dark sky (again, zero noise)? The result is not zero, it's SquareRoot(N), where N is the number of photons collected.

This result is surprising, there is still some noise as predicted by the Poisson theory. This noise is caused by *randomness*. For example, with 100 photons the noise is 10, so measuring the star you may get 92, 104, 88, 107 and so on. To understand that, remember that photons are emitted with a random trajectory and a random emission rate.

So, even with with a perfect CCD under a black sky we still need a long exposure, to be able to collect thousand of samples. For example, with 50000 photons the noise is 223, so we may expect an error of about +- 223/50000 = 0.4 % which may be acceptable. The error itself follows a gaussian distribution, so there may cases where the error is much higher than 0.4 %.

Please notice that the unit in this example is photons, while in CCD images the unit is ADU. If your camera produces 1 ADU every 4 photons then a star measured 25 ADU was sampled by 100 photons. The Poisson noise is 10 photons, so 2.5 ADU.