Dragonfly

Users manual

I n d e x

1. Introduction

Setting up the remote control system of your observatory requires quite a bit of effort – our Dragonfly has been designed to help, making it easier.

The Dragonfly is a very powerful and versatile device. Every effort has been made to make it simple to use, yet allowing for full customization. Setting it up to remotely control our observatory can be accomplished with little effort, yet you can go as deep as you want and have it performing incredibly complex tasks if needed.

The only effort *required* in all cases is to understand the way relays are used (if you are not yet familiar with relays). We'll learn step by step to setup and use the Dragonfly with a working example.

So, first of all, a bit of background.

A relay is a very useful device indeed – being an electrically operated switch, it enables appliances such us our Dragonfly to turn on and off other appliances.

Let's compare relays and push buttons.

Every relay has two sides: the control side (your finger able to push or release the button) and the switch properly said (the internals of the pushbutton opening or closing the circuit).

The most common push buttons leave the circuit open if nobody is pushing them, and close the circuit when pushed (there also exist push buttons working the other way around). These most common push buttons are called "normally open": they are open until some effort is made to change the state. The other model, yes, is called "normally closed".

The same applies to simple relays: they can be "normally open" or "normally closed". When a really is unpowered it is in its **normal** state. When it is powered (power is applied to the control side, that is, like a finger push), the state changes. If we name the contacts of the relay IN and OUT, when the relay is closed, current will flow between IN and OUT, because internally IN and OUT will be connected. Conversely, if it's open, no current will flow.

So a **normally open** relay **won't** allow current flow unless powered.

And a **normally closed** relay **will** allow current flow unless powered.

So far so good.

But there are relays a bit more complex (and useful!). They have 3 contacts – let's think of them as IN, NO (for normally open) and NC (for normally closed). The unpowered or relaxed relay will have IN connected to NC - with the ability to change this connection, and short IN with NO when powered (breaking of course the IN to NC connection).

| 3 contact relay unpowered | 3 contact relay powered |
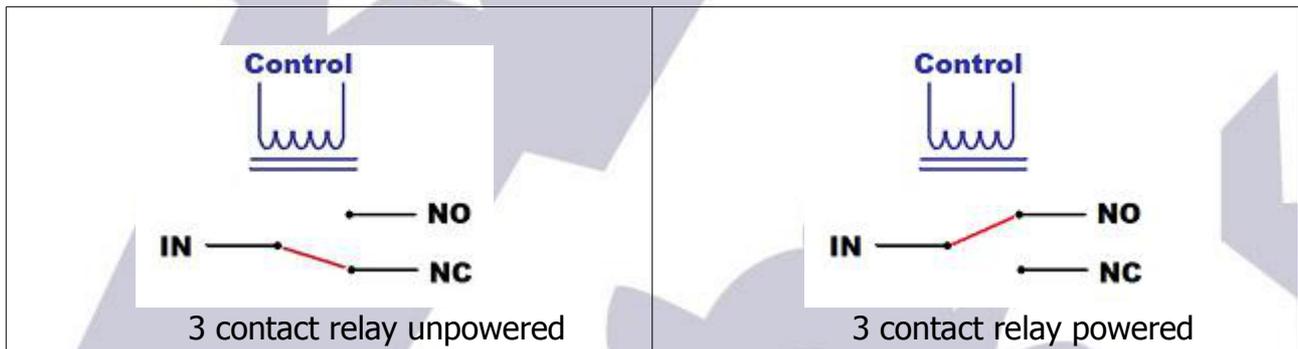
For added flexibility, we use both kinds of relays in the Dragonfly.

2. Planning

Decide exactly what you want to be able to control:

1. opening / closing the roof
2. powering the mount
3. powering the CCD Camera(s)
4. forcing off the observatory lights

and monitor:

1. position of the mount (at home or not) if needed to operate the roof
2. position of the roof
3. power (mains availability)

... remotely. The above list can be considered a typical setup. Of course it is not the same to setup your backyard observatory so it can be controlled from the living room, than setting up your truly remote observatory 300 Km away from home.

All the following examples will be aimed at setting up an observatory with this requirements; please do use the same relay / sensor numbers as this will greatly simplify the process.

3. Electrical setup

The setup of the Dragonfly is quite straightforward – it may involve, however, dealing with dangerous voltages and currents (depending on your setup); **please contact a qualified electrician if you don't have the skills needed to perform a 100% safe setup.**

> When naming the state of any relay, we'll think on the "NO" output. This applies to the software, too. So we'll say the relay is OPEN if it's NO output is. All relays will be open when unpowered. Conversely, we'll say the relay is CLOSED if its NO output is connected to the input side.

If you take a look at the sticker covering the upper side of your Dragonfly:

... you'll note we've used this naming convention for labeling the input and output connectors.

Second thing to notice: *Power Input* connectors are at the lower side, *Power Output* ones

at the upper side. Current, be it AC mains (max 240V) or DC from a battery or power supply, will enter the Dragonfly via the Power Input and will be blocked or routed to the Output side. At the Output side we'll have our devices, mount, roof, CCD Camera or whatever.

So, to make sure it's clear how the internal wiring goes:

- ❤ "COM" will be internally routed from the input to the output side - *always*
- ❤ "NO" (Normally Open) will be connected to "In" when the relay is closed by the software
- ❤ "NC" (Normally Closed, only available in relays 1 to 4), will be connected to "In" when the relay is open in the software

When the Dragonfly is unpowered, all the relays will be in the relaxed (normal) state, so:

- ❤ All "NC" contacts, present in relays 1 to 4, will be connected to their matching "In" (so they'll be closed)
- ❤ All "NO" contacts, will be, of course, just open

Graphically:

Dragonfly unpowered - or - Powered and relays not closed

Dragonfly powered and relays closed
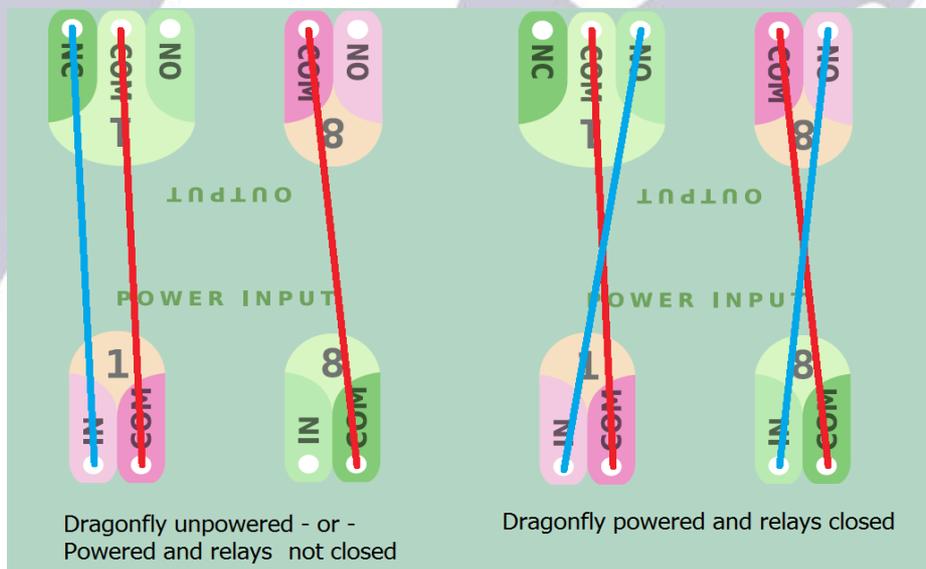
So what are the "NC" contacts for? When designing the Dragonfly, we noticed that certain things should be powered at all times except during the imaging session. One simple example is the observatory lights – you may choose to route the hand switch through the Dragonfly in such a way that no one can accidentally turn on the lights when you are imaging. You can even use the "NO" contact of the same relay to power on some other thing when the lights go off.

In my case, I have an electrical dehumidifier and I want it off for sure before the roof is opened!

So let's setup things for our minimalist observatory – adding the dehumidifier:

- **Relay 1** → Roof control (pulsed – that is, a pushbutton, as standard in many garage door automation motors)
- **Relay 2** → Mount
- **Relay 3** → CCD Camera
- **Relay 4** → Observatory lights

Bear in mind! - at the input side, for each relay:

✔ neutral (AC) or ground (DC) should be wired to "COM" (common)
✔ phase or live (AC) or positive (DC) should be wired to "In"

So the wiring, input side:

✔ the same for relays 2, 3, and 4: mains neutral to **COM**, mains phase to **IN**
✔ for relay 1, roof, as we want to "push a pushbutton", we'll wire **COM** and **IN** together (just a short wire connecting them both). This case is different as we do not need to power the roof motor, but just to push a button, something like this:
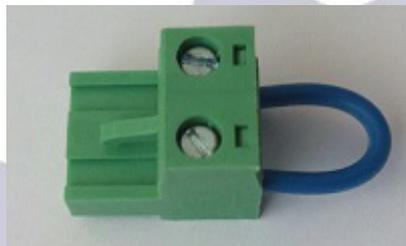
… and output side:

✔ Relay 1: **COM** and **NO** to the roof motor pushbutton
✔ Relay 2: **COM** and **NO** to the mount power supply.
✔ Relay 3: **COM** and **NO** to the CCD Camera power supply.
✔ Relay 4: **COM** and **NC** to the observatory lights. NOTE in this case it is NC, not NO, as we want the lights working in normal conditions.

Note: the box of the unit must be connected to ground (earth).

The LED for each relay will be **on** when the relay is **closed**.

We can't forget to mention earthing; the Dragonfly incorporates 8 banana-style plugs at each side to make earthing really simple. They are all connected together and to the metallic box.

The rule is simple; all A.C. current appliances should be earthed. Also, even if all current routed by the Dragonfly is D.C., the box should be earthed; this is accomplished by wiring any of the 16 plugs to earth.

Regarding the sensors, they are connected to the "sensor in" section; most of them (proximity, contact, etc) will just need the "+5v" and "In" connections. For other, more complex sensors (as our Sharp distance one), ground is also needed; there are 4 ground plugs between the "sensor in" and "power in" strips.

There's a separate instruction sheet on wiring things to the Dragonfly, should you need it please check our website.
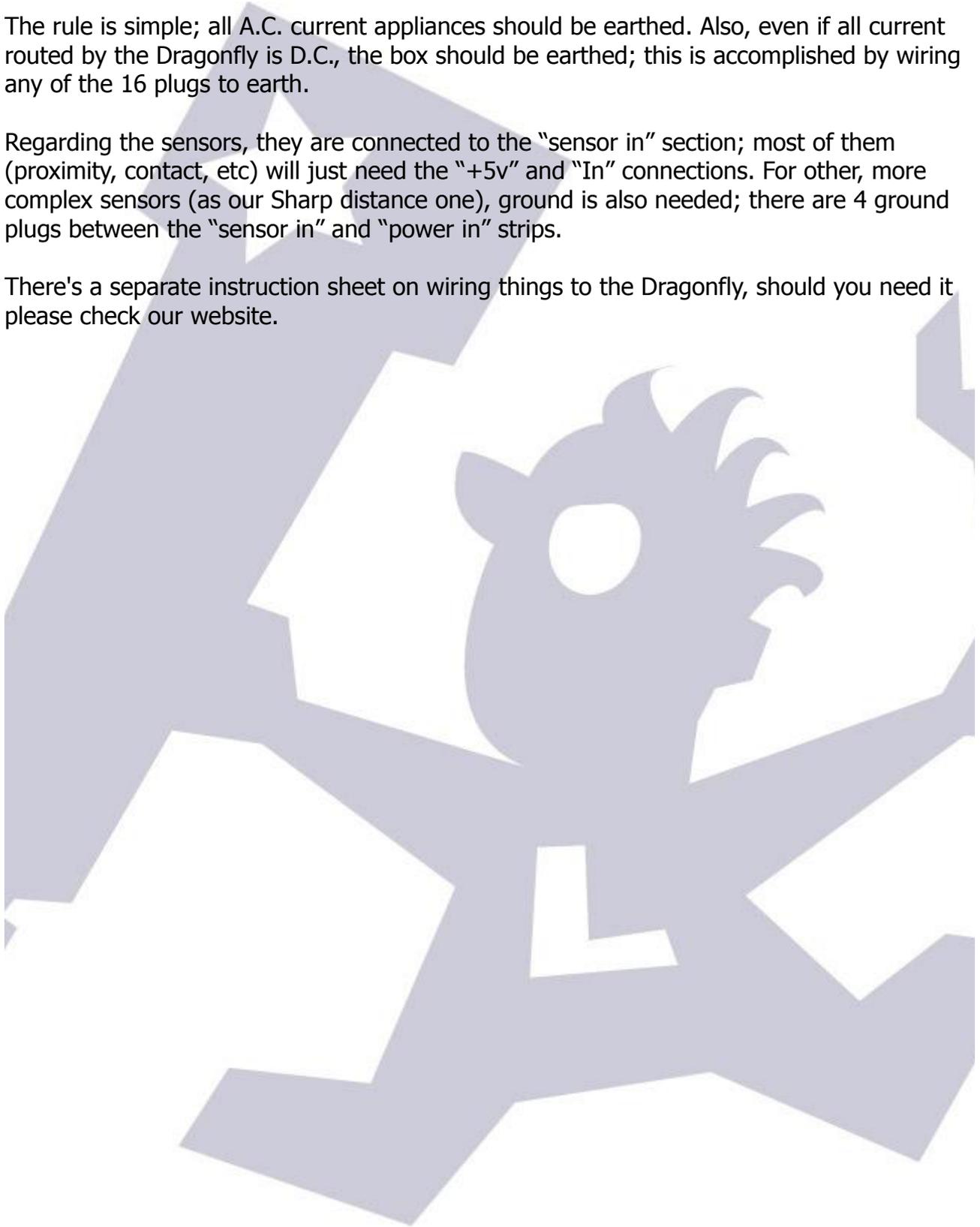
4. Software

Let's look at the supplied software step by step. We strongly suggest to first make sure everything works as intended and can be operated safely from remote, but with you supervising and actively switching things on, off, and looking at the sensors. Only when you are comfortable with this should automation be addressed.

4-a) Installation

Installing the software is as easy as it gets. Only prerequisite is to have the ASCOM platform installed. You can download it freely from:

www.ascom-standards.org

Once ready, download the Dragonfly software from our web page (there's a software download button at the bottom of the page):

Dragonfly's web page

run the installer and follow the simple instructions.

4-b) Connecting

If after installing you launch the software (Dragonfly, from the start menu or apps), two windows will appear, one for controlling the relays and checking the sensors, and a smaller one for the configuration of the network and performing future firmware upgrades:
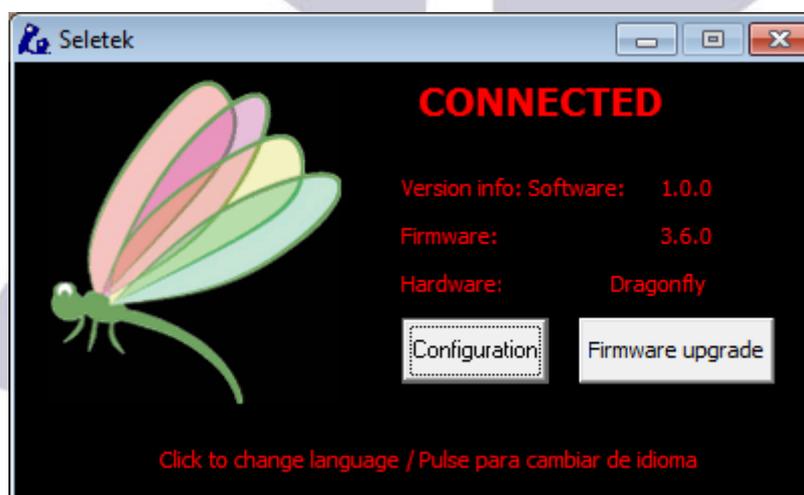
we won't be using this one for the moment, so just please minimize (do not close!) it. It will remain open, but hidden, in the lower right windows notification area.

Just in case you have a problem connecting to your controller, check section 6.

4-c) Remote control

The remote control software supplied with your Dragonfly allows you to name every physical connection so you don't have to guess. It is really easy to understand and we'll review it working for our sample observatory.

We are going to spend some time with this Dragonfly control panel, also visible:
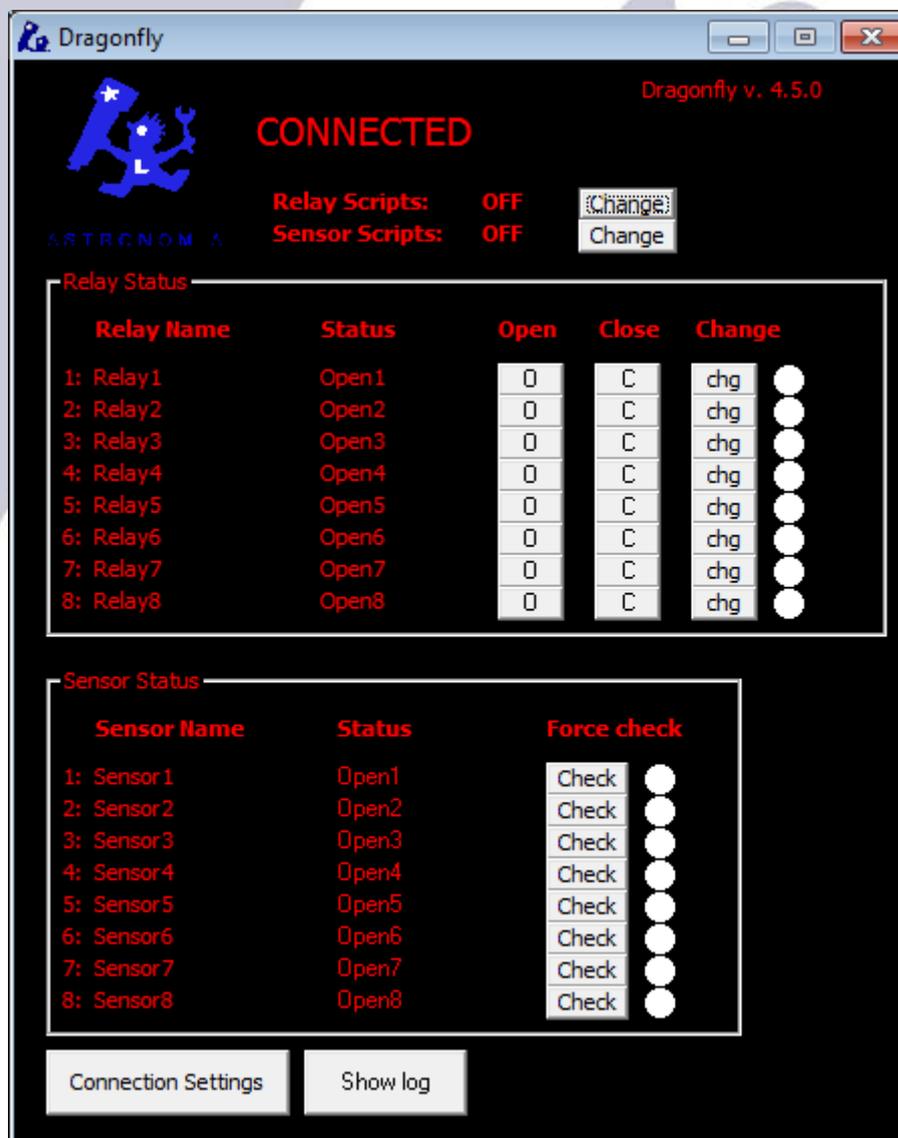
We'll focus in the two big areas, upper one for relays, lower for sensors. It's easy to see there are 8 rows or relays and 8 of sensors, just as in the box.

Let's give a name to each relay, and even to each state – so relay 1 becomes "Roof motor", relay 2 "Mount power", with closed becoming "On" and opened "Off", etc.

Just click over the current name of the relay (where the mouse cursor is in the followinge image):
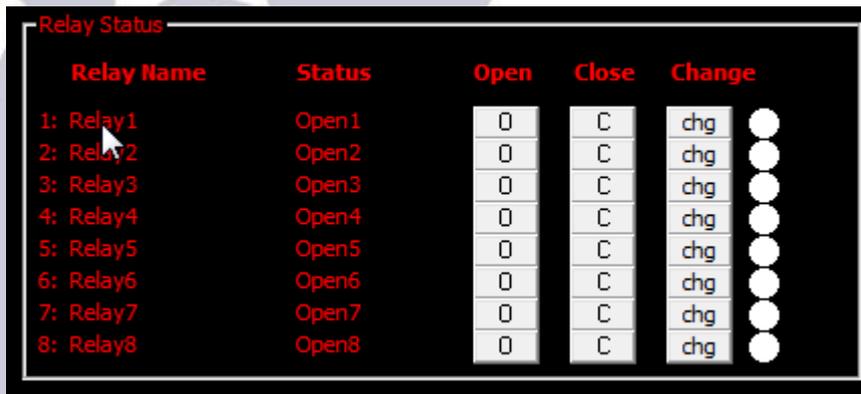
… and you'll be presented with the settings for that relay:

You can name the relay and what should appear when it's closed and open. You can also define it to be a "pulsed" relay, as appropriate in the roof motor configuration.The pulse period, you surely guessed it, is the approximate duration of the pulse in millisecods.

Keep naming relays:

Settings for relay 2

Relay name	If open display	If closed display		Pulse period (ms):
Mount power	OFF	ON	☐ Pulse?	2000

OK   Cancel   ☐ Confirm on manual change?
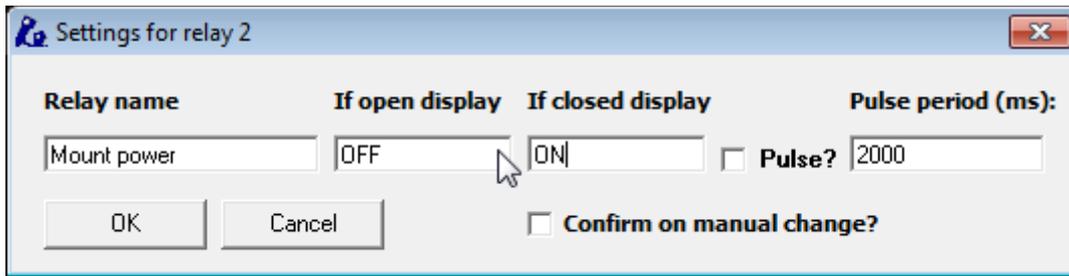
… please continue to do the same for the CCD power and observatory lights – don't forget the lights will be forced off if the relay is closed!

Settings for relay 4

| Relay name	If open display	If closed display		Pulse period (ms):
Observatory lights	ON	Forced off	☐ Pulse?	2000

OK   Cancel   ☐ Confirm on manual change?
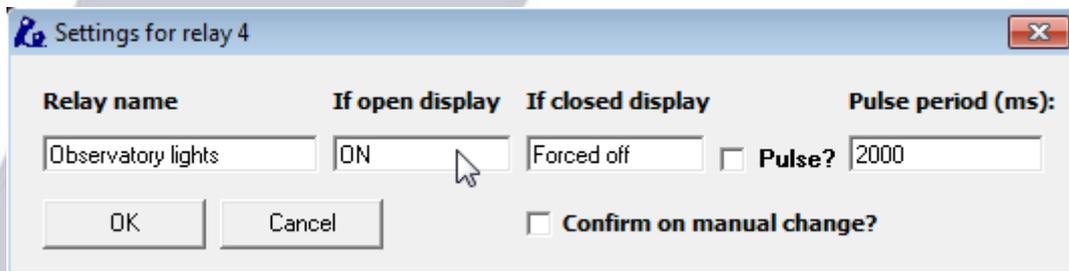
Last, and to avoid accidental clicks, get back to the roof motor relay settings; you can protect any relay selecting...

Settings for relay 1

Relay name	If open display	If closed display		Pulse period (ms):
Roof motor	---	Pulsed	☑ Pulse?	2000

OK   Cancel   ☑ Confirm on manual change?
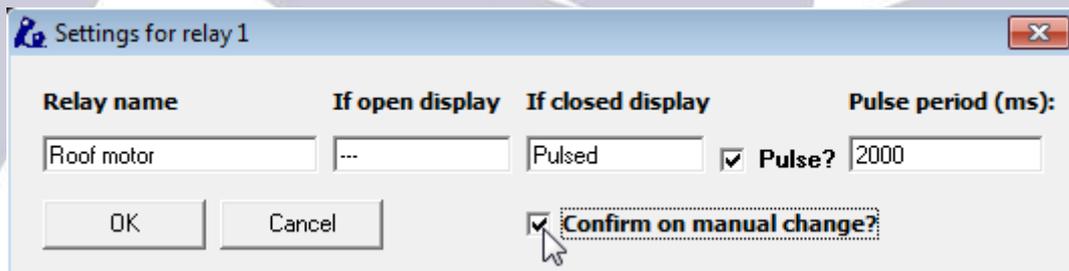
… the "confirm on manual change" option. This way you'll be asked for confirmation if you click any of the buttons concerning that relay.

Now the relay panel should look like this:

Notice both the "P" and the "!" at the right side, roof relay; the "P" means it is pulsed, the "!" means you'll be asked for confirmation before any action takes place.

The same naming system applies to sensors:

for the roof (roll off roof) we suggest using two switches, one for signaling "open" and other to signal "closed", this way we will notice should the roof stop mid-travel.

So before an imaging session, with our observatory roof closed our window will look like this:
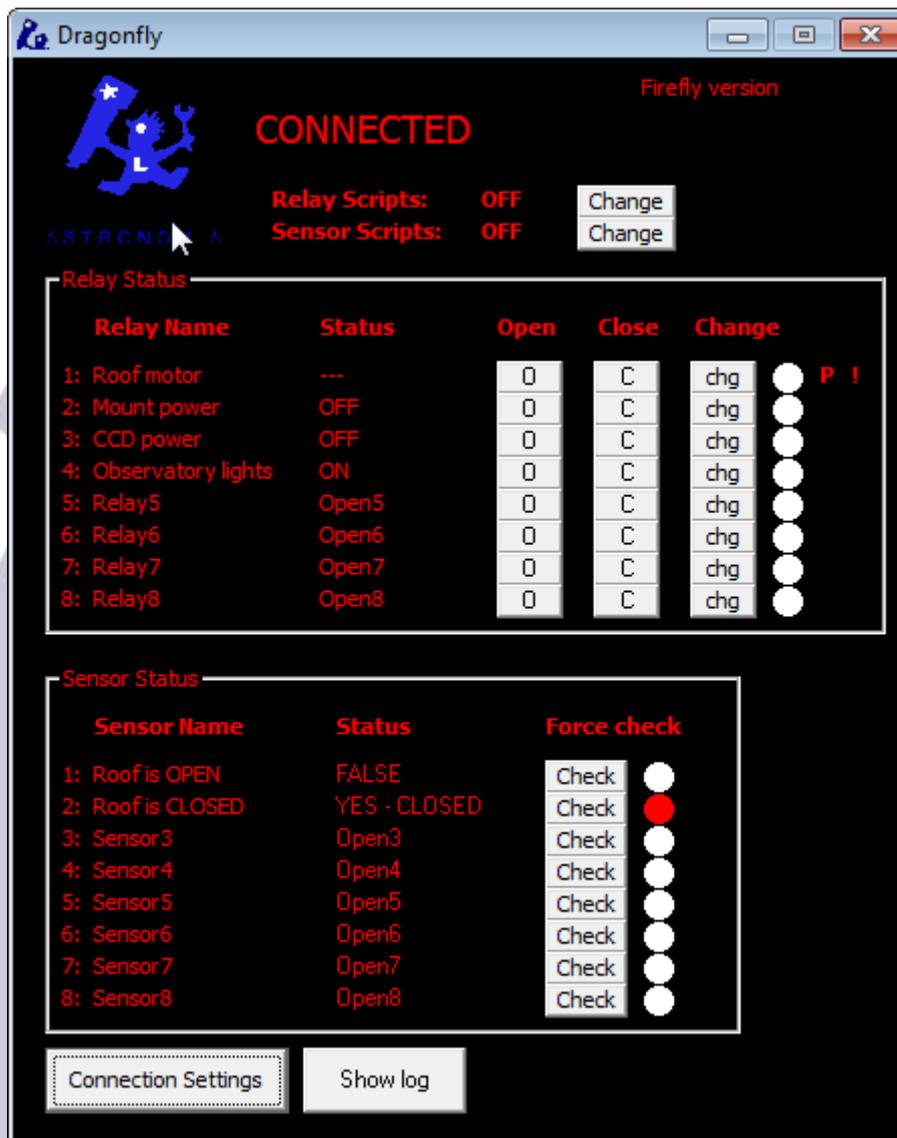
Lights enabled, Roof not opened – but closed, ccd and mount off. We are deliberately leaving the telescope park sensor for the moment. Great.

Switch off the lights (click "Close" in its row) and click "change" (chg) at the Roof motor row (NOTE: *the pulse we defined earlier will only work when **change** is clicked*).

As we also selected the "Confirm on manual change", we'll get a confirmation window:
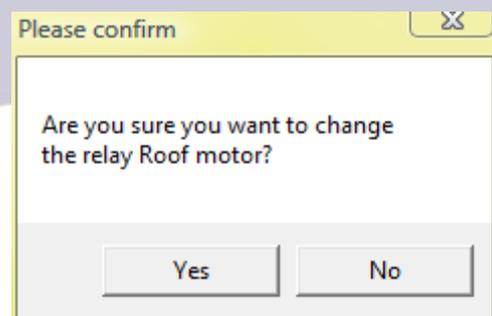
... select yes to proceed. You can now also power the mount and CCD Camera.

The roof will open, and, once fully opened, we'll have this:
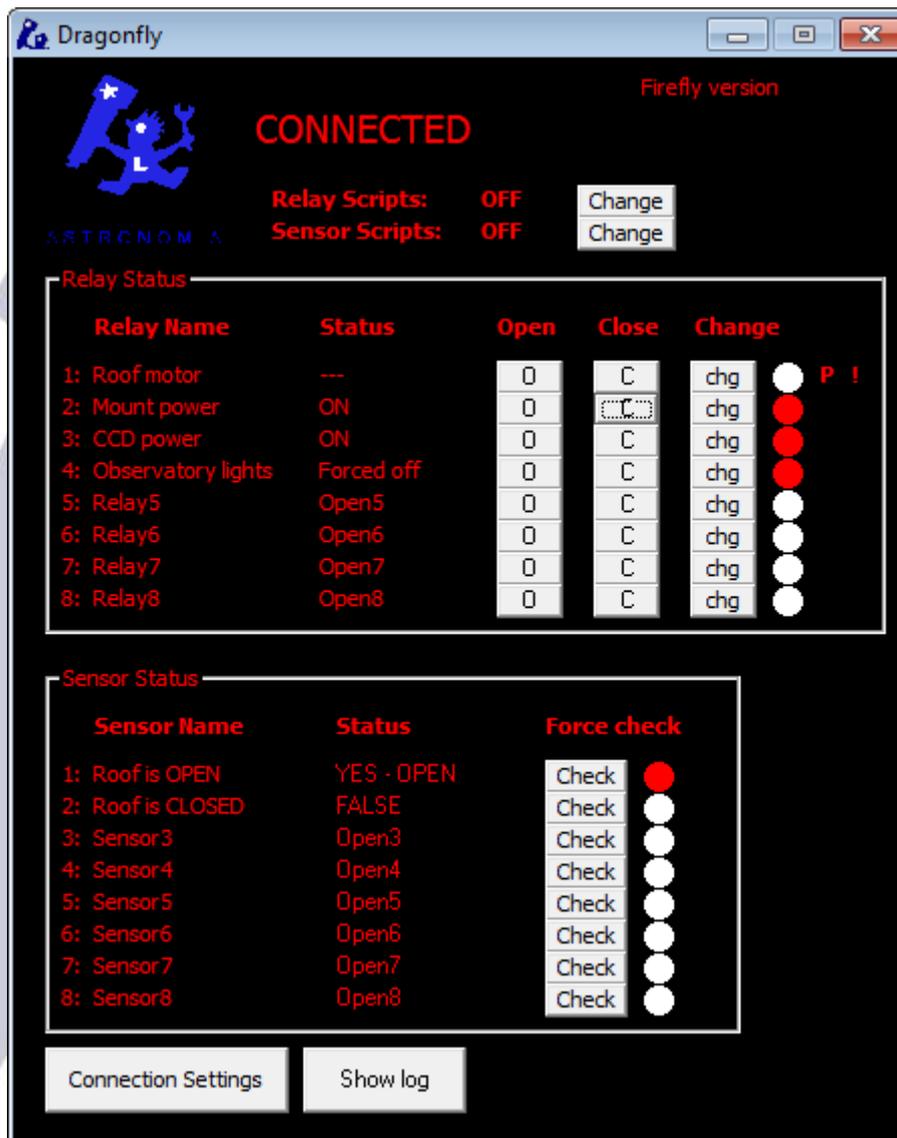
… as you can see, it reflects the current state of things.

Last thing worth mentioning is the connection settings:

The sensors (and relays) will be checked, one at a time, every 100ms (or any other frequency, between 10 and 60000 ms). You can always force the immediate check of a sensor clicking on the appropriate button "Check".
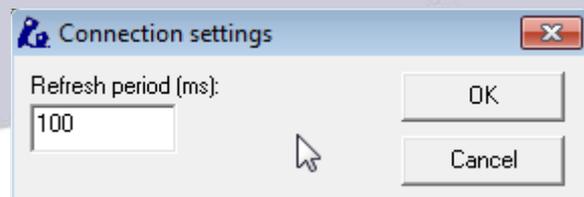
For local area networks, 100ms or even faster is ok. If you will be connecting from far away, via internet, using a higher value (1000ms or so) may be better.

Addenda: analog sensors

Some sensors, such our new IR distance measuring one, yield analog instead of digital values. This means the reading from the sensor, instead of just open or closed, will be a value between 0 and 1024.

You may want to skip this section if you're not going to use any analog sensor – suffice to know you can use them if the need arises.

In order to use one of these, you'll have to configure it as analog (note we're using sensor 8, keep reading please):
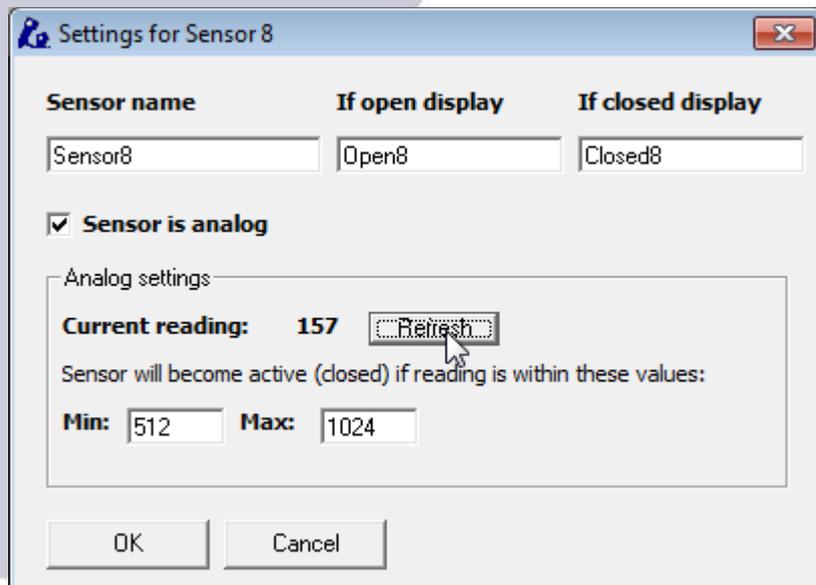
… as the image shows. Clicking "refresh" will update the current reading. The goal is determining the range of values where we can consider the object is in its place; in our example, the mount is parked, so we could:

- ✔ unpark the mount: the sensor should read near 0 (less than 20) if properly placed, but other values could do, too
- ✔ approach the mount to the parking position, clicking refresh and taking note of the values

In a few tries we'll be able to know in what range of values the mount is in a safe position; filling with these values the "Min" and "Max" fields:

Settings for Sensor 8

Sensor name	If open display	If closed display
Mount at park	FALSE	YES - parked

☑ **Sensor is analog**

Analog settings

Current reading: 164   [Refresh]

Sensor will become active (closed) if reading is within these values:

Min: 50   **Max:** 380

[OK]   [Cancel]
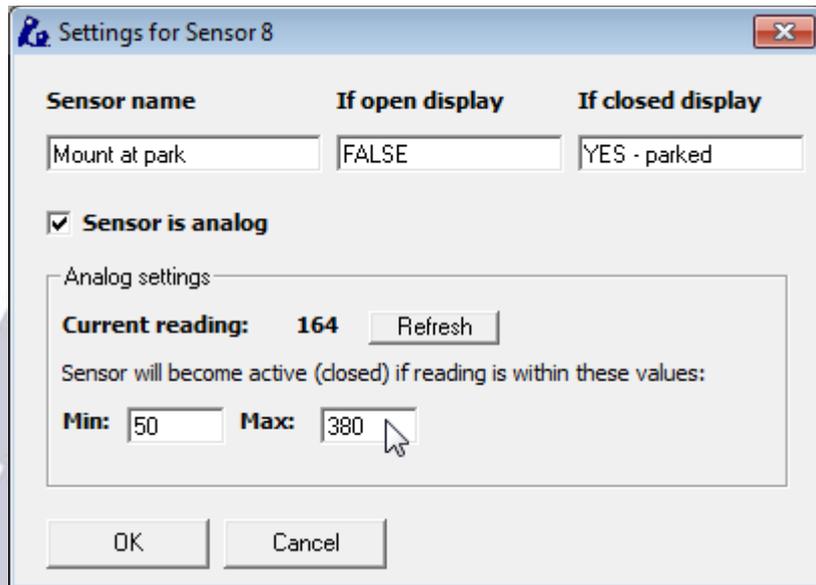
… the main Dragonfly window will update its display accordingly to the current position of the mount. As seen, we've also filled the rest of the fields (sensor name etc), and used sensor number 8 as it is more convenient due to the ground plugs placement. There is a separate instruction sheet for wiring this kind of sensors, just check our web site.

If you are going to use scripts with an analog sensor, please read the corresponding addenda.

4-d) Automated control - ASCOM

As previously mentioned, it is very important to be confident with the remote operation of the observatory before addressing any kind of automation.

Automation requires a bit more of effort; apart from the Dragonfly, we'll need some automation software (there are many in the market, such as Sequence Generator Pro, CCDCommander, CCD Autopilot, and ACP to name the most popular ones), automated focusing support, etc., and, most important, everything running smooth.

The ASCOM standard plays an important role here, as it enables different devices and programs to understand each other. In our case, the Dragonfly follows the ASCOM standard for domes, with its functionality reflecting that of a roll-off roof observatory.

So basically, thanks to the ASCOM standard, when our automation program wants to close the roof, the Dragonfly will be asked to do so. Three kind of messages or requests we'll get from the automation program:

- Open the roof
- Close the roof
- Tell me the status of the roof

For each of this messages, the Dragonfly software will launch a script. All Dragonfly scripts are located in its home folder (usually "c:\program files\dragonfly"), under the "dfscripts" folder (that is, "c:\program files\dragonfly\dfscripts"). For 64 bit Windows editions, the correct folder is "c:\program files (x86)\dragonfly\dfscripts".

> **Scared? - Don't be!**
> **Scripts can be very simple if the actions they command are simple. There are sample scripts with the Dragonfly software and we can help, even writing scripts for you, if you're stuck!**

We have one script for each ASCOM message, so:

- Open the roof: OpenShutter.vbs
- Close the roof: CloseShutter.vbs
- Tell me the status of the roof: ShutterStatus.vbs

Programming the scripts can be a bit scary at first, but it is (or can be) simple indeed. It implies, nonetheless, to think in advance and foresee possible situations.

On the plus side, scripts allow for **full customization**, and this is a big plus, worth the effort. For example, we can not only check if the mount is parked before closing or opening, but also command it to park issuing a few ASCOM calls from the script.

Let's work out the scripts for our sample observatory, leaving for the moment the park sensor as it adds complexity and not everybody needs it.

So, what do we want the Dragonfly to do when requested to open or close the roof?

I'd say, for the open case:

- check if it's already open, in that case we need to do nothing
- if it's not, then we need to push the roof control button, and wait for a period of time until the roof closes.
- If the period expires and the roof is still open, we should issue a warning.
- If the roof succesfully opens, we probably want the CCD and mount powered, and the lights powered off.

… for closing the roof it is of course very similar.

Before going on, please bear in mind:

- we can't teach how to program in this manual; we willl however explain things as clearly as possible so at the very least you can understand and modify the supplied scripts
- scripts are programs, usually simple ones but programs
- Your computer will read the scripts line by line, taking actions as per the script commands

If you are an accomplished programmer, well, you'll know what to do. There's a reference section a bit further in the manual.

So let's start...

We just have to program (most probably we'll just use one of examples, and maybe modify it) two scripts, one for opening, other for closing the roof.

The one for opening is to be called "SyncOpenShutter".

Here's a simple (slightly stripped down version of the script you'll find installed) but useful program; *the lines in blue, indented to the right, are not present in the script, just added documentation here.*

> All lines starting with a single quote ' are considered comments
> Write anything after the quote for documentation purposes

```
' Dragonfly SyncOpenShutter script
' (c) Lunatico 2016
'
' This will work "out of the box" if you follow the setup in the users'
manual:
' - you have 2 sensors for open (sensor 1) and closed (sensor 2) roof
' - the roof motor is activated by a pulse, and its control is attached
to relay 1
'
' Only thing you may want to adjust is the time alotted for the roof
moving, currently at 120 secs
' The pulse for the motor is set at 2 secs (2000 milisecs) and should
work in any setup
```

> To avoid nasty errors, any variable we use should be declared first, so...

```
Option Explicit
```

> Now we define a few numbers, that is, we give them a name, so the script is easier to understand
> The relay and sensor numbers matches the ones in the example we've been working through

```
' ASCOM Shutter State
const ShutterStatus_Open = 0  ' values from ASCOM standard, cannot be
changed!
const ShutterStatus_Closed = 1
const ShutterStatus_Opening = 2
const ShutterStatus_Closing = 3
const ShutterStatus_Error = 4

' Own constants
' define your own constants here

const Sens_OpenRoof = 1
const Sens_ClosedRoof  = 2
const Rel_Roof = 1

const Timeout_OpenClose = 120000    ' 120 secs time for roof to move
const RoofPulseLenght = 2000        ' 2 secs pulse for the roof "button"
```

> ... here we declare the variable Dfly (only one in this program) and set it as a Dragonfly "Help" object

```
Dim Dfly
set Dfly = CreateObject("Dragonfly.Help")
```

> We'll use the "Dfly" object to "talk" with the Dragonfly.
> Importante: first thing to actually do is notify we're opening the roof

```
df.AscomShutterStatus = ShutterStatus_Opening   ' opening
```

> Now we'll check if it is already open

```
' check if already open

if ( df.SensorDigRead( Sens_OpenRoof ) ) then   ' sensor marks roof as
open
    df.AscomShutterStatus = ShutterStatus_Open ' notify it is already
open
    df.LogAddLine( "Already open!" )
    ' nothing else will be done
else
```

> … in any other case – we assume it is closed, and simply go on

```
    Call Dfly.RelayPulse( Rel_Roof, RoofPulseLenght )     ' pulse to
the roof
```

> … sending a pulse to the roof.

```
    wscript.sleep( Timeout_OpenClose )
```

> … and wait a reasonable amount of time (20 seconds)
> before checking if it has opened.

```
    if ( Dfly.SensorDigRead( Sens_OpenRoof ) ) then
        Dfly.AscomShutterStatus = ShutterStatus_Open
    else
        Dfly.AscomShutterStatus = ShutterStatus_Error
    end if
```

> Now we have checked the "open" sensor if it's opened,
> update ASCOM to open, else to error

```
end if
```

> And that's all!

And that's all needed to have our simple observatory automated using ASCOM. With this scripts, any ASCOM-aware program will be able to open, close, and report roof status.

Very important

The Dragonfly software will execute the scripts found at its install folder, "dfscripts" subfolder.
If you program your own SyncOpenShutter.vbs and SyncCloseShutter.vbs scripts, you have to copy them to that folder, *__along with the supplied "OpenShutter.vbs", "CloseShutter.vbs" and "ShutterStatus.vbs" scripts.__*

Also, if you don't follow the "roof open is sensor 1, closed is sensor 2" standard, you'll have to tweak the "ShutterStatus.vbs"

The optional scripts for Relays and Sensor, explained next, also have to be placed in the same folder.

4-e) Automated control – on / off scripts

Additionaly, more scripts can be programmed for the Dragonfly to launch, adding versatility. As you may have noticed in the control panel:

| Relay Scripts: | OFF | Change |
| Sensor Scripts: | OFF | Change |
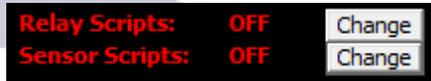
… you can enable / disable relay and sensor scripts. When enabled, and a change in the state of a relay or sensor is detected, the corresponding script will be launched.

The naming convention is as follows:

RelayClose1.vbs … RelayClose8.vbs: scripts to be executed when the given relay changes to closed.
RelayOpen1.vbs... RelayOpen8.vbs: same, when changes to open

SensorOn1.vbs … SensorOn8.vbs: when the sensor changes to ON
SensorOff1.vbs... SensorOff8.vbs: … and to OFF.

Only existing scripts will be taken into consideration, missing ones will not result in any kind of error – meaning if you want a script to run when sensor 4 becomes ON, you just have to write that one (SensorOn4.vbs) and enable sensor scripts. No problem at all with the missing scripts.

A simple but powerful application is to have a physical button terminate the session – park the scope, close the roof, switch off the computer. Another possibility is to use one sensor input detect power failures; there's an article explaining this in our web site.

Important considerations:

- The scripts will be executed when (or if) a change is detected. Very fast or spurious changes may not be detected at all (will depend on the refresh period configured).

- They will be executed **just once** for each change.

- Beware of script – script, and script – program interaction, for example:

    - If your automation program is in charge of monitoring the weather, do not just close the roof if a sensor detects unsafe weather; but you can

set it up so it just closes if after a safeguard period it's still open.

- If you want to switch off the lights (or the dehumidifier) when the roof starts to open, do it at a single place, either at the ASCOM OpenShutter script (will work only when opening via ASCOM) or with a sensor script when the sensor indicating "roof closed" changes to open (will work as long as the Dragonfly software is running).

Don't forget your scripts can do many more things than handling the Dragonfly inputs and outputs – they can launch external programs, access any ASCOM objects (such as the mount), command Windows to switch off...

Addenda: analog sensors

Even if you have properly configured the analog sensor settings as explained in the preceding section, that configuration will only affect the remote control panel (main window) of the Dragonfly.

For the sensors to behave as analog in the scripts, the method "**SensorAnRead**" must be called, and the result (will be between 0 and 1024) matched against the desired values.
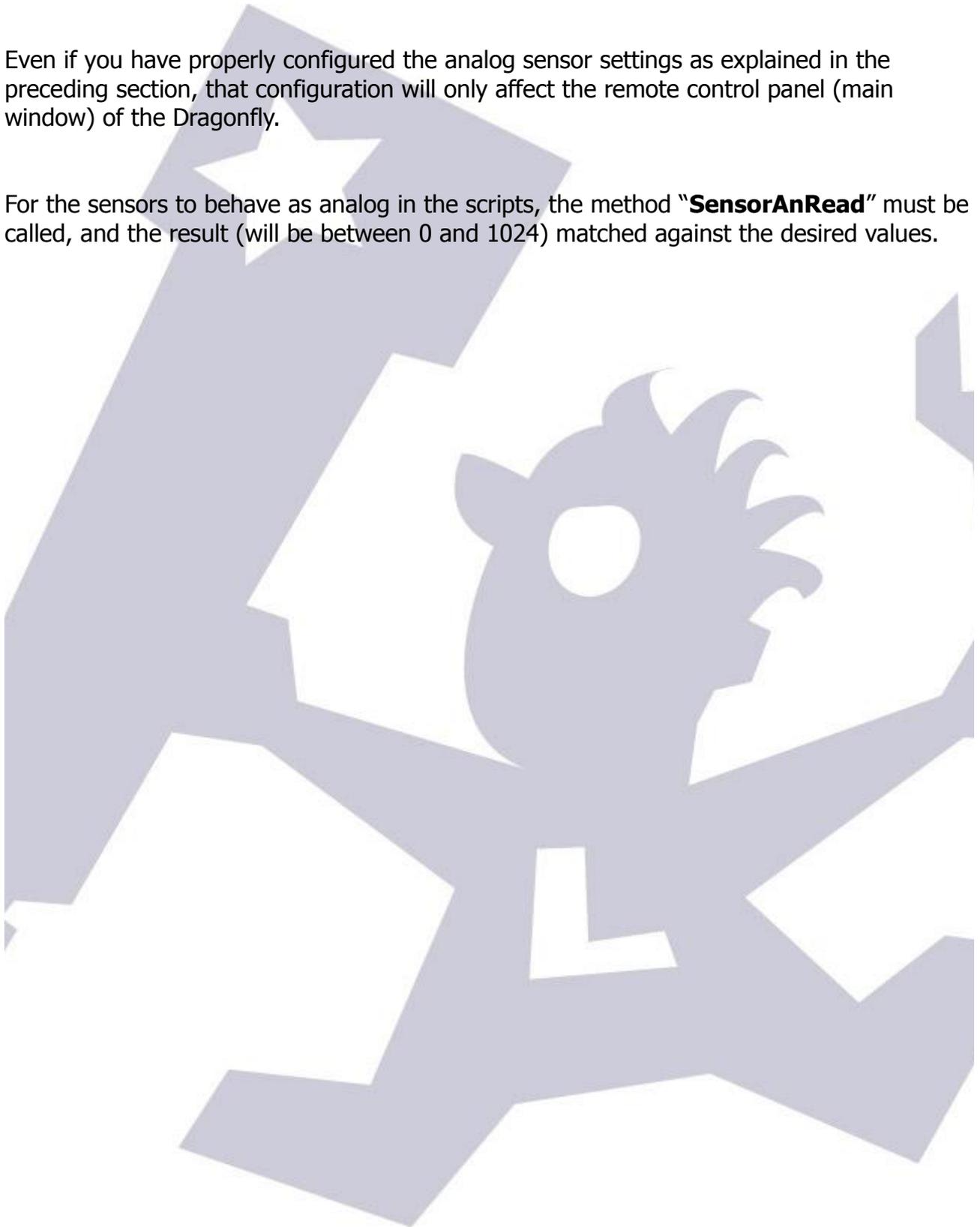
5. Dragonfly functions
(they are properly called methods and properties)

| Function	Description	Example(s)
`RelayOpen( relayNumber )`	opens the given relay	`RelayOpen(1)`
`RelayClose( relayNumber )`	closes the given relay	`RelayClose(1)`
`RelayChange( relayNumber )`	changes the relay status, opens it if closed and closes it if open. **Useful because the optional pulse applies to RelayChange**, never to RelayOpen or RelayClose.	`RelayChange( 3 )`
`RelayPulse( relayNumber, period )`	Pulses the given relay for the specified number of milliseconds.	`Call RelayPulse( 2, 1500 )`
`RelayRead( relayNumber )`	Reads if the given relay is energized / closed (TRUE) or not (FALSE)	
`SensorDigRead( sensorNumber )`	reads the given sensor digitally, that is, returns TRUE or FALSE.	
`SensorAnRead( sensorNumber )`	reads the given sensor analog, that is, returns a number between 0 and 1024.	

`timer( timerNumber )`	Reads or sets a countdown timer in milliseconds. Will stop counting when <= 0. It also sets "timerActive(timerNumber) to TRUE" - but will not set it to FALSE when 0 is reached!	`Timer( 1 ) = 5000`
		`if ( Timer(1) > 0 )`
		`then`
		`' count not finished...`
		`….`
		`end if`

`timerActive( timerNumber )`	Gets or sets the active status of a given timer. Set to TRUE if the timer is started to count	`TimerActive( 1 )`

`flag( flagNumber )`	User flag; can be read or written, to true or false	`Flag( 3 ) = TRUE`
		`if ( flag( 4 ) ) then`
		`…`
		`end if`

`UserVar( varNumber )`	Same as flags, but with integer (numeral) values	`UserVar( 7 ) = 435`
`LogShow`	Shows the log window	
`LogHide`	Hides the log window	
`LogClear`	Erases the log window contents	

`LogAddLine( text )`	Adds the given line to the log window	
`LogAddToLine( moreText )`	Appends the given text to the last written line	
`LogUpdateLine( newtext )`	Changes the contents of the last line of text	
`LogSetBold( TRUE or FALSE)`	Sets or unsets the last line as Bold characters	
`LogSetForeColor( color )`	Sets the color of the text written in the last line	
`LogSetBackColor( color )`	Sets the background color of the text written in the last line	
`AscomShutterStatus( status )`	Sets or reads the status of the shutter – this is accesible to other ASCOM programs.	

Please take a look at the included sample scripts (explore your software folder, ffscripts, for a "samples" folder) for well documented, clearly written ones.

6 – Network and controller configuration

As received from the factory, the network settings will work in almost any case. Both the program and the Dragonfly are configured to take an automatic network address and work with it.

Anyway, go ahead and press configuration from the window you minimized long ago and is now waiting in the lower left of the screen; you'll see these options:
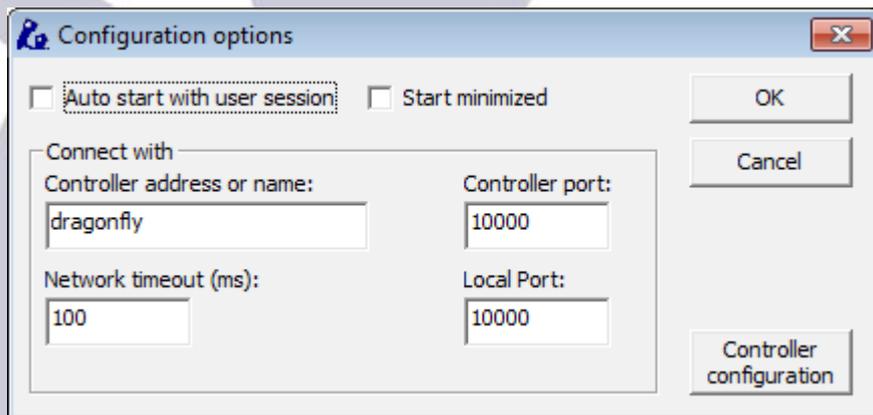
You will most likely never need to modify them. Basically:

- ✔ auto start with user session: check this option if you want the software to be launched (and connected to the Dragonfly) as soon as your windows session is started
- ✔ start minimized: will simply collapse this program to the notification area, just as we did before, so it will not occupy space in our screen.
- ✔ Controller address or name: just as any computer in your observatory network, the dragonfly has a name and an address. By default, its name is "Dragonfly" and its address will be assigned automatically. More on this later.
- ✔ Controller port, local port: unless there's a problem, this should not be changed. You can think of ports as "channels" within the computer or controller – so the controller will be listening at channel (port) 10000, and so will do the computer.
- ✔ Same applies to "Network timeout (ms)", this being the time the program will wait for a network response before assuming there's a problem.

Ok and Cancel performing as expected, only "Controller configuration" remains; these settings we've been seeing so far belong to the Windows program. Clicking "Controller configuration" gives us access to internal settings of the Dragonfly:

Nothing too complicated:

- ✔ Controller name: the name the controller will be known by across the network. Dragonfly is as good as any.
- ✔ Listen port: the port (channel in our previous explanation) the Dragonfly will be listen at.
- ✔ HTTP port: maybe better known, the port used by the internal web server of the Dragonfly – useful mostly for advanced programming or system integration

The next group will configure how the controller will get its network address, automatically (as selected by default and standard in the vast majority of networks), or static, in this later case you have to specify the address, mask and gateway.

Bear in mind: if you change any of these settings, and click "apply", they will be permanently stored in the dragonfly – and also inmediately applied -, most likely resulting in a disconnection. Just adjust the options belonging to the windows program to reestablish the link.

7. Remote observatory control tips

- Never use fluorescent lights! They are one of the strongest sources of electrical noise.

- Reed (magnetic) proximity switches are very useful to check for roof and mount position. Any will work with the Dragonfly. A simple one is available from RS components with ref. 289-7783 (don't forget to get the magnet, ref. 289-7812).

- If using pushbuttons to detect roof position (such as RS ref. 746-8605), the usual way is to place the pushbuttons in a fixed position (wall), and the part that will press them – a plastic angle is suggested, as it can bend and won't damage the pushbutton – in the moving roof.

- Plan carefully, be redundant if at all possible, and have an UPS strong enough to close the observatory in case of power failure.

8. Apendix

Dragonfly technical specifications:

- Power requirements: 12V dc (up to 24V will work), center positive, standard 5.2mm power jack, less than 1A drawn.
- External interface: 10/100 Mbit ethernet – TCP/IP v4
- Microcontroller: AT91SAM7XC256
- Relays (8):
  - Nominal switching capacity: 10A @ 125V/250V y 6A @ 277V
  - Max. switching voltage: 250 V AC, 100 V DC
  - Max. switching current: 10 A (AC), 5 A (DC)
  - Model Panasonic JS1-5V-F
- Inputs (8):
  - protected with Littelfuse 600R160UR resettable PTC
  - all 8 capable of analog and digital readout
- Aluminum folded box, connected to earth via any of the 16 earth plugs.

_____