


# Getting started - LabSpec Script and ActiveX

LS6 Knowledge Database

Exported on 10/21/2020

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	1 – Scripting automation (inside LabSpec).....	4
1.1.1	a – LabSpec interface.....	4
1.1.2	b – Programming the ActiveX. ....	5
<b>2</b>	<b>2 – ActiveX Automation ( in other code editor ) .....</b>	<b>6</b>
2.1	a – Integrating the LabSpec ActiveX.....	6
2.2	b – Programming an application using LabSpec ActiveX.....	8
<b>3</b>	<b>Getting started - LabSpec Script and ActiveX - .NET Visual Studio 2019.....</b>	<b>9</b>
<b>4</b>	<b>Getting started - LabSpec Script and ActiveX - Labview 2019 .....</b>	<b>16</b>
<b>5</b>	<b>Getting started - LabSpec Script and ActiveX - Matlab R2020a .....</b>	<b>25</b>
<b>6</b>	<b>Getting started - LabSpec Script and ActiveX - Python 3.7- Anaconda 3 - Spyder 4.0.1 IDE .....</b>	<b>29</b>

 This introduction is the first chapter of a more complete document, including LabSpec commands documentation, available in LabSpec6 release pack in :  
CURRENT\_RELEASE\_FOR\_CUSTOMER\User Manuals\LabSpec\_6\_5\_ActiveX-VBS\_documentation.pdf

# 1 Introduction

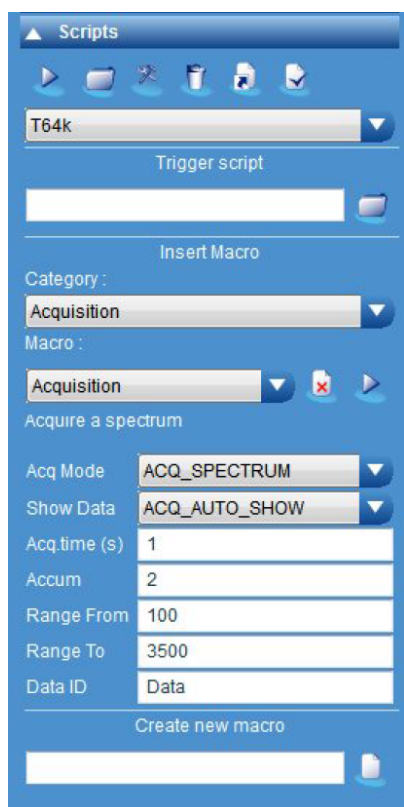
LabSpec 5/6 allows developers to automate its major functionalities, using either a scripting language or even an ActiveX module that can be integrated to any application that supports ActiveX technology.

One can choose any of these automation technics, depending on their needs : The scripting capability allows the to write macro commands in Visual Basic Script (VBScript) to drive the motors, acquire data, and use the treatments functions, using the LabSpec interface.

This is the easiest way to automate LabSpec. The ActiveX capability allows to fully integrate LabSpec in a 3rd party application. LabSpec interface is not loaded and requires a full graphical interface, giving much more flexibility.

## 1.1 1 – Scripting automation (inside LabSpec)

### 1.1.1 a – LabSpec interface



*LabSpec 6 VBS Script Management interface*

A very simple interface is available in LabSpec 5 (Scripts menu/Options) and labSpec 6 (Processing/Scripts).

It is possible to add, remove or configure a script, and to start it. A script can be executed just before and just after an acquisition using the Acquisition Trigger Before and After parameters. It is also possible to add up to 5 script controlled motors.

These motors can drive any hardware you want (see GetMotorAction Function)

### 1.1.2 b – Programming the ActiveX.

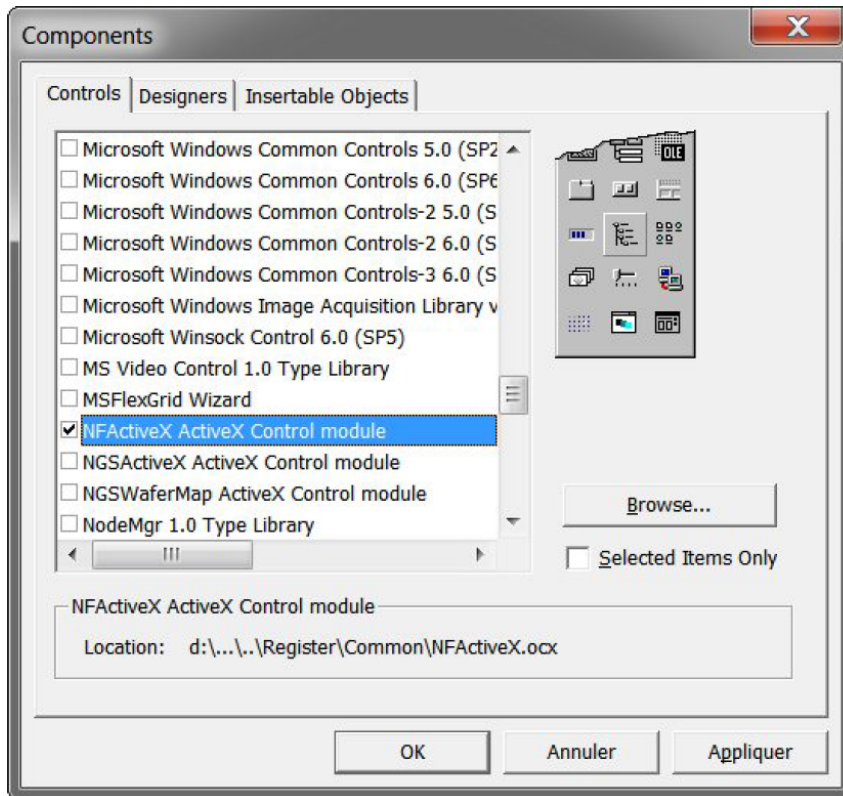
If a script is started from LabSpec, it automatically includes a object called LabSpec. This object manages the main LabSpec functionalities.

In order to use one of these functions, simply add : ReturnedValue =

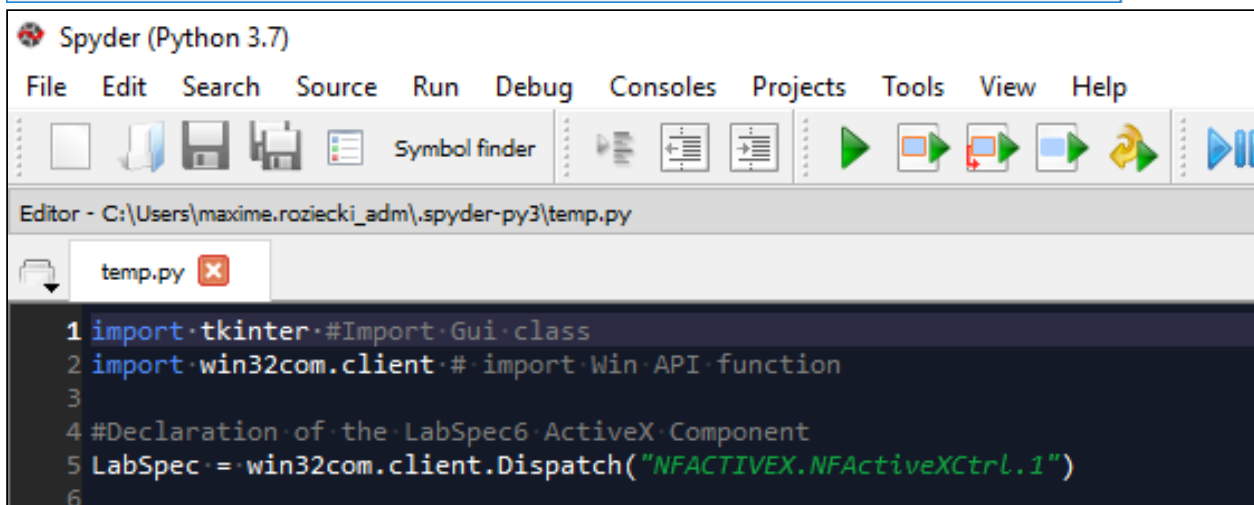
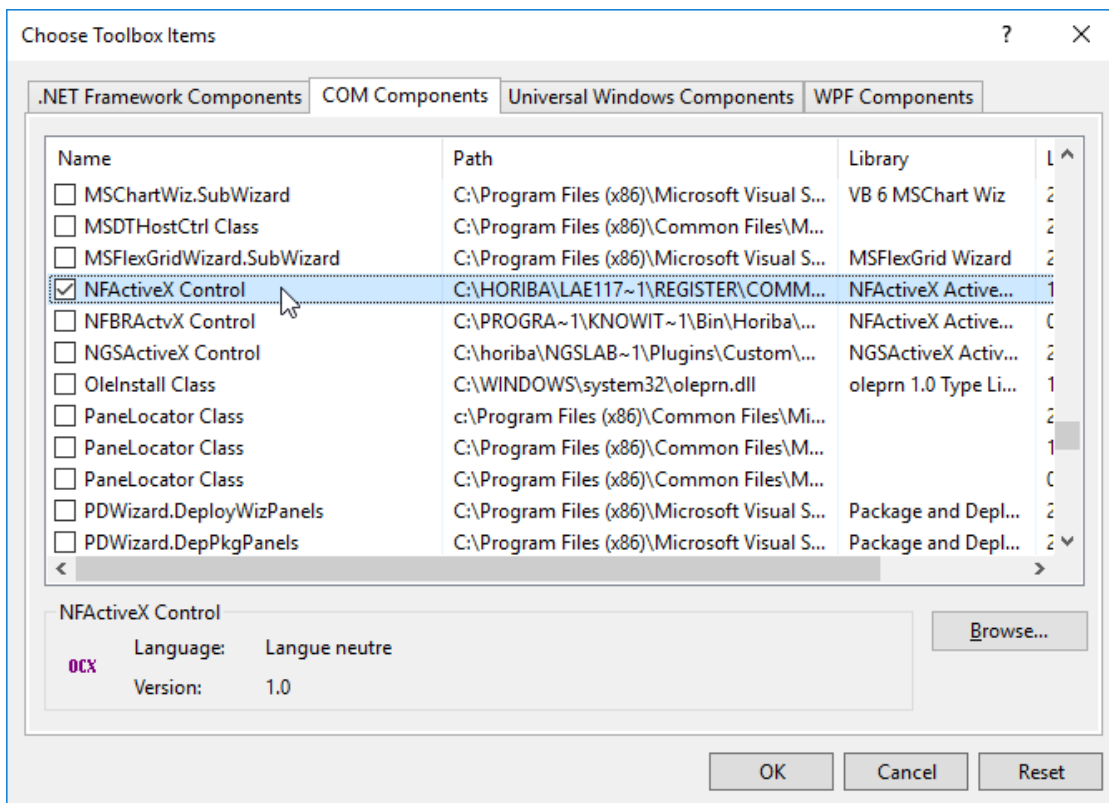
LabSpec.TheFunctionYouNeed(Param1,Param2,...) All the native VBScript functions are also of course available. (except functions that need a window handler such has MsgBox for exemple)

## 2 2 – ActiveX Automation ( in other code editor )

### 2.1 a – Integrating the LabSpec ActiveX



(Visual Basic 6 Example) – Components dialog



(Visual studio 2019) - instantiation of the class automatically  
 the class by code line

(Python 3.7) - instantiation of

See the programming language manual to get how to integrate an ActiveX in your Application.  
 LabSpec 5 ActiveX (NGSActiveX.ocx) or LabSpec 6 ActiveX (NActiveX.ocx) can be imported in a third party application.

## 2.2 b – Programming an application using LabSpec ActiveX

we have 2 rules to observe for the good working of the activeX :

- 1) LabSpec instal in your computer and properly set up for the hardware.
- 2) The activeX must be register in the computer as following video



-COM object integration on a 3rd party software depends on the programming language or host software. See IDE/ host software user manual.

The LabSpec ActiveX must be placed on a form (visible or hidden) to properly work.

For tutorial simplification, the following examples does not includes no error checking, no Synchronization/locking Loop and give a overview of some of the main functions available.

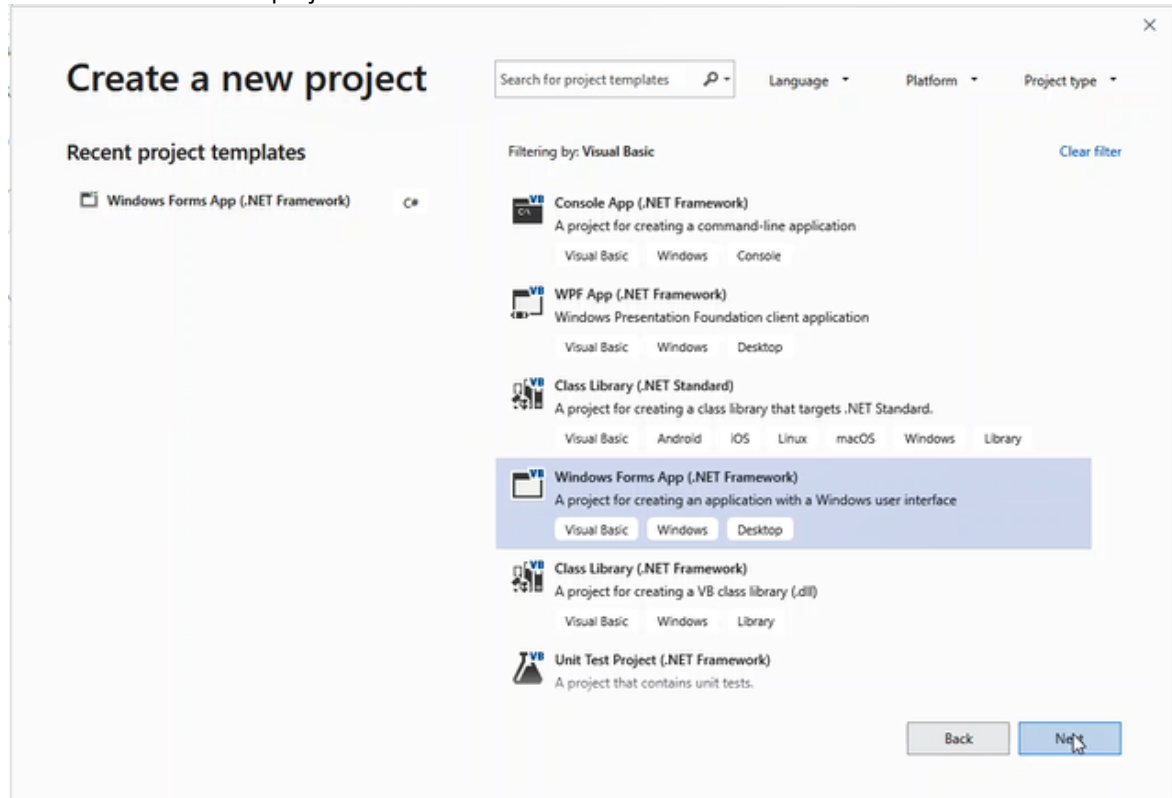
See the ActiveX user manual for an exhaustive functions and parameters description.



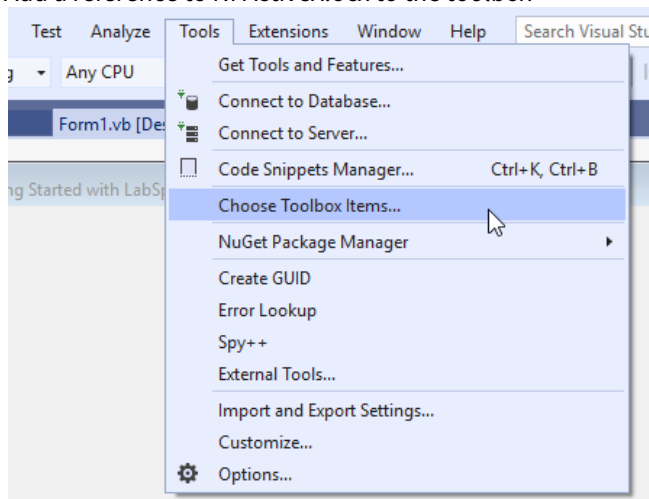
### 3 Getting started - LabSpec Script and ActiveX - .NET Visual Studio 2019

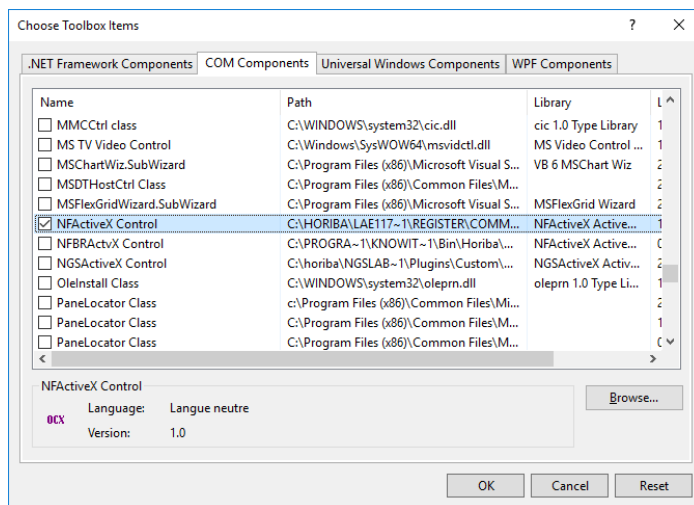
#### Step by step Visual Studio 2019 .NET application using LabSpec6 ActiveX

1. Create a windows Form project

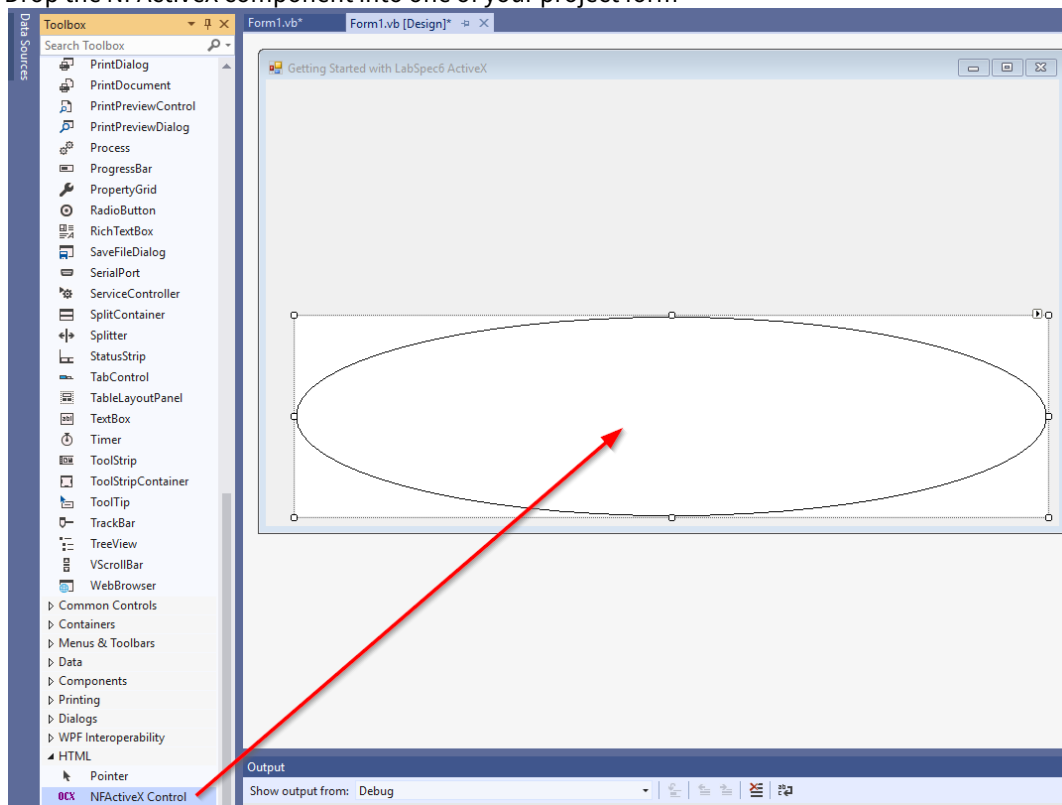


2. Add a reference to NFActiveX.ocx to the toolbox



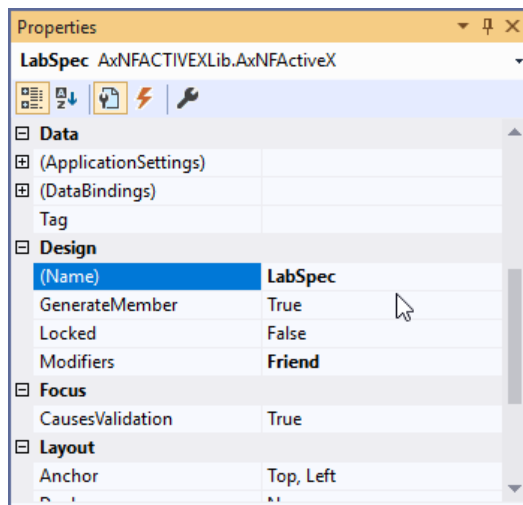


- Drop the NFActiveX component into one of your project form

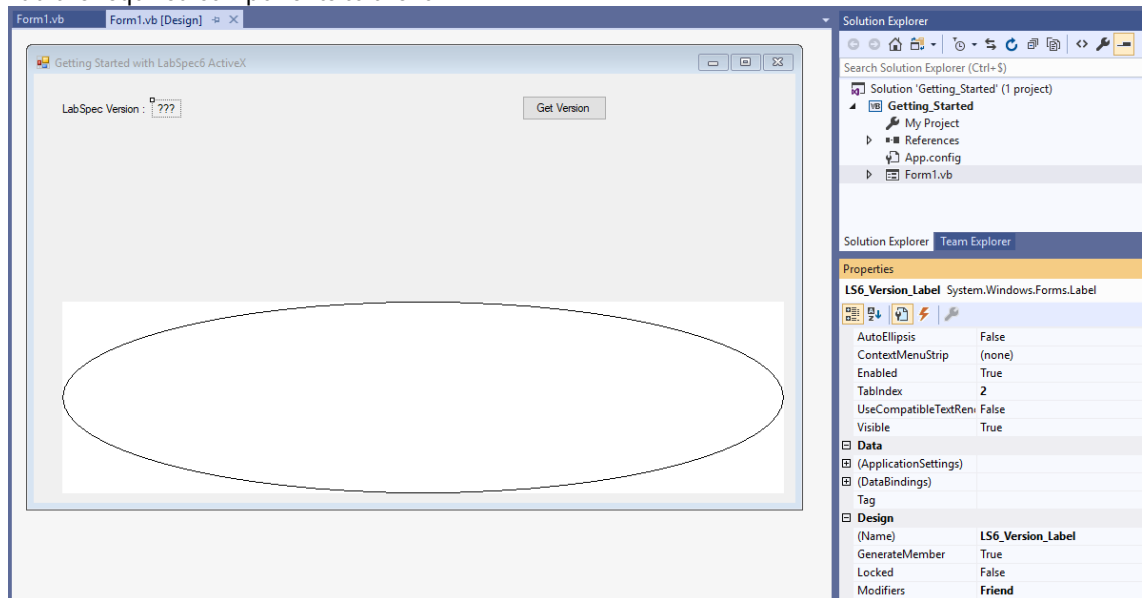


- Optional: In order to be able to use the provided snippets directly, change the NFActiveX component name to "LabSpec", or adjust the snippets with your component name

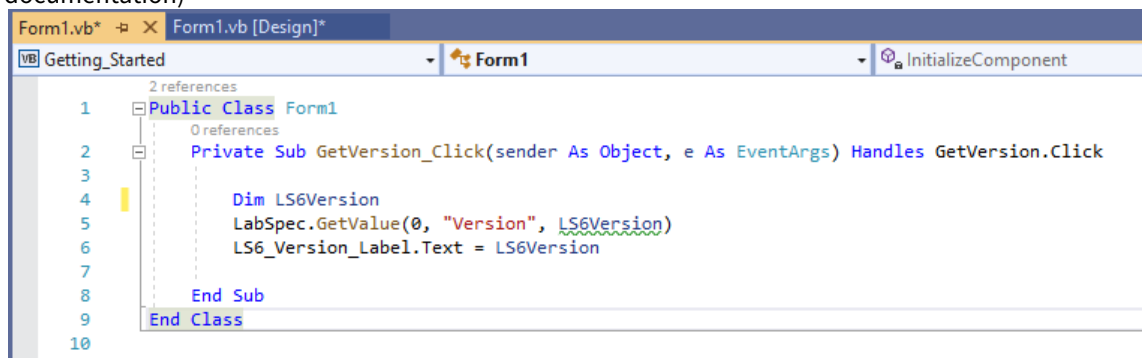
LabSpec functions can now be called directly from your code.



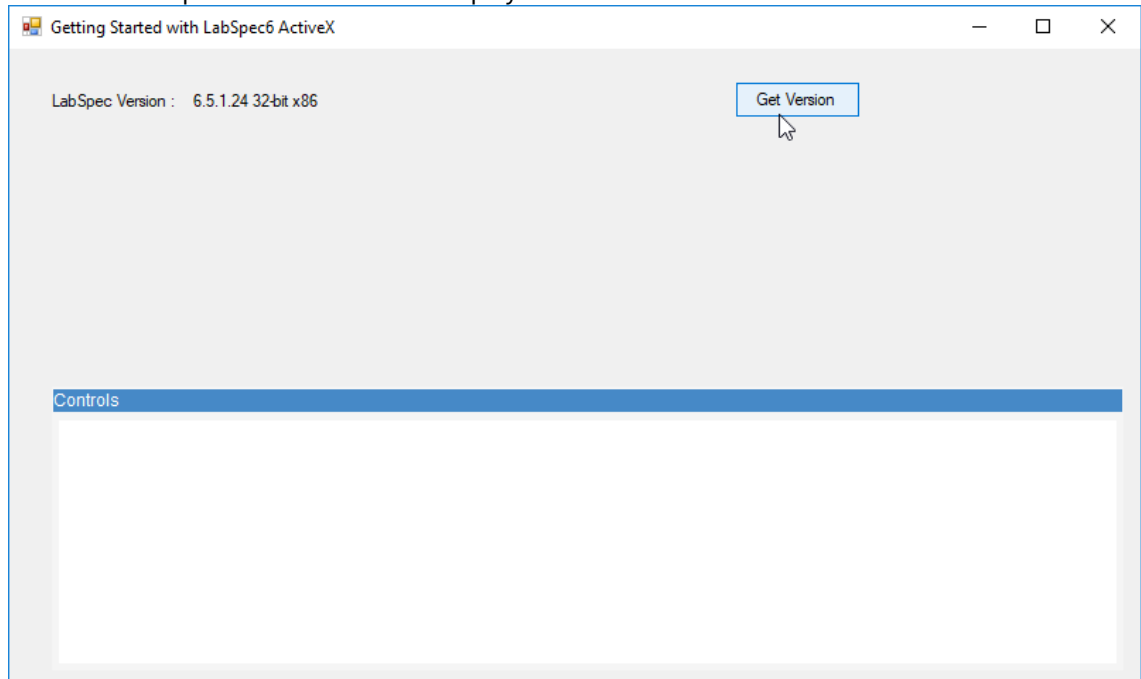
5. Example: get LabSpec6 Active version
  - a. Add the required components to the form



- b. Call the appropriate LabSpec6 function and update the version label on button click (see ActiveX documentation)



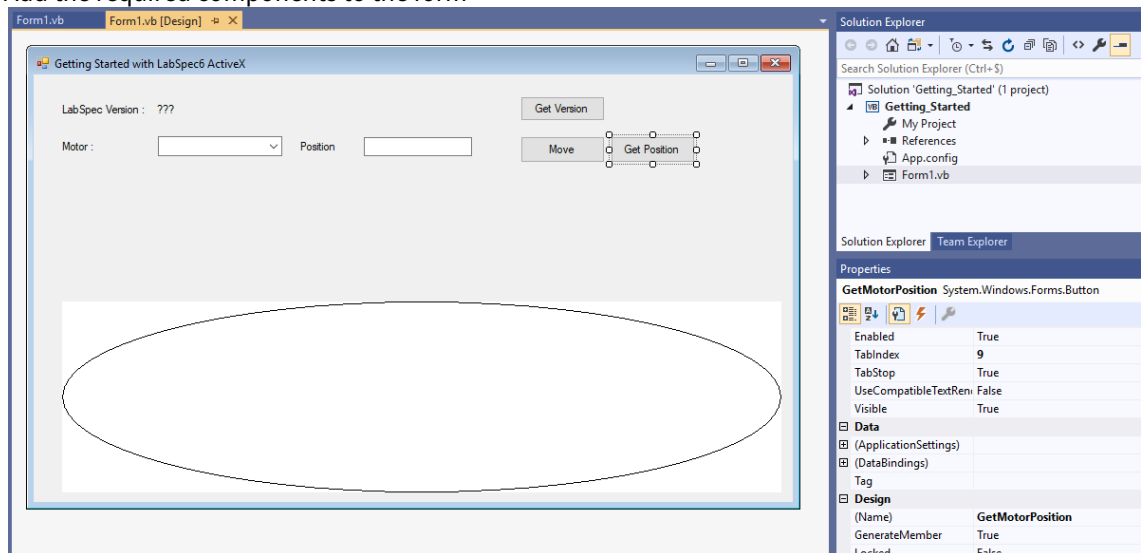
- c. Execution: LabSpec6 ActiveX version is displayed after a click on the "Get Version" button.



Note that the first call to a LabSpec6 function will initialize the ActiveX, and its control.  
The control will display any loaded data (spectra, video..).  
This control can be hidden if LabSpec6 data display is not required for your project.

6. Example: Get the current position and Move an instrument motor

- a. Add the required components to the form



- b. Call the appropriate LabSpec6 function and update the position label on button click (see ActiveX documentation)

O references

```
Private Sub MoveMotor_Click(sender As Object, e As EventArgs) Handles MoveMotor.Click
```

```
    Const MOTOR_VALUE = 0
```

```
    LabSpec.MoveMotor(MotorList.Text, MotorPosition.Text, "", MOTOR_VALUE)
```

```
End Sub
```

O references

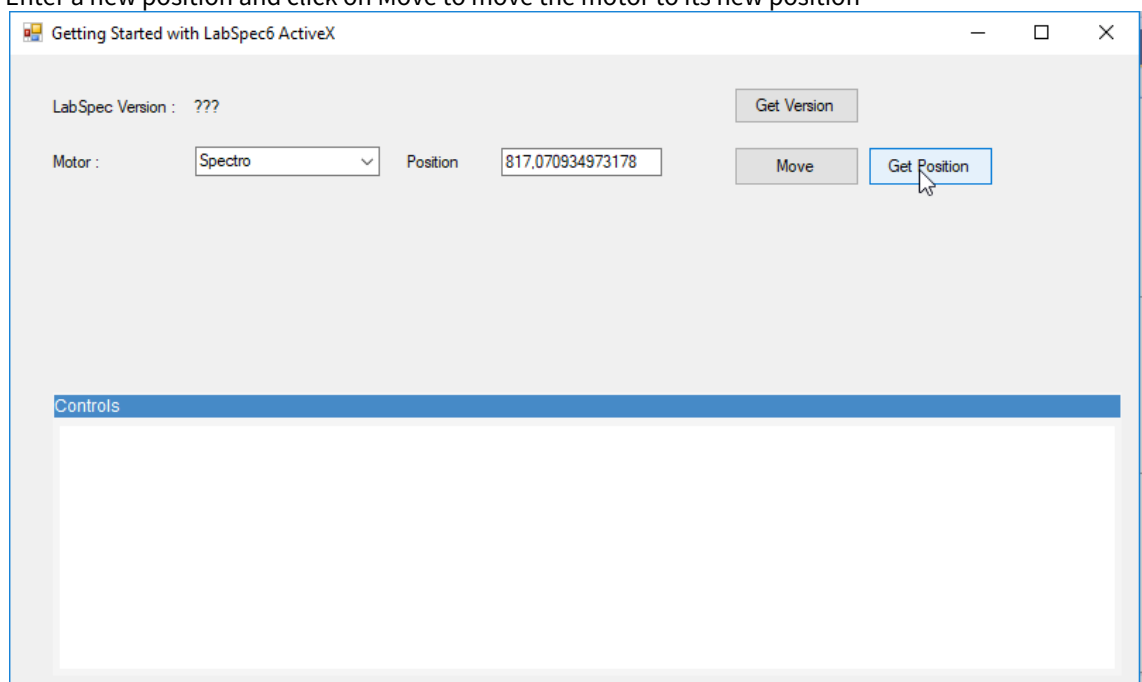
```
Private Sub GetMotorPosition_Click(sender As Object, e As EventArgs) Handles GetMotorPosition.Click
```

```
    Const MOTOR_VALUE = 0
```

```
    MotorPosition.Text = LabSpec.GetMotorPosition(MotorList.Text, MOTOR_VALUE)
```

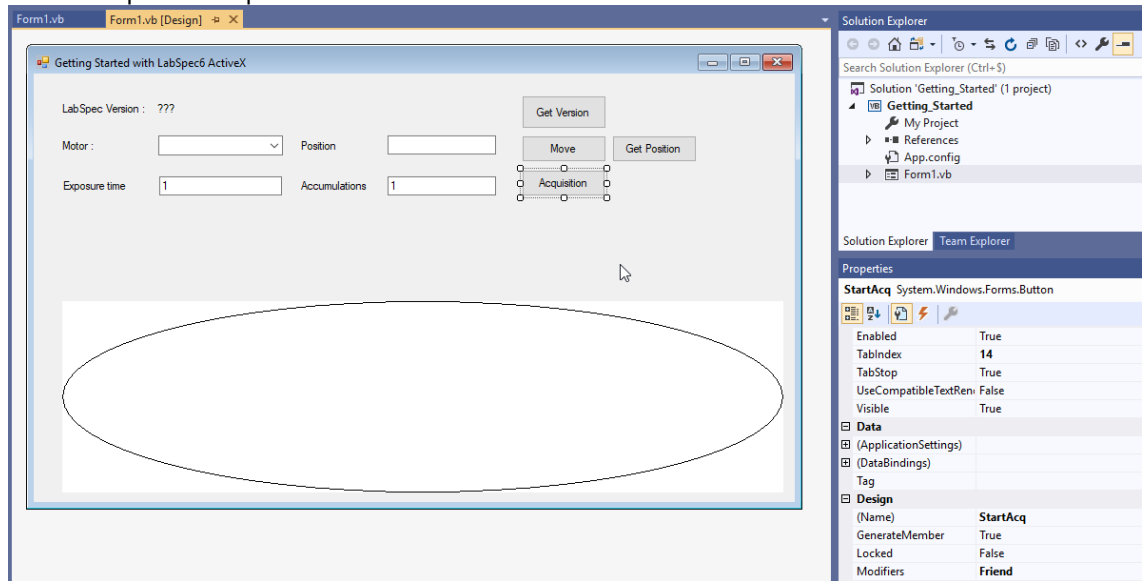
```
End Sub
```

- c. Execution: Select a motor name and click on Get Position to read its current position.  
Enter a new position and click on Move to move the motor to its new position



## 7. Example: Acquire a spectrum

- a. Add the required components to the form



- b. Call the appropriate LabSpec6 function and start an acquisition on button click (see ActiveX documentation)

```

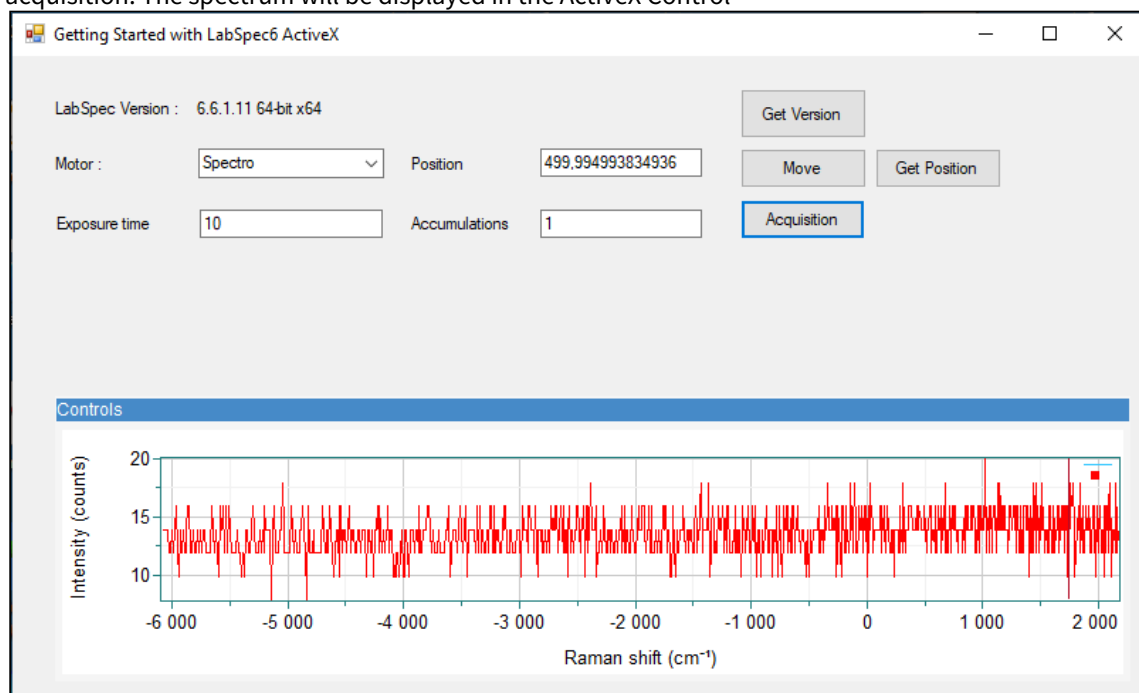
0 references
Private Sub StartAcq_Click(sender As Object, e As EventArgs) Handles StartAcq.Click
    Const ACQ_AUTO_SHOW = 10
    LabSpec.Acq(ACQ_AUTO_SHOW, Expo.Text, Accumulations.Text, 0, 0)
End Sub

```

- c. Execution

Enter an exposure time and number of accumulation. Click on Acquisition to start a spectral

acquisition. The spectrum will be displayed in the ActiveX Control



Project files:

[Getting\\_Started.zip](#)<sup>1</sup>

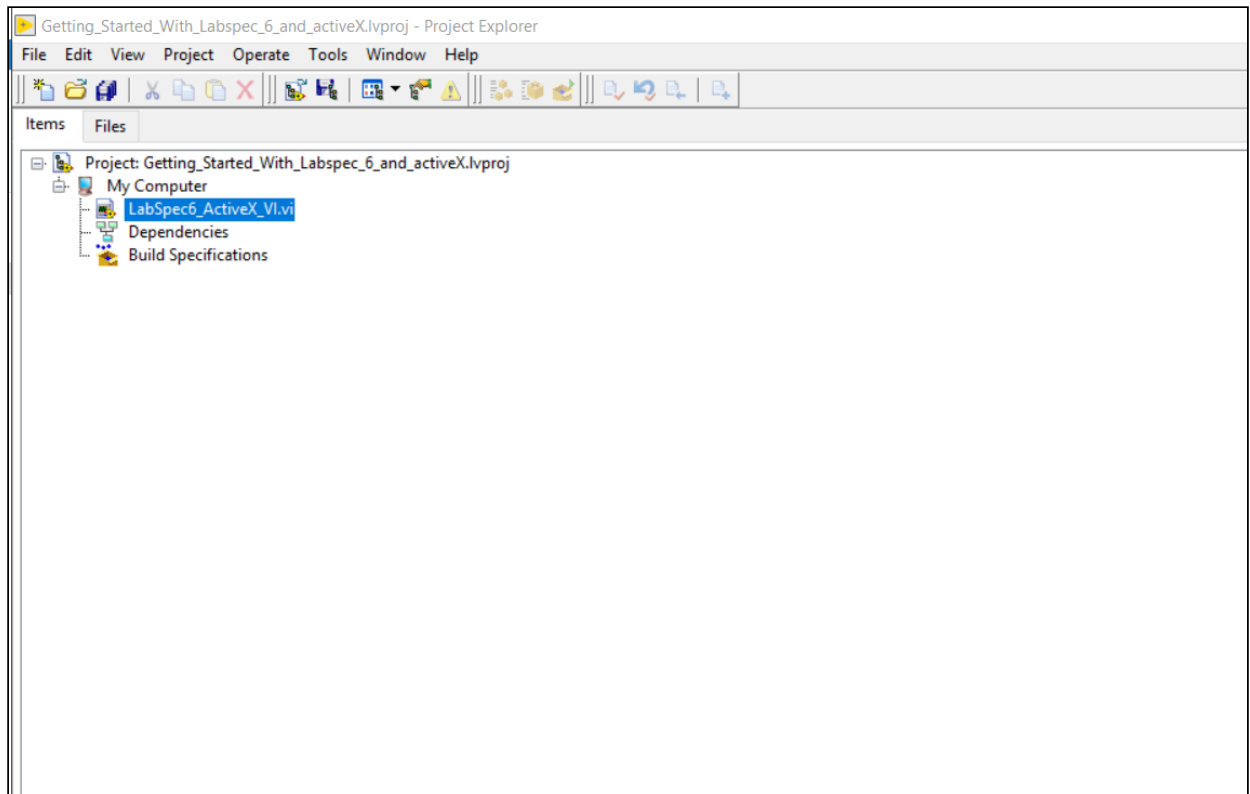
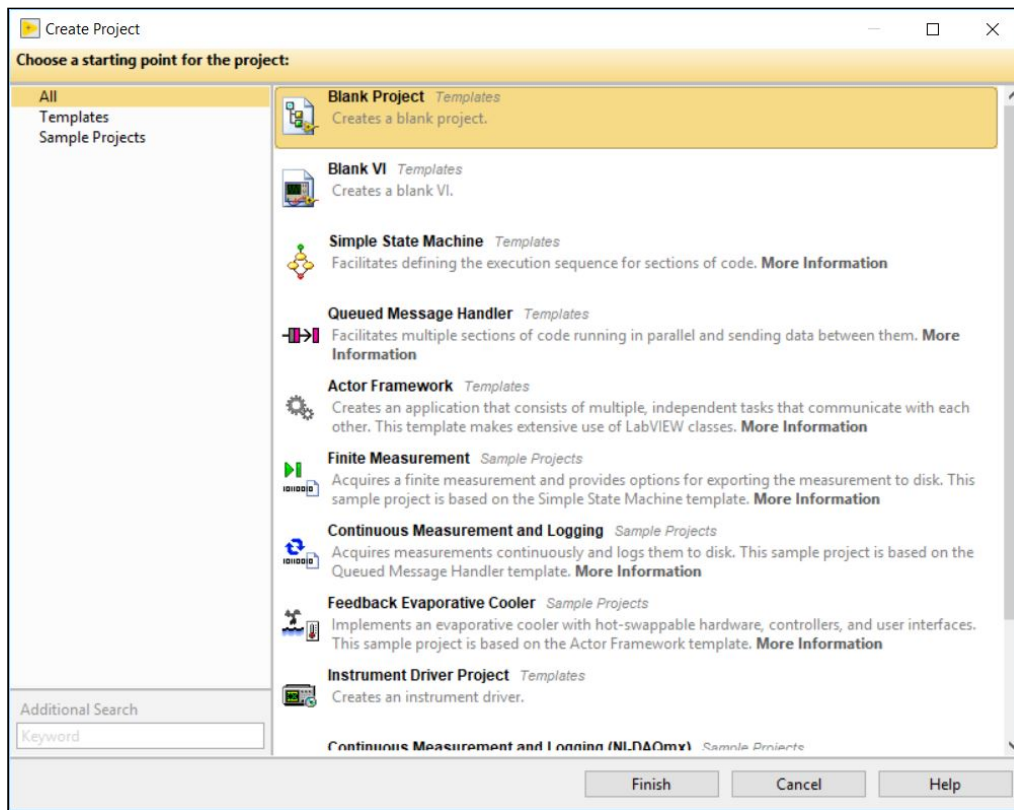
<sup>1</sup> [http://hfratlassian.jy.fr:8090/download/attachments/59342859/Getting\\_Started%20.zip?api=v2&modificationDate=1599540947119&version=1](http://hfratlassian.jy.fr:8090/download/attachments/59342859/Getting_Started%20.zip?api=v2&modificationDate=1599540947119&version=1)

## 4 Getting started - LabSpec Script and ActiveX - Labview 2019

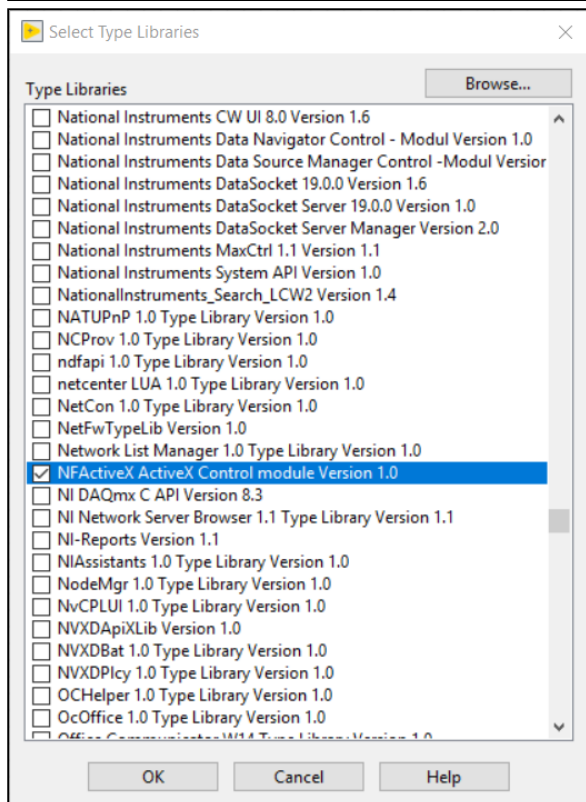
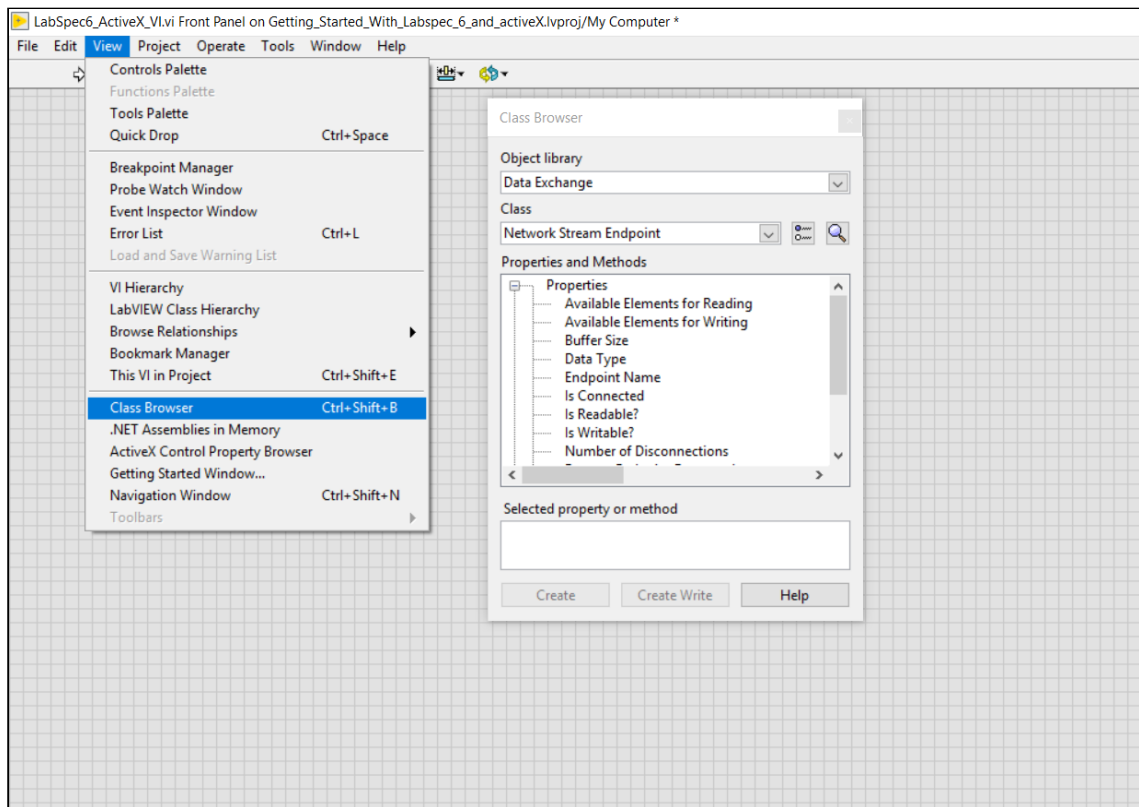
### **Step by step Labview 2019 application using LabSpec6 ActiveX**

1) Create a Blank project and a new VI.

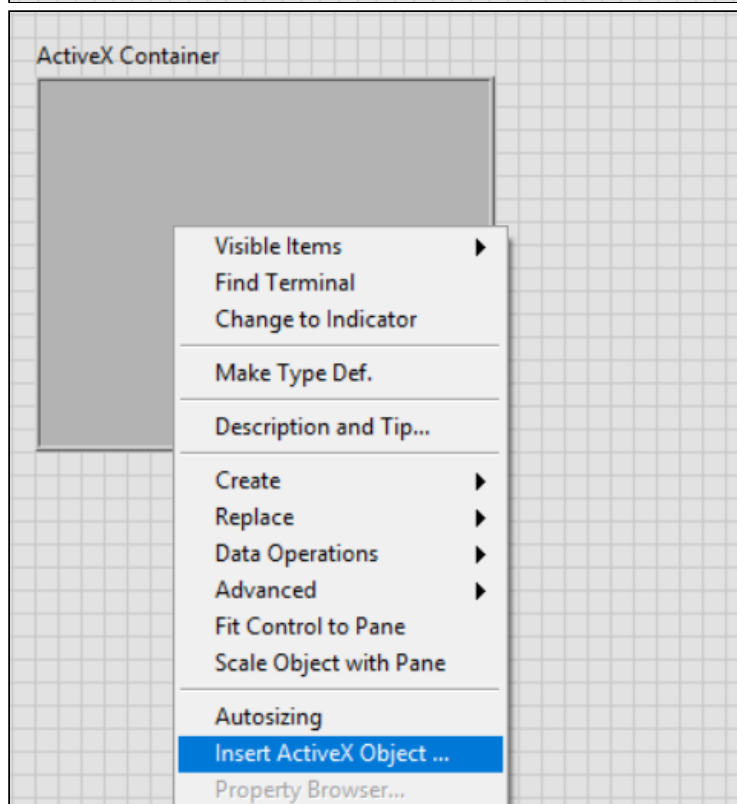
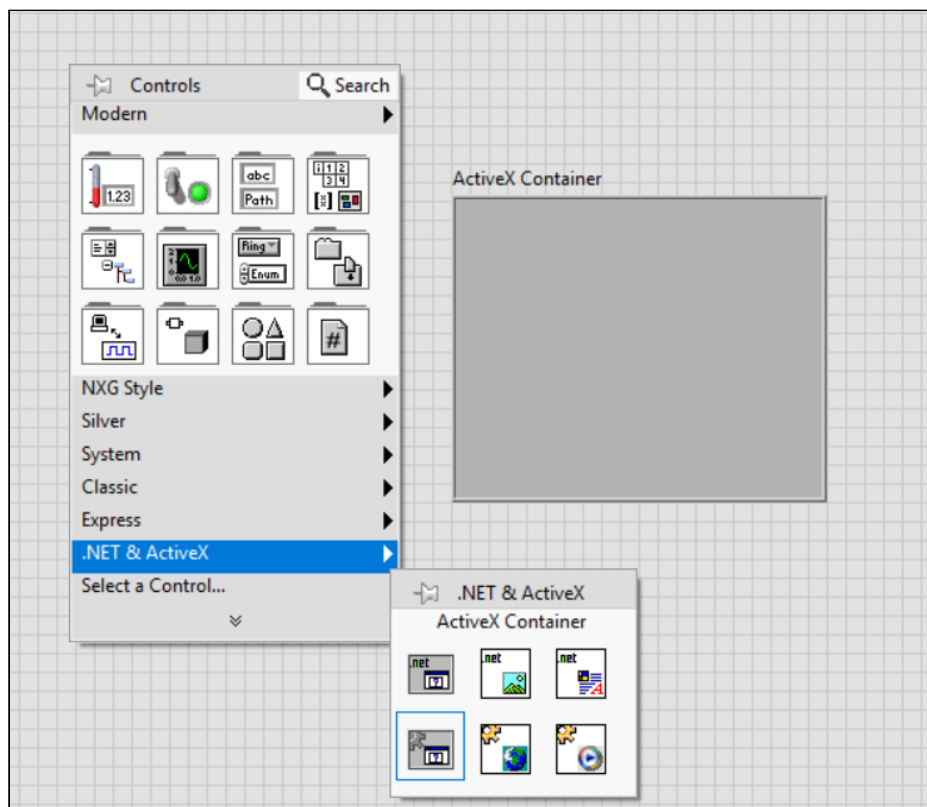


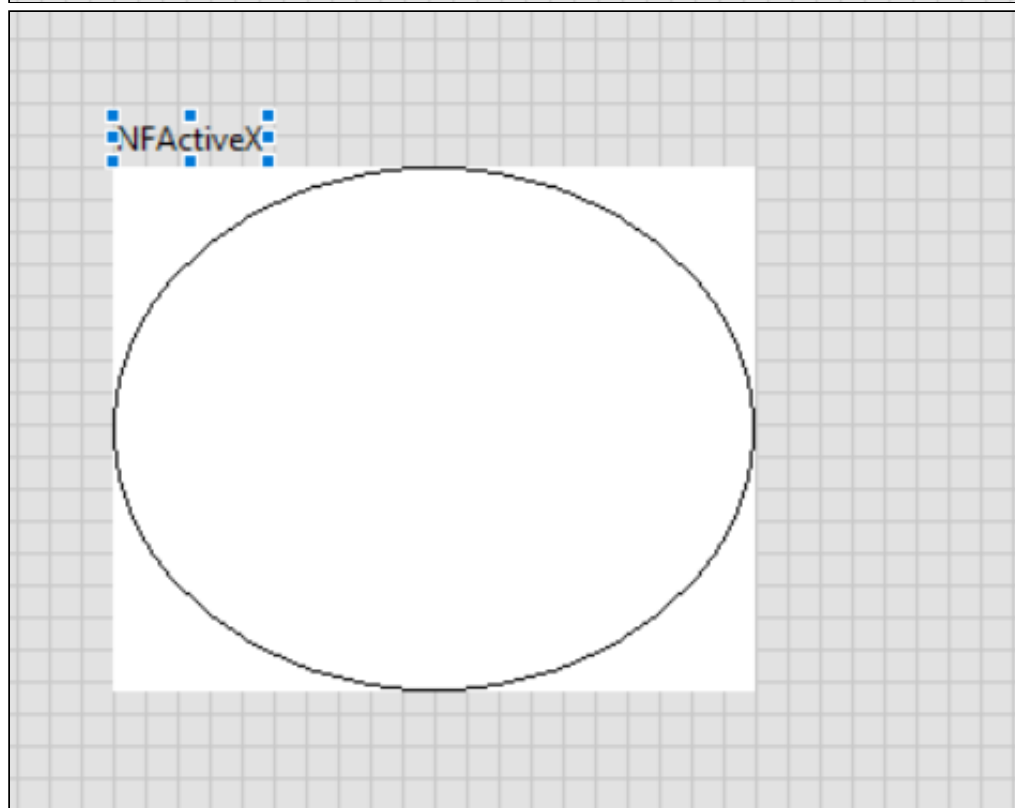
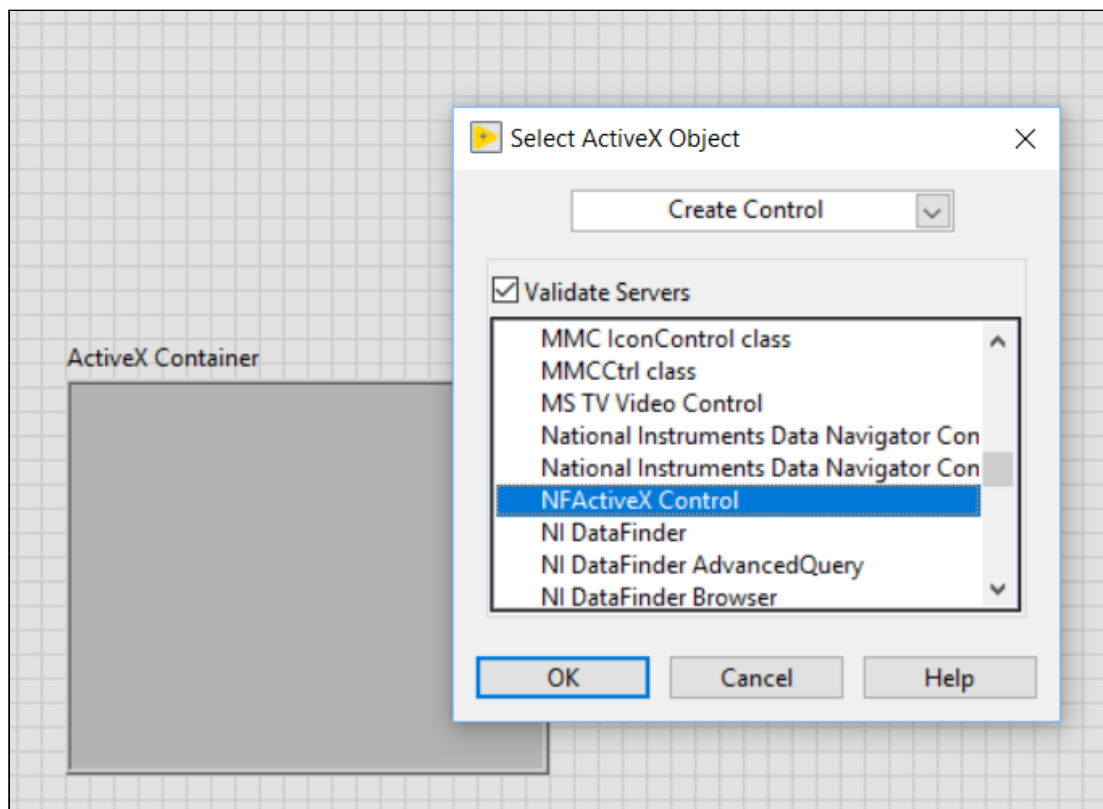


2) Add a reference to NFActiveX.ocx to the project.



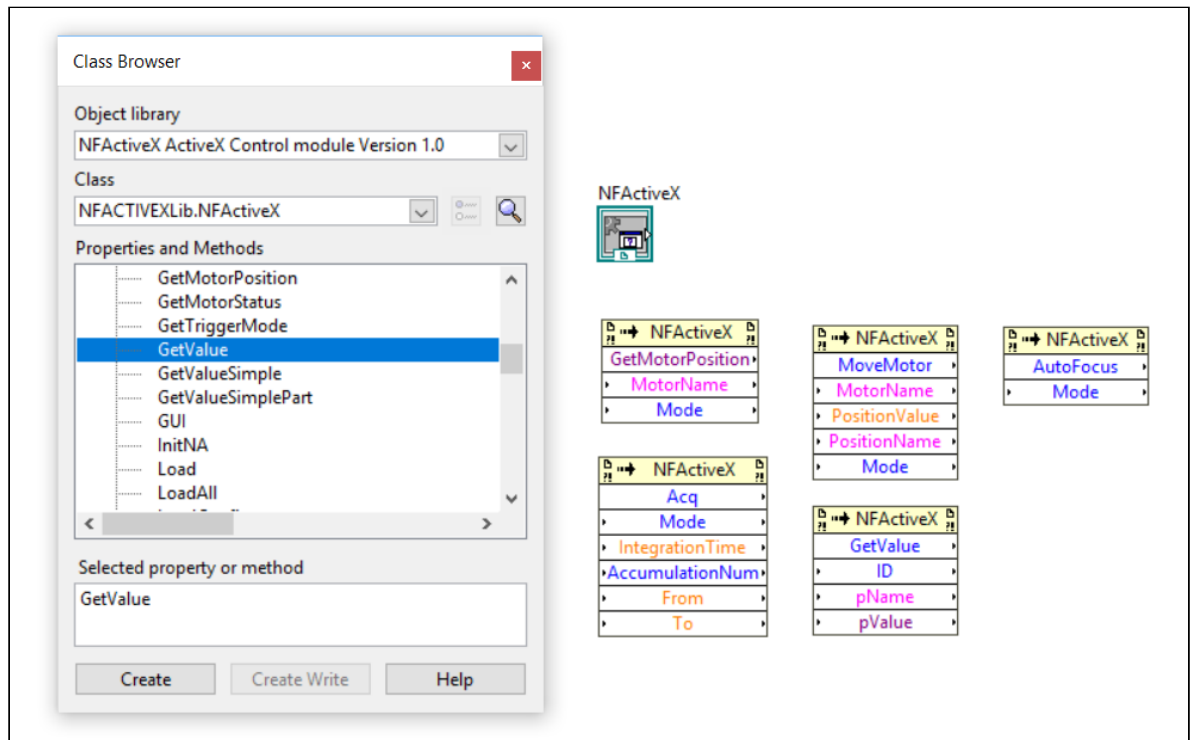
3) Add the NFActiveX component into the Front Panel.





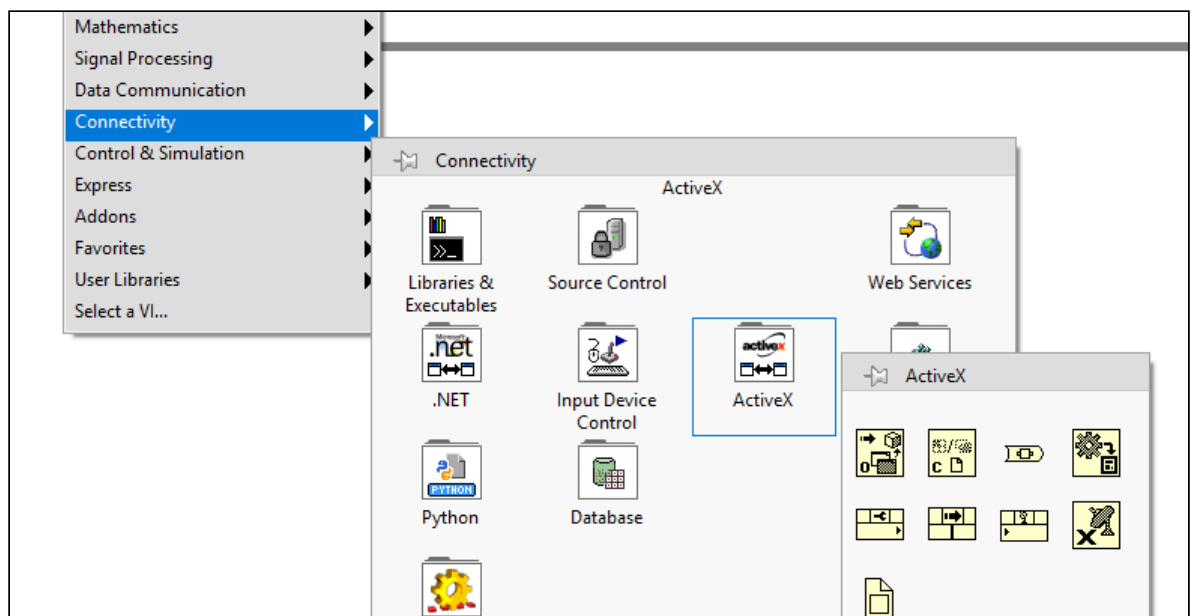
4) Add the NFActiveX Method component into the block diagram.

note : the Class browser will provide you all the function you can call from LabSpec6 (Drag and drop is available).

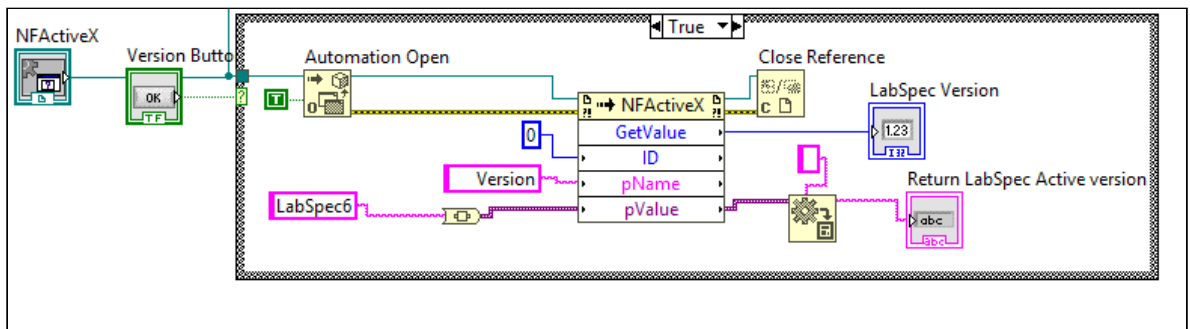


5)Example: get LabSpec6 Active version :

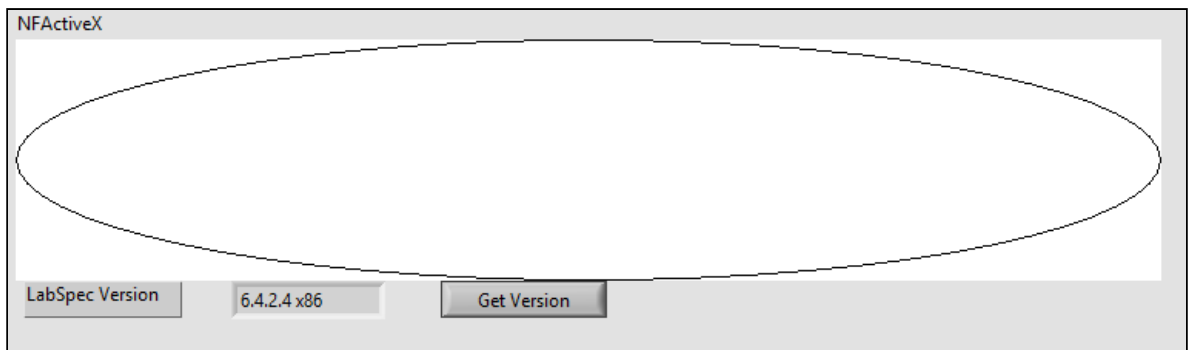
a)Add the required components to the Block Diagram and the Front Panel:



b)Call the appropriate LabSpec6 Method and update the version label on Get Version button (see ActiveX documentation)



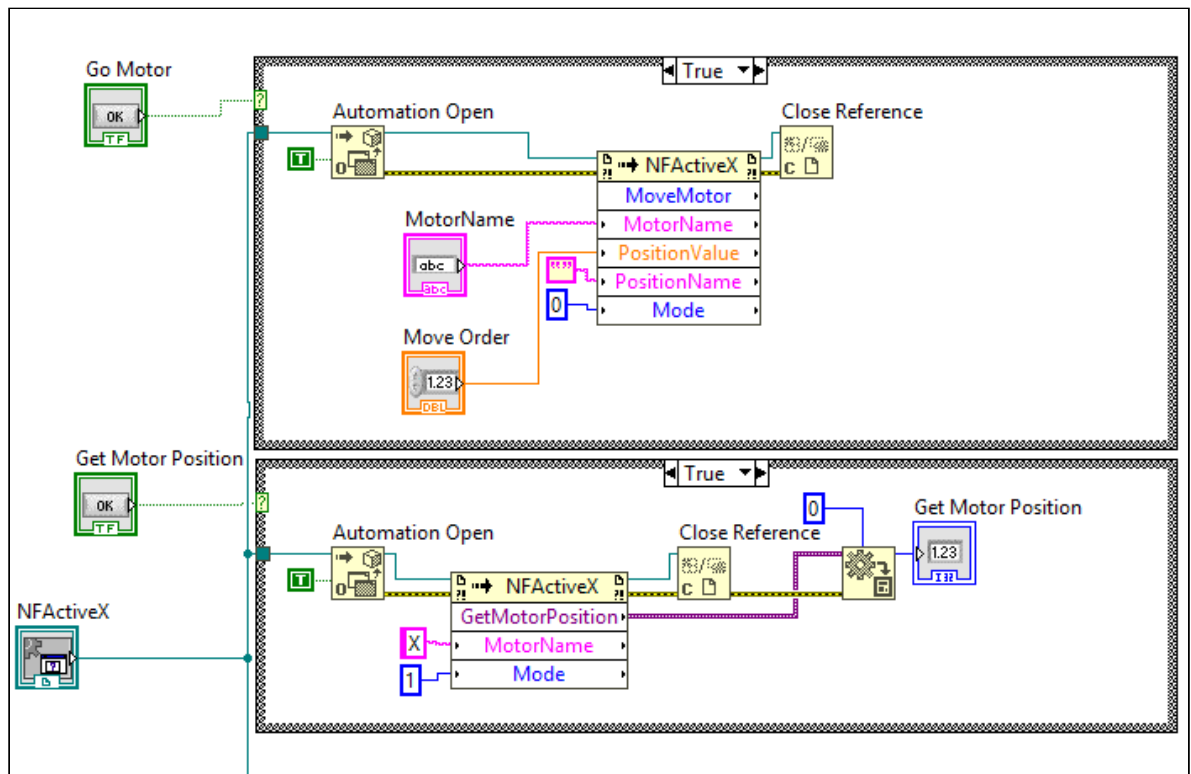
c)Execution: LabSpec6 ActiveX version is displayed after a click on the "Get Version" button.



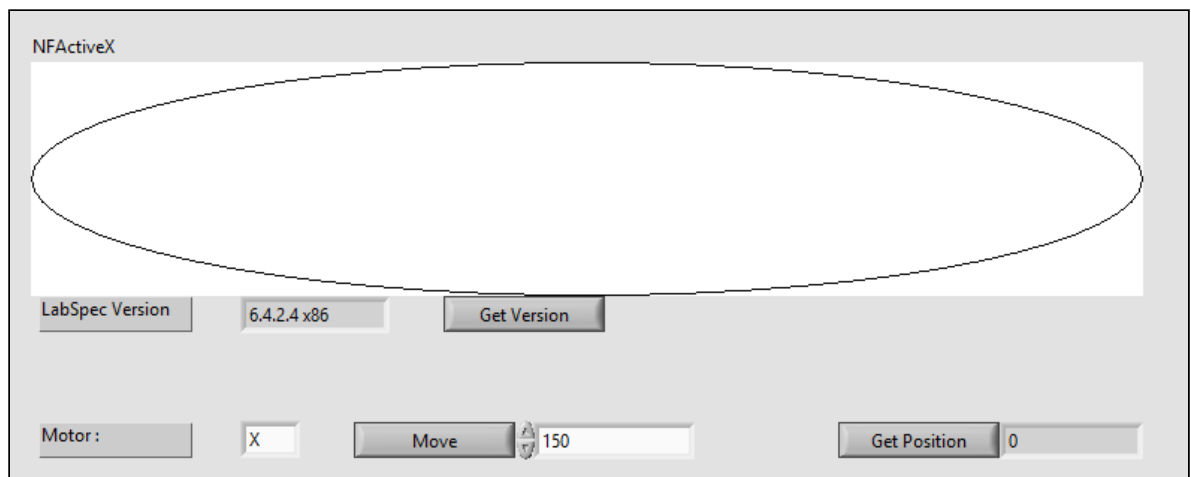
Note that the first call to a LabSpec6 function will initialize the ActiveX, and its control.  
The control will display any loaded data (spectra, video..).  
This control can be hidden if LabSpec6 data display in not required for your project.

6)Example: Get the current position and Move an instrument motor

a)Add the required components to the Block Diagram and the Front Panel:



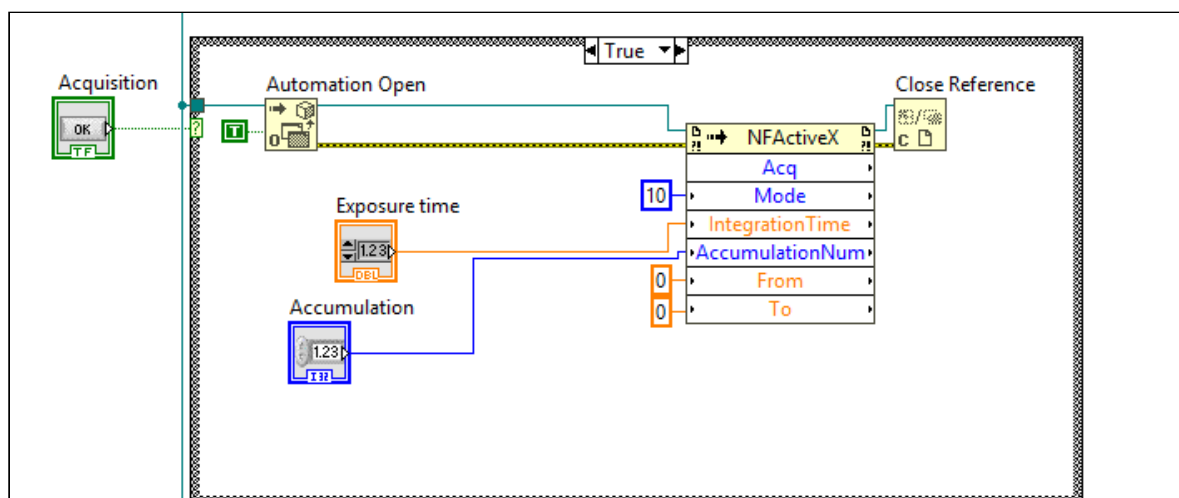
b) Call the appropriate LabSpec6 Method and update the position label on button click (see ActiveX documentation)



Execution: Select a motor name and click on Get Position to read its current position.  
Enter a new position and click on Move to move the motor to its new position

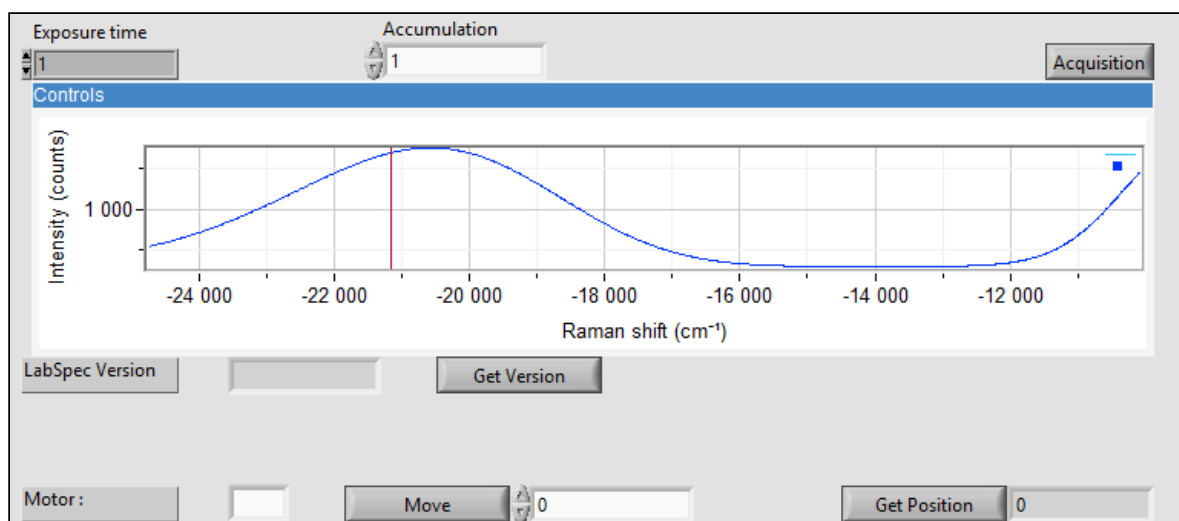
7) Example: Acquire a spectrum

a) Add the required components to the Block Diagram and the Front Panel:



b) Call the appropriate LabSpec6 Method and update the acquisition button click (see ActiveX documentation)

Enter an exposure time and number of accumulation. Click on Acquisition to start a spectral acquisition. The spectrum will be displayed in the ActiveX Control



Project files:

[Labview2019.zip](#)<sup>2</sup>

<sup>2</sup> <http://hfratlassian.jy.fr:8090/download/attachments/59342880/Labview2019.zip?api=v2&modificationDate=1598617442244&version=1>

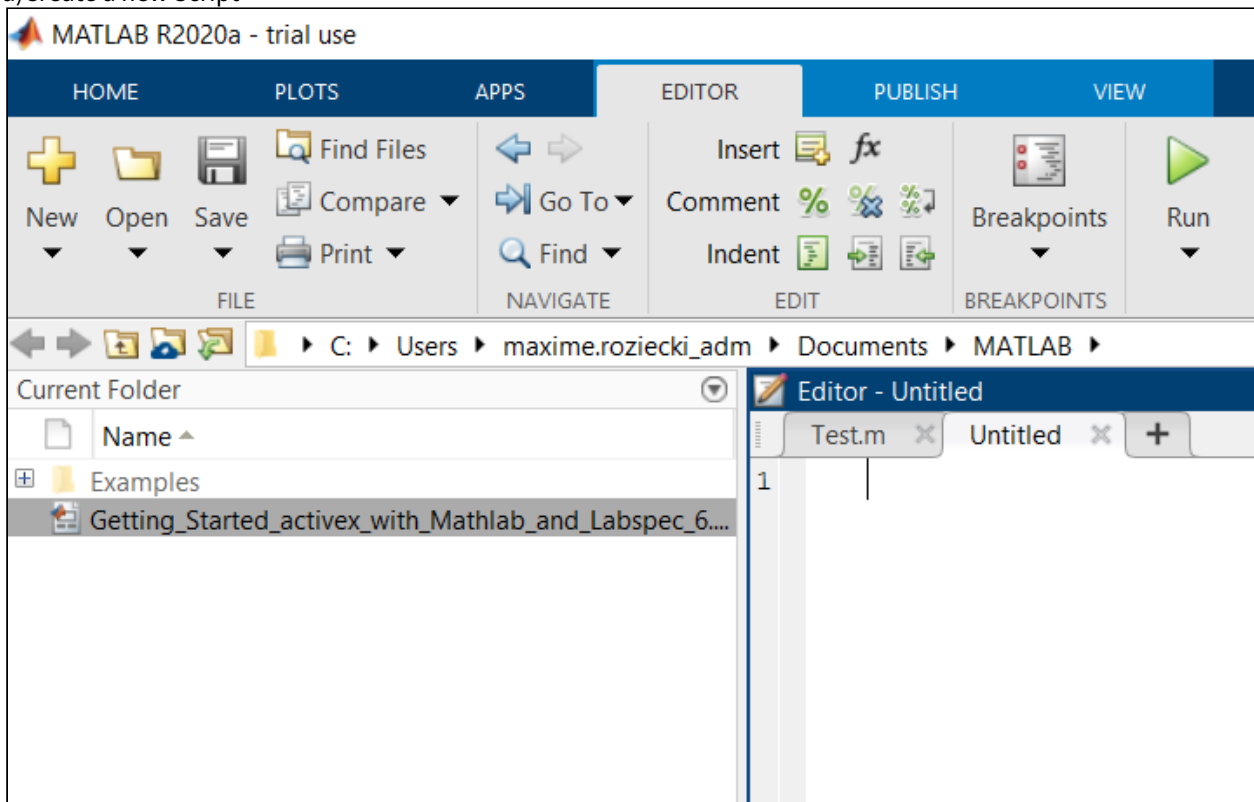


## 5 Getting started - LabSpec Script and ActiveX - Matlab R2020a

- ❗ -Exist only in x64
- Not able to display OLE component

### 1. Step by step MATLAB R2020a application using LabSpec6 ActiveX

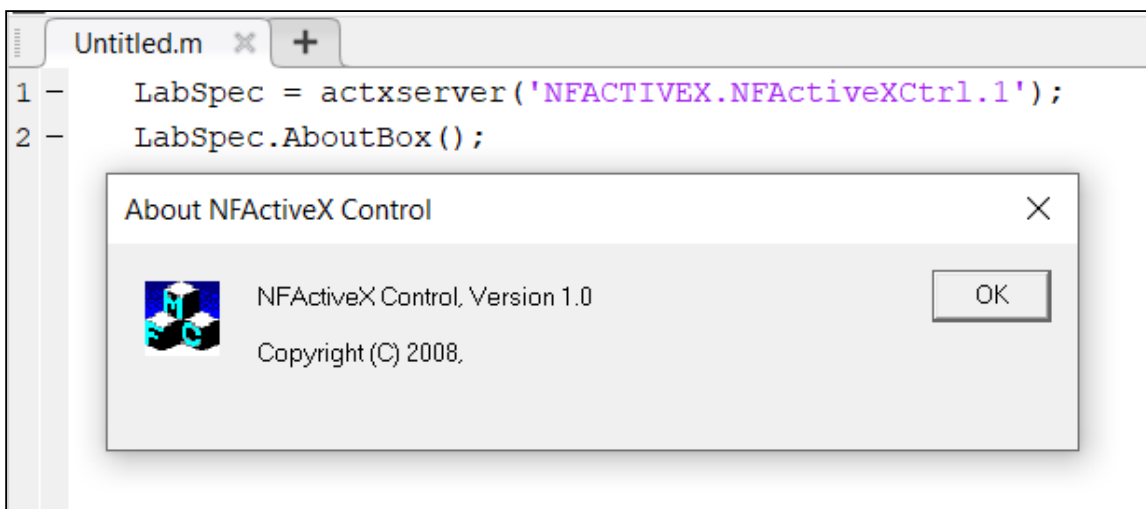
a) Create a new Script



b) Add a reference to NFActiveX.ocx to the Editor

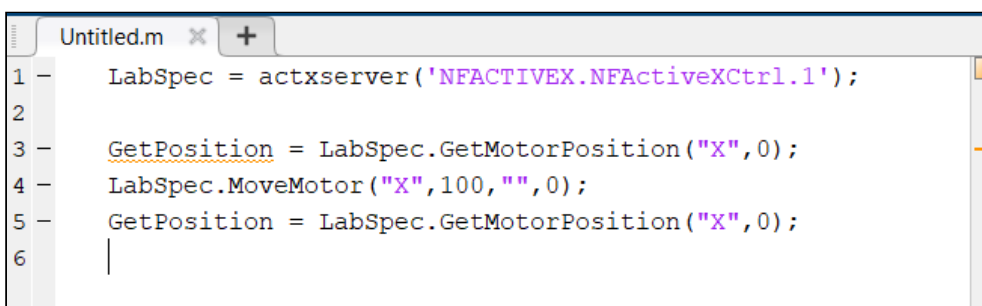


c) check if the ActiveX work :



2)Example : Get the current position and Move an instrument motor

a) Call the appropriate LabSpec6 function and update the position(see ActiveX documentation)



b)Execution Select a motor name and a new position , you can see the update in workspace

Workspace	
Name ^	Value
ans	0
GetPosition	120
LabSpec	1x1 NFACTIVEX_...
Move	0
test_999	494549203

### 3)Example: Acquire a spectrum

a)Call the appropriate LabSpec6 function and display the data

Untitled.m
 

9 -

ACQ\_AUTO\_SHOW = 10;

10 -

Exposure =1;

11 -

Accumulations=1;

12 -

LabSpec.Acq(ACQ\_AUTO\_SHOW,Exposure,Accumulations,0,0);

13

%Get the spectrum ID

14 -

SpectrumID=LabSpec.GetAcqID();

15

%load and display the data

16 -

Spectrumvalue=LabSpec.GetValueSimple(SpectrumID,"XYData",0,5);

17 -

Dx=[];

18 -

Dy=[];

19 -

Dxd=[];

20 -

Dyd=[];

21 -

Dv=[1,2,3];

22 -

Dx=Spectrumvalue(1,1:end);

23 -

Dy=Spectrumvalue(2,1:end);

24 -

Dxd=cell2mat(Dx);

25 -

Dyd=cell2mat(Dy);

26 -

plot(Dxd,Dyd,'-');

27 -

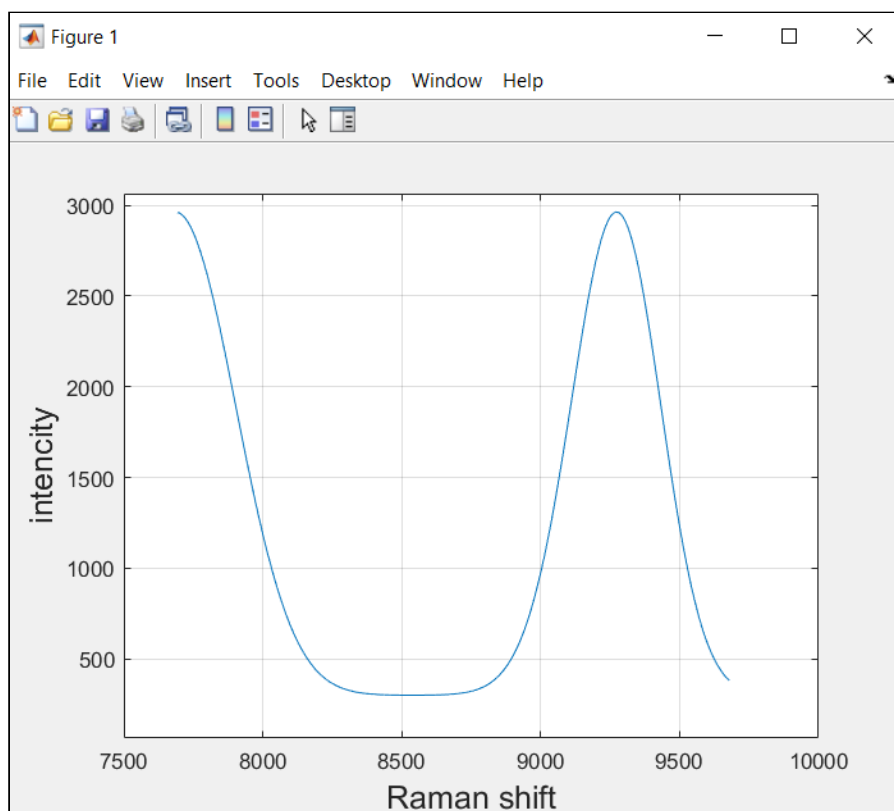
grid on;

28 -

xlabel('Raman shift','FontSize',15);

Name ^	Value
Accumulations	1
ACQ_AUTO_SHOW	10
ans	0
Dv	[1,2,3]
Dx	1x1024 cell
Dxd	1x1024 double
Dy	1x1024 cell
Dyd	1x1024 double
Exposure	1
GetPosition	390
LabSpec	1x1 NFACTIVEX_...
SpectrumID	12930
Spectrumvalue	2x1024 cell
tf	1

b)Enter an exposure time and number of accumulation. Start a spectral acquisition. The spectrum will be displayed in the plot



Project files :

[LS6ActiveX.zip](#)<sup>3</sup>

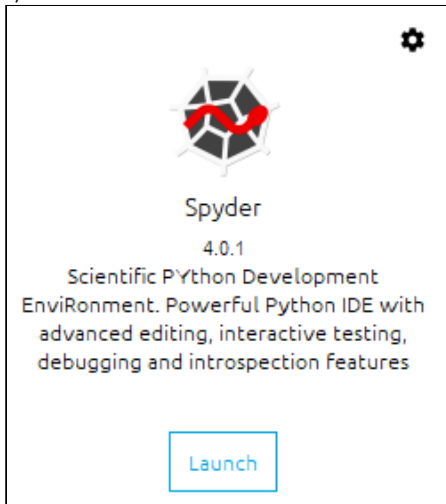
---

<sup>3</sup> <http://hfratlassian.jy.fr:8090/download/attachments/60063765/LS6ActiveX.zip?api=v2&modificationDate=1600142540567&version=1>

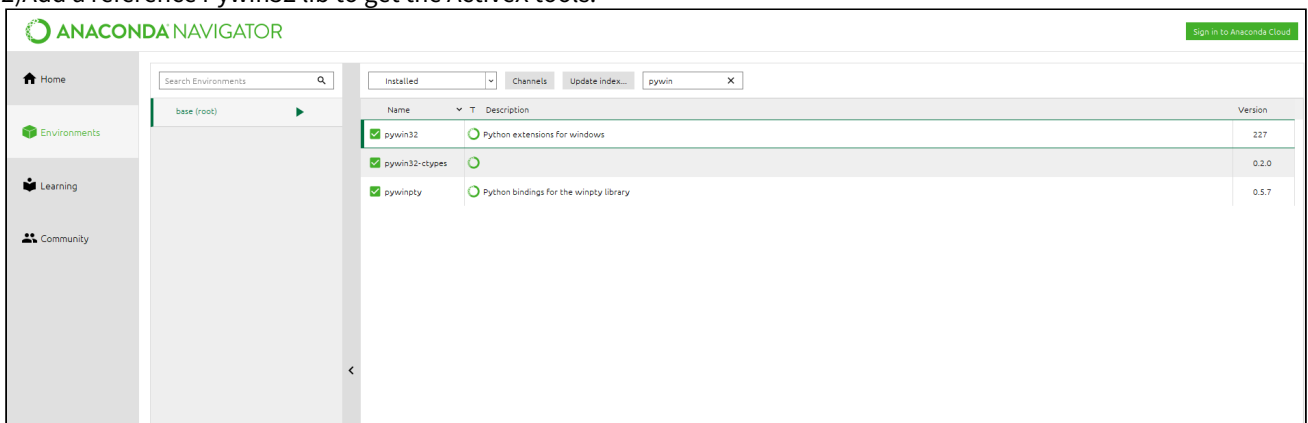
## 6 Getting started - LabSpec Script and ActiveX - Python 3.7- Anaconda 3 - Spyder 4.0.1 IDE

### Step by step Python application using LabSpec6 Active X

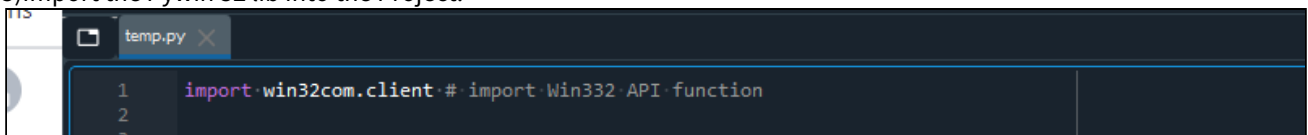
1)DL ANACONDA NAVIGATOR and create a Python Project in Spyder.



2)Add a reference Pywin32 lib to get the ActiveX tools.



3)Import the Pywin 32 lib into the Project.

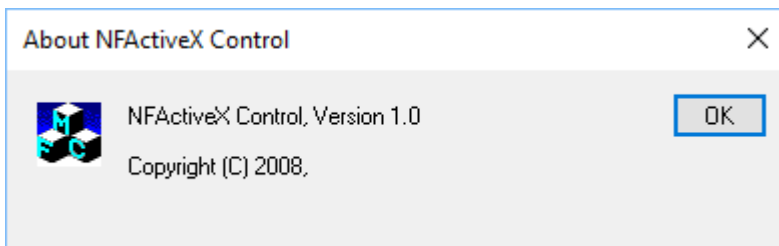


4)In order to be able to use the LS6 controls declared the labSpec6 ActiveX component

```
#Declaration of the LabSpec6 ActiveX Component
LabSpec = win32com.client.Dispatch("NFACTIVEX.NFACTiveXCtrl.1")
#Test if the activeX control is Enable
LabSpec.AboutBox()
```

=>Note that the first call to a LabSpec6 function will initialize the ActiveX, and its control.

5)LabSpec functions can now be called directly from your code.

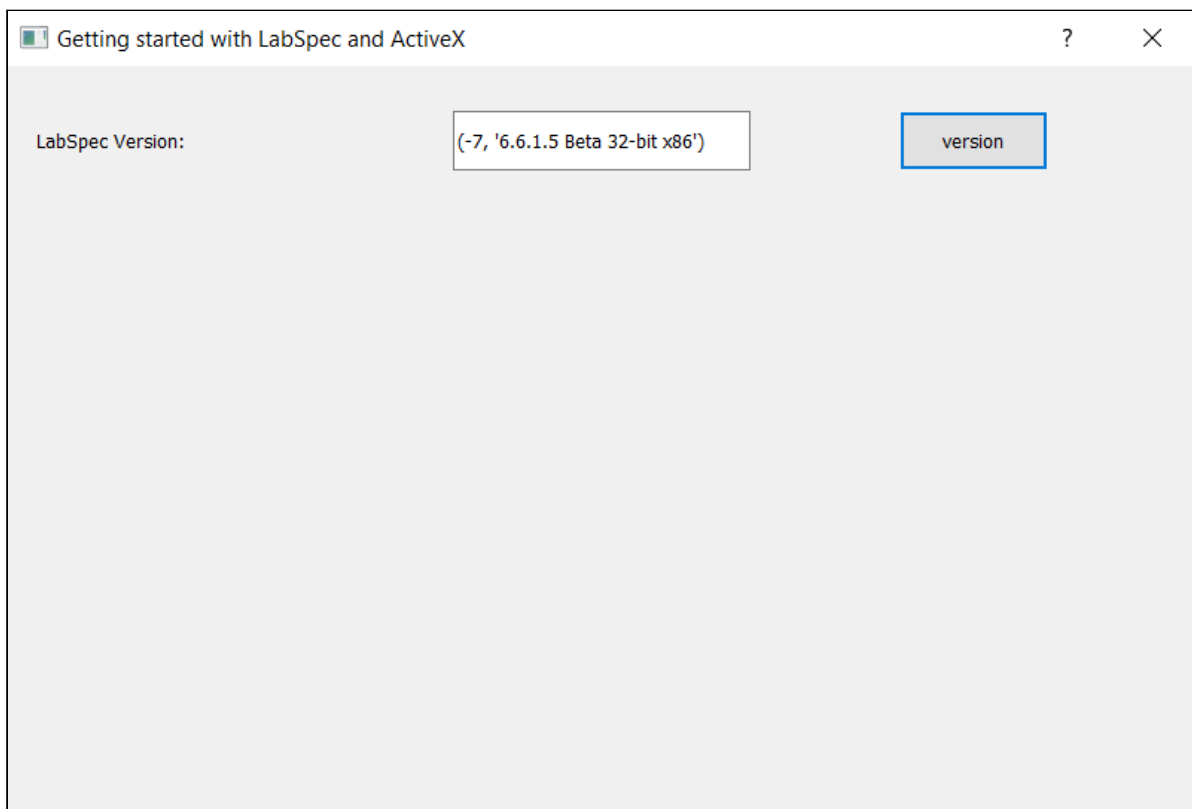


6)Example: get LabSpec6 Active version

a)Call the appropriate LabSpec6 function and update the version label on button click (see ActiveX documentation)

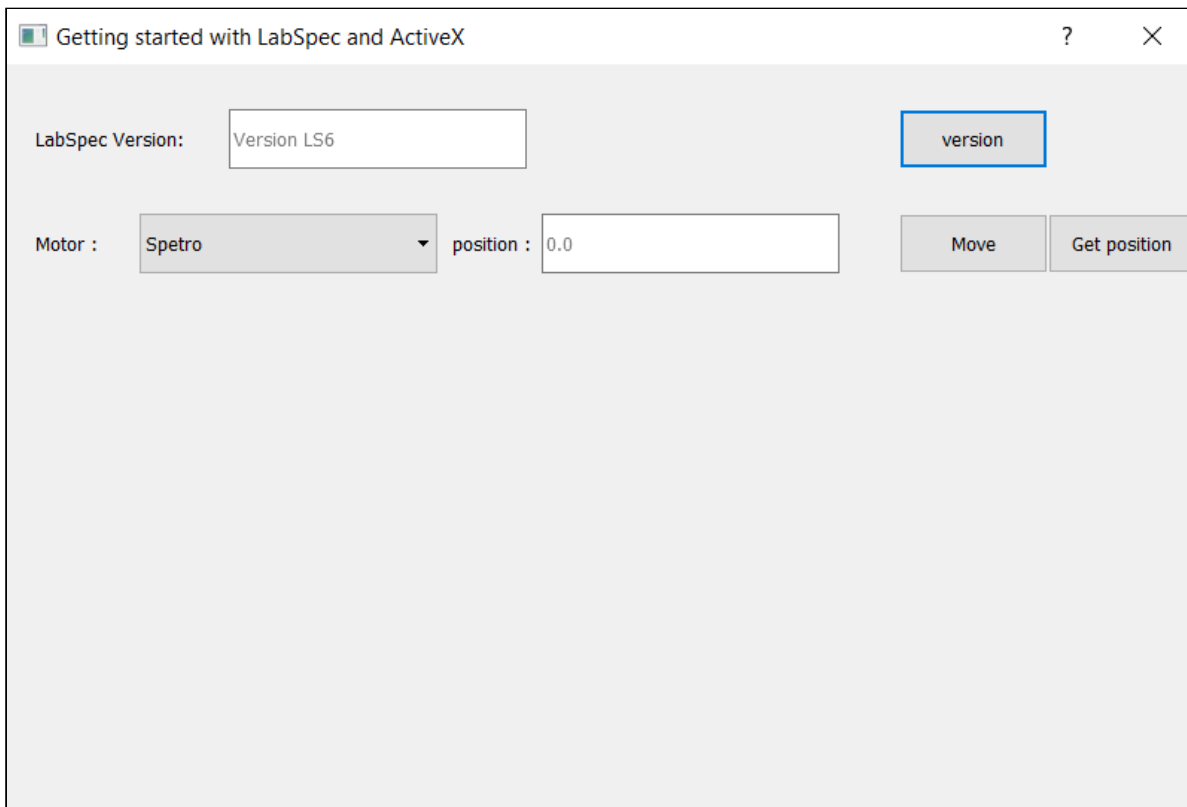
```
.....#Buttongetversion event handler
.....def on_click_version(self):
.....#Get the LS6 current version
.....LS6Version = ""
.....LS6Version = LabSpec.GetValue(0,"Version",LS6Version)
.....self.textbox_Version.setText(str(LS6Version))
```

b)Execution: LabSpec6 ActiveX version is displayed after a click on the "Version" button.



7)Example: Get the current position and Move an instrument motor.

a)Add the required components to the form



b) Call the appropriate LabSpec6 function and update the position label on button click (see ActiveX documentation)

```
.....#Buttongetposition event handler
.....def on_click_position(self):
.....    pos = LabSpec.GetMotorPosition(Axis,0);
.....    self.textbox_Position.setText(str(pos))
.....
.....#Combo event handler
.....def selectionchange(self,i):
.....    Axis = self.cb.currentText()
.....
```

**c) Execution: Select a motor name and click on Get Position to read its current position.**

Enter a new position and click on Move to move the motor to its new position



Getting started with LabSpec and ActiveX

LabSpec Version: (-7, '6.6.1.5 Beta 64-bit x64') version

Motor : Spectro position : 100.00138802318091 Move Get position

Spectro  
X  
Y  
Z

8Example: Acquire a spectrum

a)Add the required components to the form

Getting started with LabSpec and ActiveX

LabSpec Version: Version LS6 version

Motor : Spectro position : 0.0 Move Get position

Exposure time : 1 Accumulations : 1 Acquisition

0 0.2 0.4 0.6 0.8 1

b) Call the appropriate LabSpec6 function and start an acquisition on button click (see ActiveX documentation)

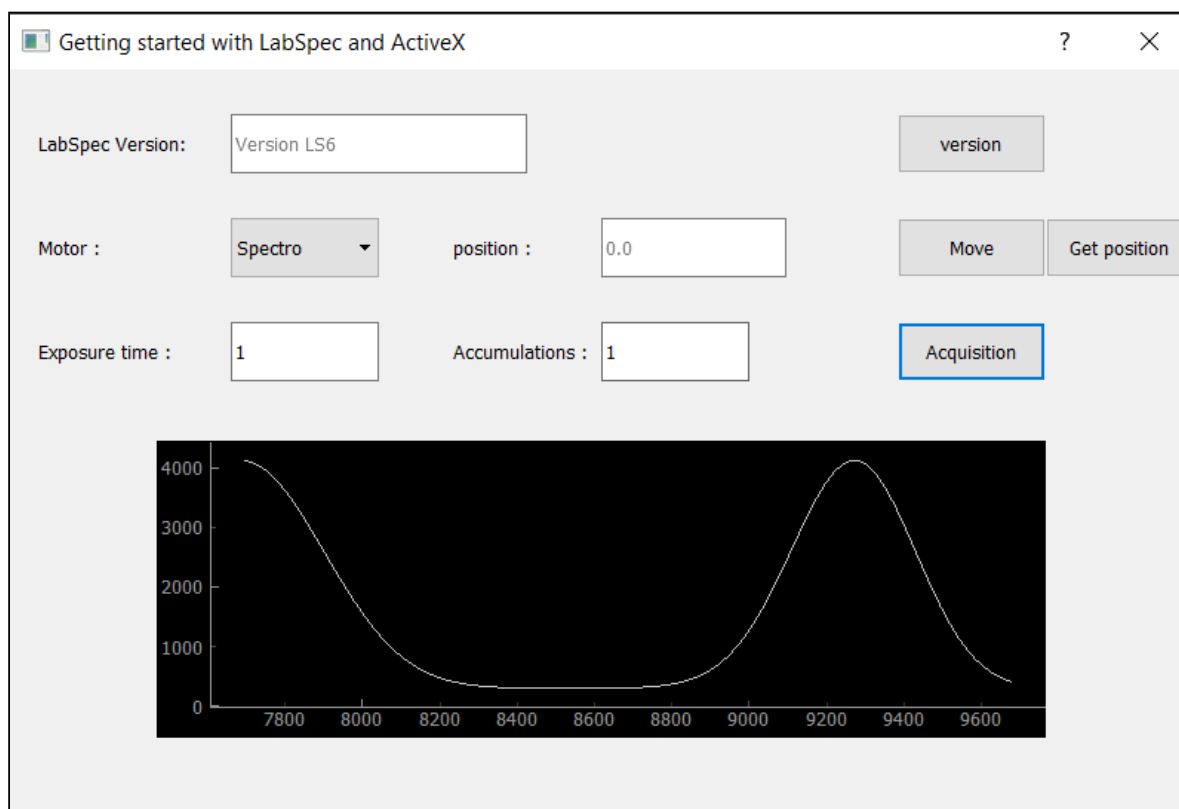
```

def on_click_Acq(self):
    global Exposure
    Exposure = self.textbox_Exposure.text()
    global Accumulation
    Accumulation = self.textbox_Accumulation.text()
    #Start the acquisition
    LabSpec.Acq(10, Exposure, Accumulation, 0, 0)
    #wait the acquisition is finish
    SpectrumID = LabSpec.GetAcqID()
    while SpectrumID < 0:
        SpectrumID = LabSpec.GetAcqID()
    #Get the value array and display
    arrY = LabSpec.GetValue(SpectrumID, "Data")
    arrX = LabSpec.GetValue(SpectrumID, "YData")
    DispX = arrX[1]
    DispY = arrY[1]
    self.my_plot.plot(DispX, DispY)

```

c) Execution

Enter an exposure time and number of accumulation. Click on Acquisition to start a spectral acquisition. The spectrum will be displayed in the ActiveX Control



Project files:

[LS6ActiveX.py](http://hfratlassian.jy.fr:8090/download/attachments/60784666/LS6ActiveX.py?api=v2&modificationDate=1600160503472&version=2)<sup>4</sup>

<sup>4</sup> <http://hfratlassian.jy.fr:8090/download/attachments/60784666/LS6ActiveX.py?api=v2&modificationDate=1600160503472&version=2>