

**SPOX**  
**User manual**



**François Cochard**  
(francois.cochard@shelyak.com)

3 décembre 2018

Ref. DC0033 - rev A

# 1

## Introduction

SPOX is an accessory designed for the Shelyak Instruments Lhires III, Alpy 600 and LISA spectroscopes. It has two functions.

- It allows for remote calibration & flat lamp control to minimise disruption of a sensitive telescope mount and spectrograph during the acquisition of calibration images.
- It provides a USB interface for PC software control to provide for automated calibration frame acquisition as part of unattended spectroscopic observations.

In addition, SPOX includes a lamp failure detection, with an Alarm LED (see section 2.6). *Note : this feature is not available for the LISA calibration module, for technical reasons.*

This documentation describes all the features, and the different ways to control the SPOX device locally or remotely.

The SPOX project has been developed by Shelyak Instruments, with a deep contribution of the Astronomy community members. We specially want to thank main contributors :

- **Joël Setton**, for the electronic design.
- **Jean-Paul Godard**, for all the software element under Windows (including ASCOM driver), as well as for the Arduino firmware.
- **Jean-Baptiste Butet**, for the software under Linux, including the Indi driver.
- **Paul Luckas**, for testing and help with documentation.

### 1.1 Package contents

SPOX comes shipped with the following items (see fig. 1.1)

- The SPOX control box
- A flyer to get this documentation and the software on Shelyak website<sup>1</sup>
- USB cable (1.8m)

Depending on the configuration you ordered, you can also get (see fig. 1.2)

- A 4 pin connector cable (0.8m or 5m)
- The Alpy & LISA adapter cable

SPOX interfaces with the motorised Shelyak calibration lamp assembly included on all Alpy calibra-



FIGURE 1.1 – SPOX and USB cable



FIGURE 1.2 – 4 pins cable and Alpy / LISA adapter cable (right)

tion module, LISA with calibration module and newer Lhires III spectrographs<sup>2</sup>. For older Lhires III models, this assembly can be purchased as an upgrade kit.

SPOX requires a user supplied 12V DC power supply delivering at least 1 Amp. Power input is through the 5.5mm DC socket (2.5mm centre positive) located on the SPOX control box.

The SPOX software has been tested for Windows 7 and Windows 10, but should operate equally well un-

1. At anytime you can find the latest version of the software and documentation on <http://www.shelyak.com>

2. All Lhires III shipped since 2016, or those equipped with the calibration upgrade kit are compliant with SPOX. If your Lhires III has two buttons for flat and calibration lamp, it is compliant with SPOX

der versions from Windows XP to Windows 10. SPOX can also work under Linux, through the Open Source Indi driver developped by Jean-Baptiste Butet.

## 1.2 Care in handling

SPOX is a solid state electronic device and requires some care in use. The SPOX control box, cables and power supply must be kept away from liquids. In the field, SPOX should be kept clear of damp or excessively dusty and humid conditions.

## 1.3 Enjoy SPOX & Spectroscopy

When developing SPOX, we had in mind to help the increasing spectroscopy community to simplify their observations. We also developed it in an Open Source mind, using an Arduino and working with this great community. As a result, we can provide all the detailed specification (mechanics, electronics) and source code for both Windows and Linux.

We hope that you'll enjoy using it, and thanks to SPOX, you'll do more and more high quality spectroscopic observations !

François Cochard  
Shelyak Instruments  
October 2018

# 2

## Connecting SPOX

If you plan to use Spox locally & manually, it should be install close to the operator – ideally at the computer workstation where it can be comfortably accessed while operating telescope and acquisition software.

If you plan to use SPOX remotely, then it must be placed as closed as possible to the spectroscope and the USB plug it will be connected to.

The minimal connections are Power supply and Spectroscopic cable (the connection depends on the spectroscopic model, see below sections). If SPOX is used remotely, you must also connect a USB cable between SPOX and your PC (or a USB hub, in some cases) – see fig. 2.1.



FIGURE 2.1 – All connectors plugged

Note : if you've ordered your SPOX without any cable, you can find in 'Appendix C – Cables technical description' the technical details to make your own cable.

### 2.1 To the Alpy calibration

To connect SPOX to your Alpy 600 :

1. Connect one end of the 4 pin control cable to the matching socket on the SPOX (fig. 2.2).
2. Connect the other end of the control cable to the additional cable for Alpy 600 (fig. 2.3).
3. Connect the black connector of the additional cable to the center connector of the Alpy calibration module (this is the power supply connector) - see fig. 2.4.



FIGURE 2.2 – Connect the 4 pins cable



FIGURE 2.3 – Connect the 4 pins cable to the Alpy adapter cable

4. Connect the yellow connector of the additional cable to the Calibration lamp connector of the Alpy calibration module (fig. 2.5).
5. Connect the white connector of the additional cable to the Flat lamp connector of the Alpy calibration module (fig. 2.6).
6. Ensure that both switches on the Alpy calibration module are in the "off" position (fig. 2.7).
7. Connect a 12V DC 1Amp power supply (center positive) to the SPOX. The red LED switches on for one second to confirm that power supply is OK (fig. 2.8).

Your SPOX is now ready for use.



FIGURE 2.4 – Connect the black connector to the Alpy center pin



FIGURE 2.5 – Connect the yellow connector to the Alpy calibration lamp

## 2.2 To the Lhires III

To connect SPOX to your Lhires III :

1. Connect one end of the 4 pin control cable to the matching socket on the SPOX (fig. 2.2).
2. Connect the other end of the control cable to the 4 pin socket located on the side of the Lhires III (fig. 2.9).
3. Ensure that both switches on the Lhires III are in the “off” position (fig. 2.10).
4. Connect a 12V DC 1Amp power supply (center positive) to the SPOX. The red LED switches on for one second to confirm that power supply is OK (fig. 2.8).

Your SPOX is now ready for use.

## 2.3 To the LISA

To connect SPOX to your LISA calibration module :

1. Connect one end of the 4 pin control cable to the matching socket on the SPOX (fig. 2.2).
2. Connect the other end of the control cable to the additional cable for LISA (fig. 2.3).
4. Connect the yellow connector of the additional



FIGURE 2.6 – All connections must look like this

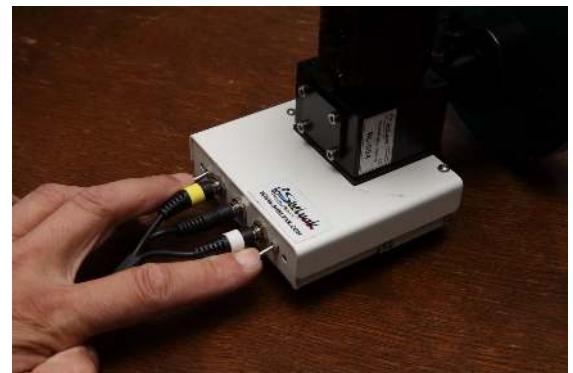


FIGURE 2.7 – Switch off both buttons

cable to the Calibration lamp connector of the LISA calibration module (fig. 2.11).

5. Connect the white connector of the additional cable to the Flat lamp connector of the LISA calibration module (fig. 2.11).
5. The black connector of the additional cable is not connected. To prevent any short-circuit, you can attach it to the cable like on fig. 2.12.
6. Ensure that both switches on the LISA calibration module are in the “off” position.
7. Connect a 12V DC 1Amp power supply (center positive) to the SPOX. The red LED switches on for one second to confirm that power supply is OK (fig. 2.8).

Your SPOX is now ready for use.

## 2.4 Sticker on back face

You'll find a sticker on SPOX back face to indicate details for connexions (fig. 2.13).

## 2.5 Basic operation

The buttons on SPOX correspond to the lamp switches on the Alpy, Lhires III and LISA spectrographs and are used to control the flat and neon lamps in the



FIGURE 2.8 – Connect the power supply to SPOX



FIGURE 2.9 – Connect the cable to the Lhires III

same way. However, using the SPOX instead of the switches on the spectrograph ensures that the image train is ‘un-touched’ – greatly reducing any chance of disturbing the telescope as it tracks an target centered on the slit (fig. 2.14).

**Important :** For SPOX to be able to control the lamps, both lamp switches on the spectrograph need to be in the “off” position.

A description of the buttons and LED indicators is as follows :

1. Calib. Pressing this button activates the spectrograph’s calibration lamp. An LED indicates that the lamp is on and in position – ready for calibration frame acquisition. Pressing the button again deactivates the calibration lamp and the LED turns off.
2. Flat. Pressing this button activates the spectrograph’s tungsten lamp used for acquiring flat field frames. An LED indicates that the lamp is on and in position. Pressing the button again deactivates the flat lamp and the LED turns off.
3. Pressing both the Calib and Flat switches in succession will activate “dark mode”. In this mode, the spectrograph’s calibration assembly is positioned to shield the slit from incoming light to facilitate the acquisition of dark frames. Both Flat and calibration lamps are turned off. Both LEDs



FIGURE 2.10 – Switch off both buttons

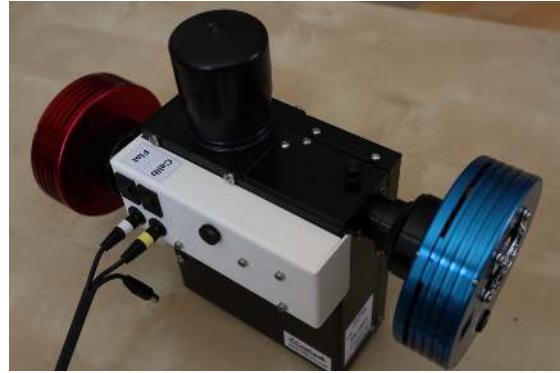


FIGURE 2.11 – white &amp; yellow connectors to the LISA

will flash simultaneously to indicate that the slit is covered and that the lamps are off. Pressing one or both buttons in succession will return the unit to one of three modes, Calib, Flat or Off – as indicated by the LED status.

*Important note : this feature is not available for the LISA calibration module.*

**Note :** If inadvertently left on, the SPOX lamp circuitry will power off automatically after 30 minutes to prolong lamp life. When this happens, the corresponding LED will flash. To reset operation, simply switch the lamp off and on again using the SPOX buttons.

It is still possible to operate the switches on the Lhires III, ALPY 600 and LISA without adversely affect SPOX. However, for SPOX to provide remote control, switches on the Lhires III must be in the “off” position.

## 2.6 About the Alarm LED

It may happen that one of the lamp (calibration or flat) is broken. In this case, you’ll have a black image instead of the flat or calibration spectrum. To help you understanding what is happening, we’ve added a current detection in SPOX. If the actual current is below a given threshold, then the alarm LED (red) switches on.



FIGURE 2.12 – Attach the unplugged connector



FIGURE 2.13 – Sticker on back face

In other words, when Alarm LED is on, it means that the current within the LED is below the expectation, suspecting that the matching lamp is broken.

You can tune the Alarm thresholds, to adapt it to your own setup. Tuning the Alarm threshold is detailed in the ‘Appendix B – Tuning the Alarm threshold’

*Note : if you use SPOX without any spectroscope calibration plused in, then the Alarm LED will switch on each time you try switching on any lamp. This is normal : no current is sent to the spectroscope. See fig. 2.15*



FIGURE 2.14 – Alpy calibration remote control in the field



FIGURE 2.15 – The Alarm LED is on

# 3

## Computer control

Most users of SPOX will want to take advantage of its computer interface. In this manner, SPOX can be used to drive lamp control using software, ultimately providing for unattended and scripted operation where calibration and spectral images can be acquired in succession automatically and without user intervention.

There are many ways to control SPOX with your PC, under Windows or Linux :

- Use the provided standalone software, under Windows
- Use an ASCOM compliant software (like The SkyX, Prism, and many others) ; SPOX is considered as a switch box.
- Create your own software, using the provided DLL (Windows), or even the COM protocol (see Appendix A – SPOX protocol).
- Under Linux, you can use the Indi driver ; SPOX is considered as a switch box.
- Under Windows, you can also use a command line terminal.
- Under Linux and Windows, you can use a Python script (an example is given in Appendix A – SPOX protocol).

### 3.1 Software installation & configuration

The SPOX software is dedicated to Windows. It includes both stand alone software and an optional ASCOM compliant driver that provides the software interface between the SPOX hardware and compliant control software. There is one installer for the stand alone software, and another one for the ASCOM driver.

Software installation follows the conventional Windows procedure - with SPOX disconnected from USB during software installation.

The SPOX software can be found at Shelyak website : [www.shelyak.com](http://www.shelyak.com) (in section Products / Accessories / SPOX, tab ‘Software’).

#### Standalone software

The file to install is named SPOX\_setup.exe.

To enable software control :

1. Verify the basic operation of SPOX as per section 2.5.
2. Do not connect the USB cable.
3. run the file SPOX\_setup.exe, and follow the installation instructions.
4. Once the software is installed, ensure SPOX is powered by plugging in the DC connector.
5. Connect SPOX to your PC using the USB cable. Within a few seconds, Windows should report the successful installation of your SPOX hardware. Look at the Windows parameters to see which COM port SPOX is connected to.
6. Run the SPOX program (Spox.exe). You should see the fig. 3.1.

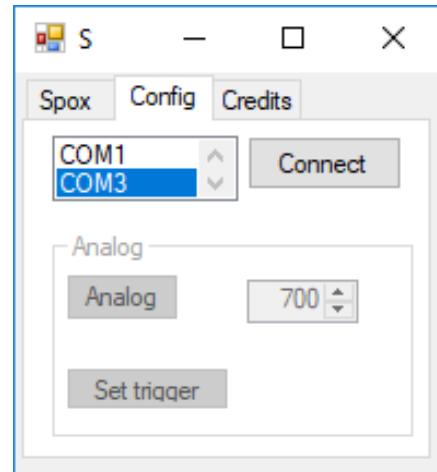


FIGURE 3.1 – Connexion tab

7. Select the COM port that matches the SPOX connexion, and click on connect. Within a few seconds, the program is connected, and switches to the ‘Control’ tab (fig. 3.2).

At this stage, you can use SPOX the same way as the hardware box (fig. 3.3).

*Important note :* when using the standalone software, you can still control manually the Spox hardware (push any button). The manual changes will be reported in the software.

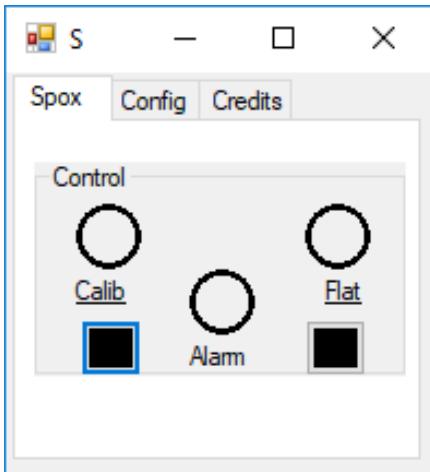


FIGURE 3.2 – SPOX usage tab

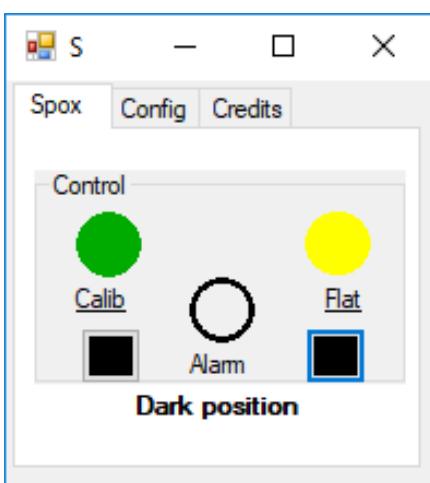


FIGURE 3.3 – SPOX usage with LEDs on

## ASCOM driver

The file to install is named SPOX-ASCOM Setup.exe  
To enable software control :

1. Verify the basic operation of SPOX as per section 2.5.
2. Do not connect the USB cable.
3. run the file SPOX-ASCOM Setup.exe, and follow the installation instructions.
4. Once the software is installed, ensure SPOX is powered by plugging in the DC connector.
5. Connect SPOX to your PC using the supplied USB cable. Within a few seconds, Windows should report the successful installation of your SPOX hardware. Look at the Windows parameters to see which COM port SPOX is connected to.
6. A basic ASCM SPOX test module is installed, for preliminary tests. Run the ASCOM SPOX program (ASCOM.Spox.Test.exe). You should see the fig. 3.4.
6. Go to the parameters tab, select ASCOM SPOX

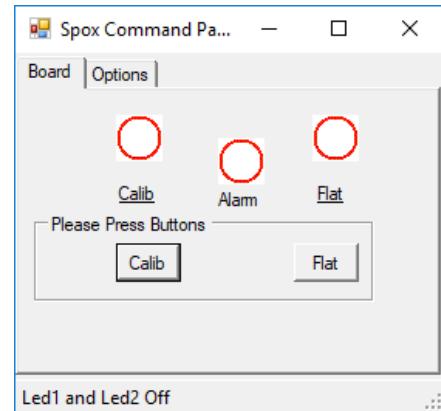


FIGURE 3.4 – SPOX ASCOM test software

device (fig. 3.5)

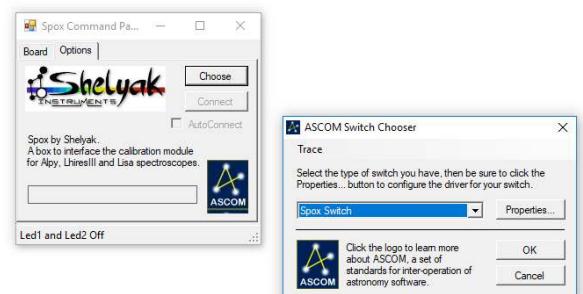


FIGURE 3.5 – SPOX ASCOM connexion

At this stage, you can use SPOX the same way as the hardware box.

Now, you can go to your own ASCOM compliant software, and connect SPOX the same way (it is considered as a switch box).

## 3.2 Other software options

### Using Indi driver

You can work under Linux with the Indi driver developed by Jean-Baptiste Butet. It is now part of the official Indilib distribution. Then, any Indi compliant software like K-Stars, Skychart, CCDCiel, and so on can control SPOX as a switch box.

### Scripted operation (Windows)

When the Standalone software is installed, SPOX can be operated from the Windows command line specifying arguments that correspond to button functions. In this way it is possible to execute SPOX functionality in user created scripts as well as astronomy automation software that supports scripting to fully automate the unattended acquisition of calibration

images during a spectroscopy. Command line execution is accomplished using the following syntax :

```
COMx i on/off
```

where :

- COMx is the COM Port used by SPOX (COM1, COM3...).
- i is the button number : 1 for “calib” and 2 for “flat”
- “on” or “off” is the desired state.

For example, to activate the calibration lamp with SPOX connected to COM5 from the command line :

```
COM5 1 on
```

Functions can be executed from scripts, batch files and programs using the above syntax.

### 3.2.1 Using your own code

Another way to work with SPOX is to create your own code. To do so, you can either use the Windows DLL, which is part of the package software, or control directly SPOX through the serial communication. You’ll find all details in the Appendix A – SPOX protocol.

### 3.2.2 Using Python

Python is a common language in the Astronomical community. We’ve written a quick demo script in Python to show you how to use SPOX through the serial port. This code can be reused in any of your application. See Appendix A – SPOX protocol.

---

# 4

## Troubleshooting

---

Most problems encountered with SPOX can be traced to problems with power or cabling. The following is a list of the most common issues likely to be encountered when using SPOX.

- The SPOX buttons do not activate anything, and LEDs on the SPOX control box do not illuminate.
  - Check that SPOX is receiving power.
- LEDs on the SPOX control box illuminate, but nothing is happening on the spectrograph.
  - Check that the control cable is correctly plugged into the SPOX control box and the spectrograph.
  - Check that the control cable or socket pins are not damaged.
  - Check that the switches on spectroscppe are in the off position.
  - Check operation of the spectrograph calibration assembly by operating the manual switches on the Lhires III or Alpy 600.
- Operation seems normal, but there is no calibration or flat image produced.
  - Check the state of the lamps inside the spectrograph.
- An LED is flashing on the SPOX.
  - This is normal and indicates that a lamp has been left on for over 30 minutes and is now in safety mode. Switch the associated button off and on again.
- Both LEDs are flashing on the SPOX.
  - This is normal and indicates that the unit is in “dark mode” (the calibration assembly is positioned to shield the slit with lamps off). Switch one or both SPOX buttons to off.
- Control via PC is not working.
  - Verify basic operation of SPOX (refer to section 2.5).
    - Check the connection of the USB cable.
    - Try a different USB cable or different USB port on the PC.
    - Troubleshoot USB driver installation.

# 5

## Technical Specifications

- Dimensions (without cables)
  - Width = 77mm
  - Depth = 60mm
  - Height = 30mm
- Weight (without cables)
  - Weight = 110 gr
- Power Supply Voltage and Current requirements
  - 12V, 1 amp min.
- Power connector
  - This is a standard connector : 5.5mm (ext.) x 2.5mm (int). Center pin is positive.
- Control Connector (to spectroscope)
  - This is a ‘tiny XLR’ type connector, with 4 pins.
- USB connector
  - USB ‘type B’ connector
- Operating Temperature
  - -15°C to +45°C

---

# Appendix A – SPOX protocol

---

This chapter defines the SPOX protocol. A python example is also given, to illustrate the protocol.

## 5.1 Hardware connexion

The SPOX device is based on an Arduino Pro Mini, with an ATmega168. The connexion with the PC is made through a USB interface that emulates a RS232 serial port (using a chip FTDI FT232R USB UART). The baudrate for the serial port is 9600 bauds. Other parameters are standards. Opening the serial port with a Python script is done by this instruction :

```
port_serie = Serial(port="/dev/ttyUSB0", baudrate=9600,  
                     timeout=1, writeTimeout=1)
```

At the connexion, SPOX replies with this string : 'Spox Initialized'. Reading the serial port and checking this string is received is the way to check that connexion is properly established.

## 5.2 Protocol

### 5.2.1 Sending an order

The protocol is very basic : you can switch on/off the two channels (flat or Calibration) by sending an ASCII string on the serial port with this format :

NX+CRLF, where N is the channel (Calib = 1, Flat = 2) and X is the order (1 = ON, 0 = OFF). CRLF means "Carriage Return and Line Feed", which is a common way to close an order on a serial port.

For instance, with a Python script, you can switch on the calibration light by sending this order to serial port :

```
orderString = '11\r\n'
```

Spox replies to each order by the same string. With the above order, you can read the following on the serial port : '11\\r\\n'

### 5.2.2 Query the SPOX status

Since SPOX can be controlled manually, the status of each LED (or output) can change out of the PC control. Then, it is necessary to be able to query the status of each LED at anytime. It can be done in a very similar way to sending an order : just replace the order (1 for ON, 0 for OFF) in the string by a '?'. Then, requesting the Flat LED status is made by sending the string :

```
orderString = '2?\r\n'
```

The reply from SPOX is the same as for an order ; you'll get '21\r\n' if the LED is ON, and '20\r\n' if it is OFF.

### 5.2.3 Query the Alarm status

SPOX includes a current detection to ensure that lamps are really working. If the Alarm LED is ON, it means that the lamp is not working properly. To query the Alarm LED, send this order to the serial port :

```
orderString = '0X\r\n'
```

The reply from SPOX is 'X1\r\n' if the LED is ON, and 'X0\r\n' if it is OFF.

### 5.2.4 Other commands

The Alarm detection is made thanks to an analog current measurement. You can change the current threshold by sending this order to the serial port (the value is kept in flash memory) :

```
orderString = 'NAdddd\r\n'
```

where N is the channel (1 for Calib, 2 for Flat), and dddd is the digital value for the Threshold.

For instance, if you want to set the threshold for the LED "Flat" to 532, send this order to the serial port :

```
orderString = '2A0532\r\n'
```

You can also query the Analog actual value, to help you defining the thresholds (see SPOX documentation). The reply is 'As\r\n', whatever is the channel

To query the actual current value, send this order to the serial port :

```
orderString = '0A\r\n'
```

The reply is something like 'An361\r\n'. In this example, the value measured for the current is 361.

Last available command, you can switch off the two channels at the same time by this order :

```
orderString = '00\r\n'
```

### 5.2.5 Error message

If you send a message that SPOX cannot understand, it replies by a 'SPOX\r\n' message.

## 5.3 Python demo script

The Python script 'SPOX\_protocol\_demo.py' gives an illustration of most of the above commands. It can be found on the Shelyak website<sup>1</sup>.

This script has been written with the Python 3 Linux IDLE environment, and the commands (like FlattOn() for instance) can be written in the IDE terminal.

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
#-----
#-----#
# Script for SPOX protocol demo
# Original Version : F. Cochard - August 2018
#-----#
#-----#
#-----#
#-----#
#-----#
from serial import * # Manages the serial port

global port_serie # serial port number
#-----#
# Functions definition
#-----#
def SpoxConnect():
    global port_serie
    TIMEOUT_VALUE = 3 # Timout = 3s for connexion
    try: # I try to open the COM Port
        port_serie = Serial(port="/dev/ttyUSB0",
                             baudrate=9600, timeout=1, writeTimeout=1)
    except SerialException:
        print("Serial port not responding")
        return (-1)
    Reply_received = False
    timeout_limit = time.time() + TIMEOUT_VALUE
    timeout_reached = False
    while (Reply_received == False and timeout_reached == False):
        if port_serie.in_waiting:
            reply = str(port_serie.readline(),'ascii')
            print(reply)
            Reply_received = True
            return("Connexion OK")
        if (time.time() >= timeout_limit) :
            timeout_reached = True
            return("Timeout reached")

def SpoxClose():
    global port_serie
    port_serie.close()

def SendToSpox(Button, Status):
    global port_serie
    TIMEOUT_VALUE = 3 # Timout = 3s for connexion
    data = str(Button) + str(Status) + '\r\n'
    data = data.encode('ascii')
    print("Message sent =", data)
    port_serie.write(data) # envoie le message
```

```
Reply_received = False
timeout_limit = time.time() + TIMEOUT_VALUE
timeout_reached = False
while (Reply_received == False and timeout_reached == False):
    if port_serie.in_waiting:
        reply = str(port_serie.readline(),'ascii')
        reply = port_serie.readline()
        Reply_received = True
    if (time.time() >= timeout_limit) :
        timeout_reached = True
        reply = "Timeout reached"
    print("Reply =", reply)
    if (reply == data):
        return("OK")
    else :
        return("Bug")

def QuerySpoxLED(Button):
    global port_serie
    TIMEOUT_VALUE = 3 # Timout = 3s for connexion
    data = str(Button) + '?\r\n'
    data = data.encode('ascii')
    print("Message sent =", data)
    port_serie.write(data) # envoie le message
    Reply_received = False
    timeout_limit = time.time() + TIMEOUT_VALUE
    timeout_reached = False
    while (Reply_received == False and timeout_reached == False):
        if port_serie.in_waiting:
            reply = port_serie.readline()
            Reply_received = True
        if (time.time() >= timeout_limit) :
            timeout_reached = True
            reply = "Timeout reached"
        print("Reply =", reply)
        if (int(chr(reply[1])) == 0):
            return("Led OFF")
        elif (int(chr(reply[1])) == 1):
            return("Led ON")
        else :
            return("Bug")

def QuerySpoxAlarm():
    global port_serie
    TIMEOUT_VALUE = 3 # Timout = 3s for connexion
    data = '0X\r\n'
    data = data.encode('ascii')
    print("Message sent =", data)
    port_serie.write(data) # envoie le message
    Reply_received = False
    timeout_limit = time.time() + TIMEOUT_VALUE
    timeout_reached = False
    while (Reply_received == False and timeout_reached == False):
        if port_serie.in_waiting:
            reply = port_serie.readline()
            Reply_received = True
        if (time.time() >= timeout_limit) :
            timeout_reached = True
            reply = "Timeout reached"
        print("Reply =", reply)
        if (int(chr(reply[1])) == 0):
            return("Alarm OFF")
        elif (int(chr(reply[1])) == 1):
            return("Alarm ON")
```

---

1. <http://www.shelyak.com>, see SPOX page.

```
else :
    return("Bug")

def FlatOn():
    reply = SendToSpox(2,1) # Flat=Button 2, 1=ON
    return(reply)

def FlatOff():
    reply = SendToSpox(2,0) # Flat=Button 2, 0=OFF
    return(reply)

def CalibOn():
    reply = SendToSpox(1,1) # Calib=Button 1, 1=ON
    return(reply)

def CalibOff():
    reply = SendToSpox(1,0) # Calib=Button 1, 0=OFF
    return(reply)

def AllOff():
    reply = SendToSpox(0,0) # '00" is the order to switch
                           # off both LEDs
    return(reply)

def QueryFlat():
    reply = QuerySpoxLED(2) # Flat=Button 2
    return(reply)

def QueryCalib():
    reply = QuerySpoxLED(1) # Calib=Button 1
    return(reply)
```

## Appendix B – Tuning the Alarm threshold

As explained in section 2.6, SPOX includes a current detection to alert you in case of breaking a lamp<sup>2</sup>. The system is simple : if the current passing to the spectroscope (when flat or calibration lamp is on) is below a given threshold, then the Alarm LED (red) is switched on.

The detection threshold can be tuned, with the Windows software (or with RS232 commands). Here is the procedure to do it.

First, you must define the threshold value to set, depending on your configuration. Go to the ‘Config’ tab of the software, and click on the ‘Analog’ button. A value will be displayed, like in fig. 5.1. This value gives the actual current sent to the spectro. In fact, this is not the real physical value (in Amps), but this is the digital conversion of this value – then, it has no unit.

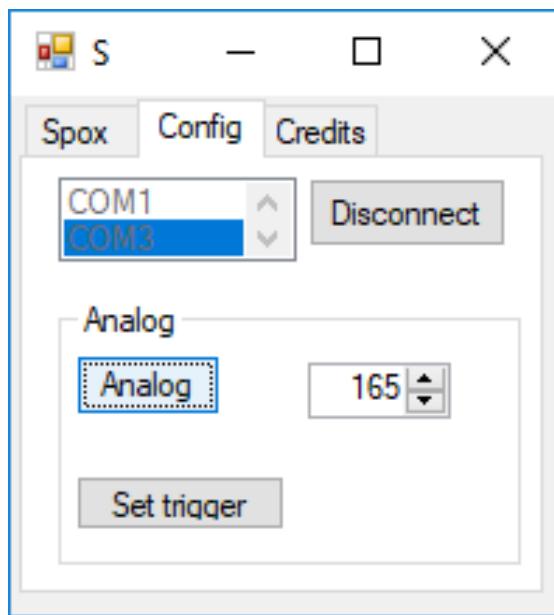


FIGURE 5.1 – Getting actual current value

You can do the measurement several times : it is stable, but not always exactly the same value.

Depending on the position of the switches (off, Calib, or Flat), you will not get the same value. For example, with a Lhires III, the value for OFF position is ~13, it is ~172 for the calibration lamp, and ~377 for the flat lamp. This is normal that current is higher for the flat lamp.

In our case, the current is above, say 170 when one lamp is on (flat or calibration). Then, we can use this threshold for the detection. We can also take some

margin, and choose 120.

To set the threshold to 120, just write 120 in the selector, and click on ‘Set trigger’ button. That’s it : the threshold is now set to 120 (see fig. 5.2). You can now test that the Alarm LED never switches on when flat or calibration lamp is on, and it switched on when you unplug the cable to the spectroscope.

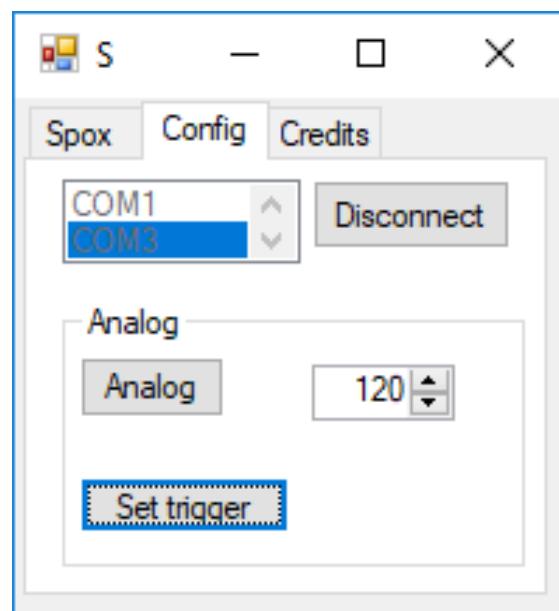


FIGURE 5.2 – Setting the threshold value

*Note : if you put 0 as a threshold, you totally deactivate the current detection – the Alarm LED will never switch on.*

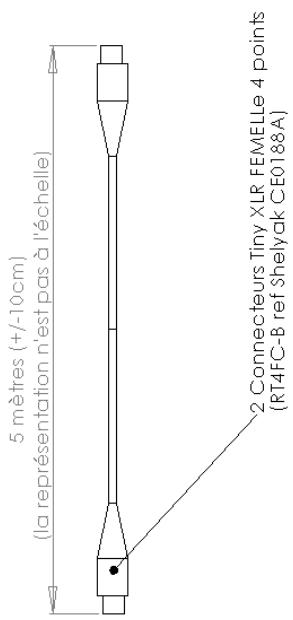
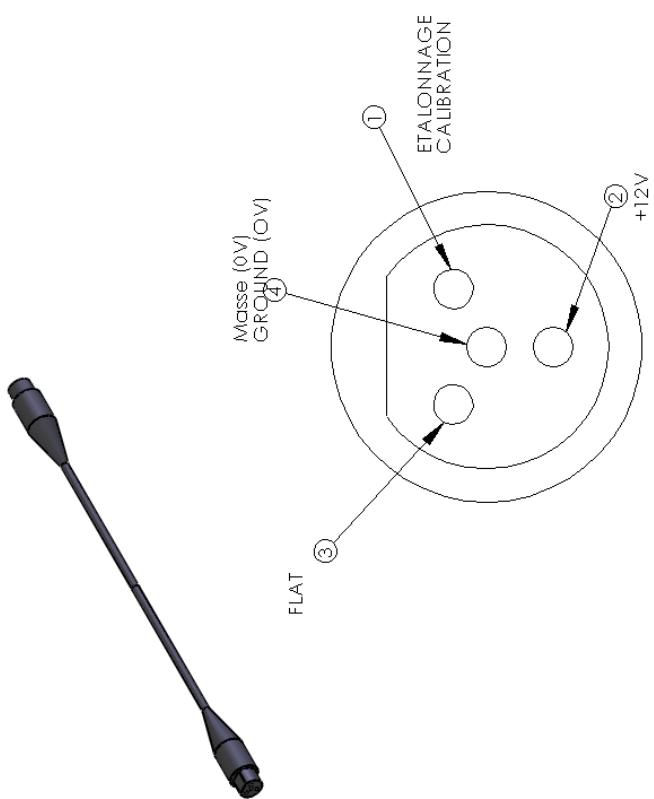
2. Reminder : this feature is not available for the LISA, for technical reasons

---

## Appendix C – Cables technical description

---

In the following pages, you'll find the technical drawings for the two SPOX cables.



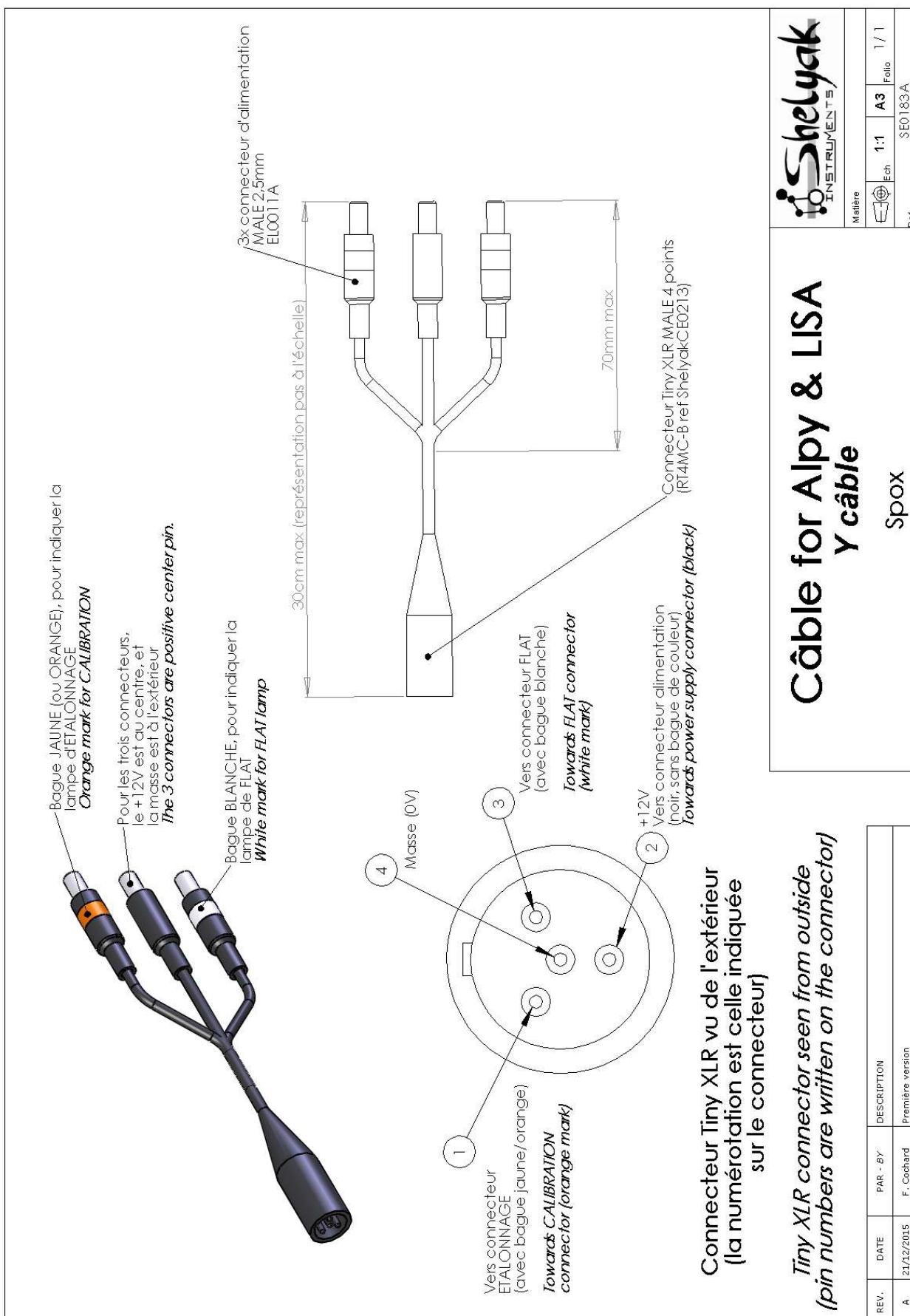
Connecteur Tiny XLR vu de l'extérieur  
[la numérotation est celle indiquée  
sur le connecteur]  
Les deux connecteurs sont reliés point à point.

Connector seen from outside.  
Pin numbers is the one written on the connector  
Both connectors are pin-to-pin connected.

## Cordon principal **Main cable**

(SPOX)

	Matière
Électro	Ech 1.2 A3 Folio 1/1 Ref SEO182A



**Shelyak**  
INSTRUMENTS

Matière  
Ech 1:1 A3 Folio 1 / 1  
Ref SEO183A

## Câble for Alpy & LISA Y câble

SPOX

REV.	DATE	PAR - BY	DESCRIPTION
A	27/12/2015	F. Cochard	Première version