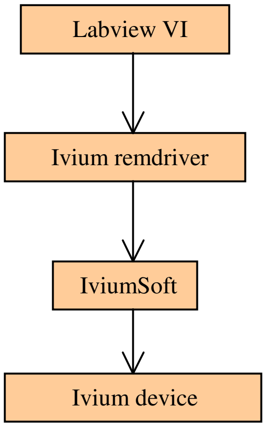
Product note for Ivium Labview Driver 1.0
(Developed in Labview 8.2)

1. Order of communication hierarchy

For optimum results when using Labview to control an Ivium device it is very important to realise the order of hierarchy for communication between the various software and hardware. In this case Labview (the VI) communicates with the Ivium remdriver, which in turn communicates with the IviumSoft-software. Only the IviumSoft communicates directly with the Ivium device (IviumStat/CompactStat). This implicates that the IviumSoft-software should always be running in the background when using a Labview VI to operate the device.
The IviumSoft can be minimised when operating Labview.



Order of hierarchy

2. Directions for use

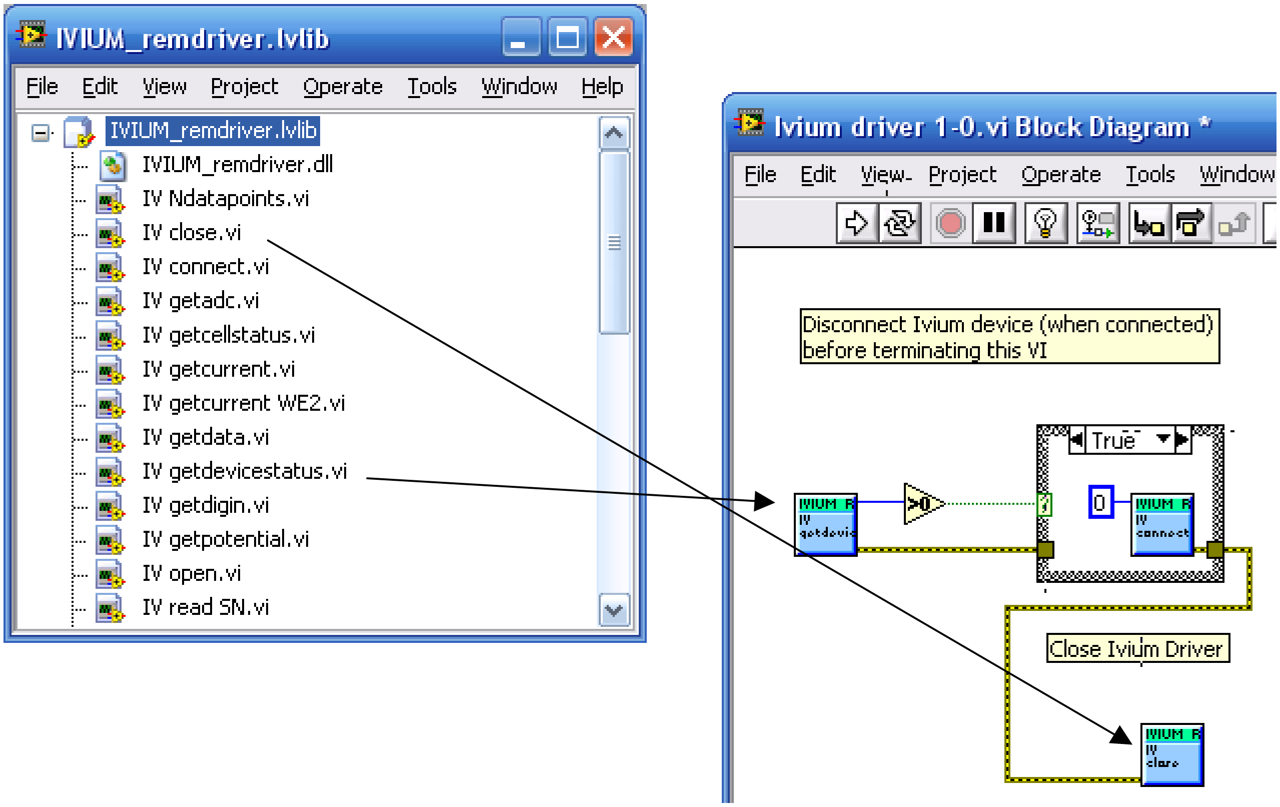
To install, copy the 'Labview' directory into the 'IviumStat' directory of your computer.

The Ivium remdriver (dll) was imported into Labview to create a Labview library (lvlib). In this library, a vi is present for all the functions that can be used to control the Ivium device. These vi's can be found in the Ivium_remdriver directory. A list of functions that can be controlled is given below, a detailed description for each of these functions is given in the IviumSoft development driver which can be found in the Software Development Driver directory.

Imported function	**description**
GENERAL	
IV_open	Opens the driver
IV_close	Closes the driver
IV_getdevicestatus	Returns status of device
IV_readSN(*char)	Returns serial number of selected device
IV_connect(int)	Connect to selected device
DIRECT MODE	
IV_getcellstatus(int)	Returns cell status
IV_setconnectionmode(int)	Select configuration, (off; EStat4EL, EStat2EL, EstatDummy1, EStatDummy2, EstatDummy3, EstatDummy4, Istat4EL, Istat2EL, IstatDummy, BiStat4EL, BiStat2EL)
IV_setpotential(double)	Set cell potential
IV_setpotentialWE2(double)	Set BiStat offset potential
IV_setcurrent(double)	Set cell current (galvanostatic mode)
IV_getpotential(double)	Returns measured potential
IV_setcurrentrange(int)	Set current range, 0=10A, 1=1A, etc,

IV_setcurrentrangeWE2(int)	Set current range for BiStat, 0=10mA, 1=1mA, etc,
IV_getcurrent(double)	Returns measured current
IV_getcurrentWE2(double)	Returns measured current from WE2 (bipotentiostat)
IV_setfilter(int)	Set filter
IV_setstability(int)	Set stability
IV_bistat_mode(int)	Select mode for BiStat
IV_setdac(int,double)	Set dac on external port
IV_getadc(int,double)	Returns measured voltage on external ADC port
IV_setmuxchannel(int)	Set channel of multiplexer
IV_setdigout(int)	Set digital lines on external port
IV_getdigin(int)	Returns status of digital inputs from external port
IV_setfrequency(double)	Set ac frequency
IV_setamplitude(double)	Set ac amplitude
METHOD MODE	
IV_readmethod(*char)	Loads method procedure from disk
IV_savemethod(*char)	Saves method procedure to disk
IV_startmethod(*char)	Start method procedure
IV_savedata(*char)	Saves actual result data to disk
IV_setmethodparameter (*char1,*char2)	Modify method parameter
IV_Ndatapoints(int)	Returns actual available number of datapoints: indicates progress during a run

To use the functions in Labview open the ' IVIUM_remdriver.lvlib' (in the Labview directory) in the getting started window of Labview. Then drag the desired function from the lvlib onto the Block Diagram of the VI that you are building.



When a VI is build to control an Ivium device make sure that the driver is opened before operating the device and closed again before the VI is stopped.

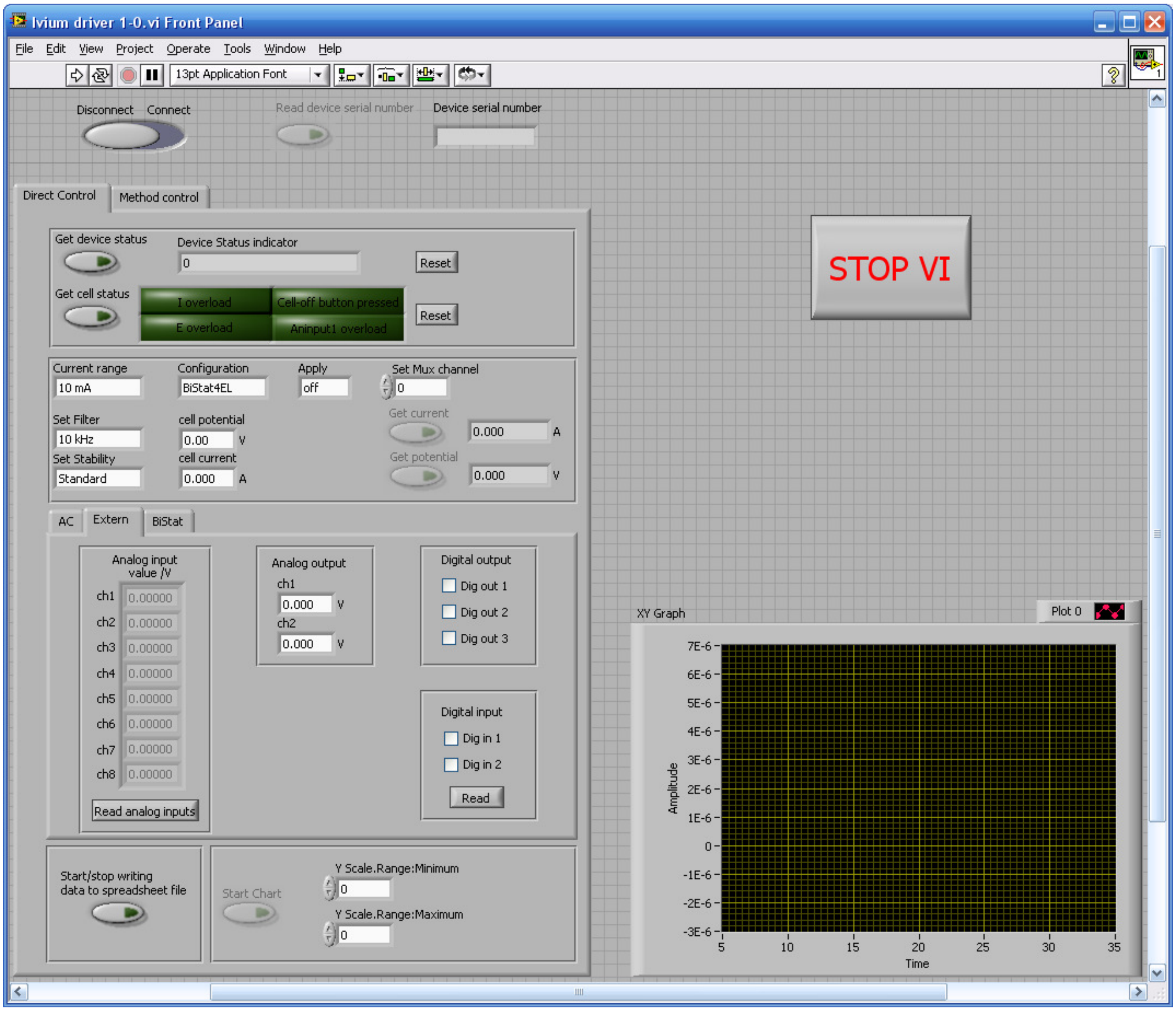
Boundary conditions

Some of the settings of the Ivium device that can be controlled via Labview change an actual setting in the Ivium device, like for example the current range. When one (or some) of these settings are changed outside Labview, i.e. in the IviumSoft or by the device itself when for example using the AutoCR setting in a method, this change of setting is *NOT* read back into Labview. That implicates that the value in the User Interface in the Front Panel may not be the actual setting anymore.

When using the method mode a number of strings (file names and locations) are communicated to the IviumSoft software. These can only be communicated when the device is *CONNECTED*. If the device is not connected it will result in an error.
Note should be taken that the IviumSoft-software has many conditions and safeties incorporated. When using Labview to control an Ivium device the programmer her/himself will be responsible for supplying valid input parameters.

3. Example Ivium Driver 1.0

As an example how Labview may be used to control an Ivium device a Labview VI is build using Labview 8.2. In this example all functions that can be controlled through Labview are included. This example has been set up in such a way that the user interface resembles the user interface of the IviumSoft.



Device serial number:

Device Status indicator: 0

Current range: 10 mA

Configuration: BiStat4EL

Apply: off

Set Mux channel: 0

Set Filter: 10 kHz

cell potential: 0.00 V

Set Stability: Standard

cell current: 0.000 A

Get current: 0.000 A

Get potential: 0.000 V

Analog input value /V

ch1	0.00000
ch2	0.00000
ch3	0.00000
ch4	0.00000
ch5	0.00000
ch6	0.00000
ch7	0.00000
ch8	0.00000

Analog output

ch1: 0.000 V

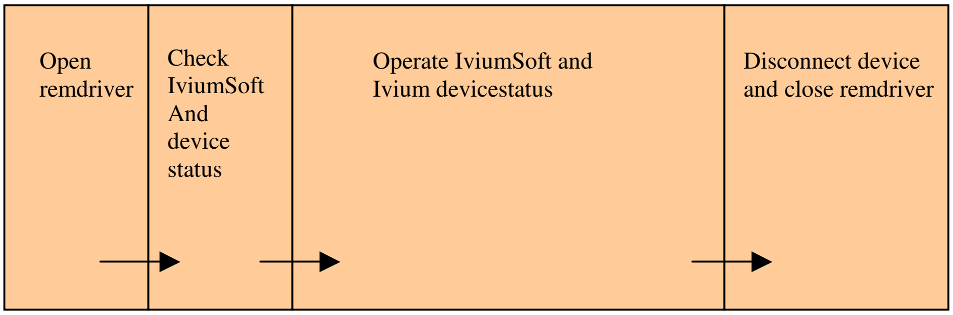
ch2: 0.000 V

Y Scale.Range:Minimum: 0

Y Scale.Range:Maximum: 0

Set up of Ivium Driver 1.0

The Ivium Driver 1.0 has been set up using a flat sequence to ensure that the order of events is correct.

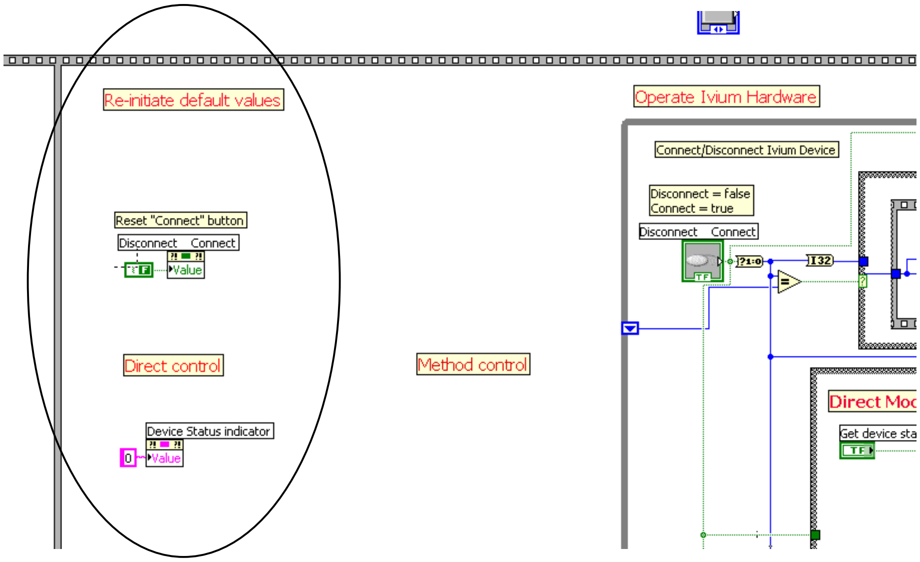

Open remdriver
Check IviumSoft And device status
Operate IviumSoft and Ivium devicestatus
Disconnect device and close remdriver

Ivium Driver 1.0 set-up

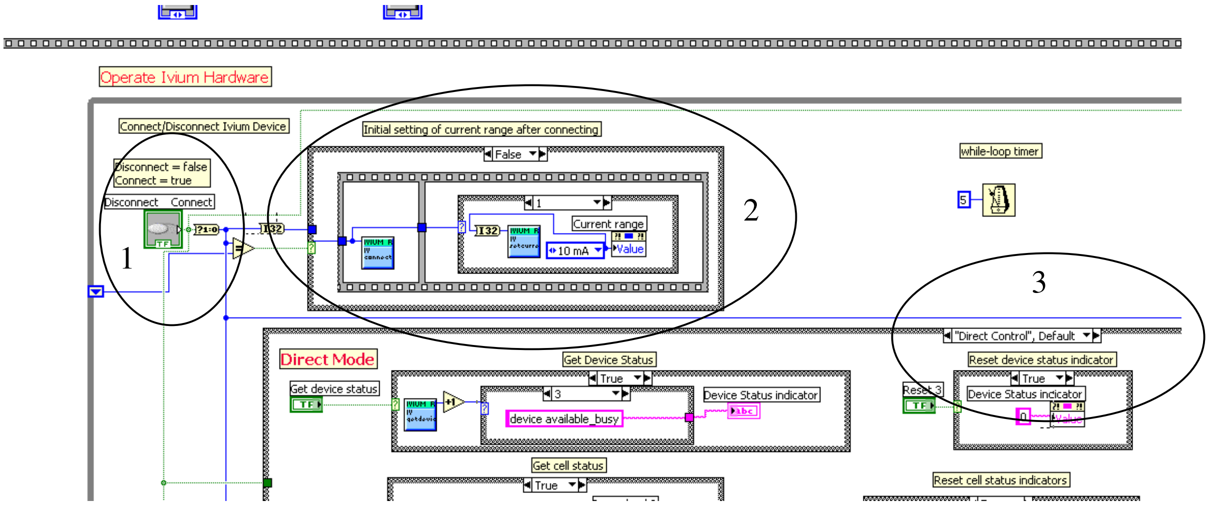
- The first operation is to open the Ivium_remdriver.dll so the IviumSoft and device can be operated.
- Next a check is performed to see whether the IviumSoft is running. If not the VI is stopped. If IviumSoft is running a check is performed to see if the device is connect, and if so it will be disconnected.
- Third sequence is the actual operation of the Ivium device, either in direct-mode or in method-mode.
- Finally, upon closing the VI, the Ivium device is disconnected and the Ivium_remdriver.dll is closed.

Operating sequence of Ivium Driver 1.0: Block Diagram set-up

To ensure the continuance of operating the Ivium device the actual running-sequence of the VI is incorporated in a while-loop. However, before the while-loop commences an initiation of the default values is executed for all the controls for the direct-mode operation. This has been done to ensure that the values shown in the user interface or Front Panel are the actual values. After this has been done the while-loop starts.



First in the while loop (refer to Block Diagram) is the possibility of connecting with the Ivium device (1). Upon connecting, a flat sequence is carried out (2) connecting the device and subsequently setting the current range, again to ensure that the shown value is the actal one.



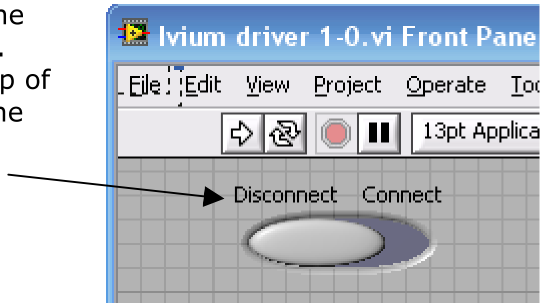
The rest of the while-loop is reserved for a tab controlled case structure (3). This case structure operates depending on the choice of 'direct' or 'method' mode.

The 'direct' case includes all operating parameters, as well as the tab controlled case structure for the AC/Extern/BiStat controls. It also includes the writing-data-to-file operation and the simple chart.

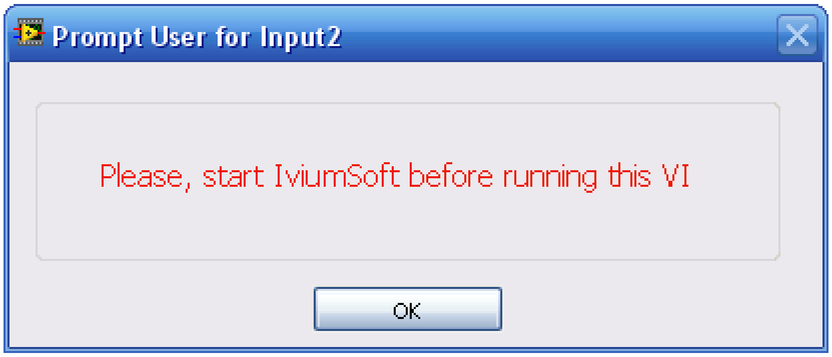
The 'method' case includes all the operating parameters for reading, starting, changing and saving experimental methods.

Operating the Ivium Driver 1.0

Operating the Ivium Driver 1.0 mostly resembles the IviumSoft software. First start the VI from Labview. Clicking the 'Disconnect - Connect' button at the top of the user interface above the tab control connects the Ivium device.



If the device is not connected, none of the direct operating controls or the method controls, work. After connecting, all functions can be operated. This includes the reading of the device serial number. If the IviumSoft is not running a pop-up window will show mentioning this. Upon clicking the 'OK'-button the Ivium Driver 1.0-VI will stop running.



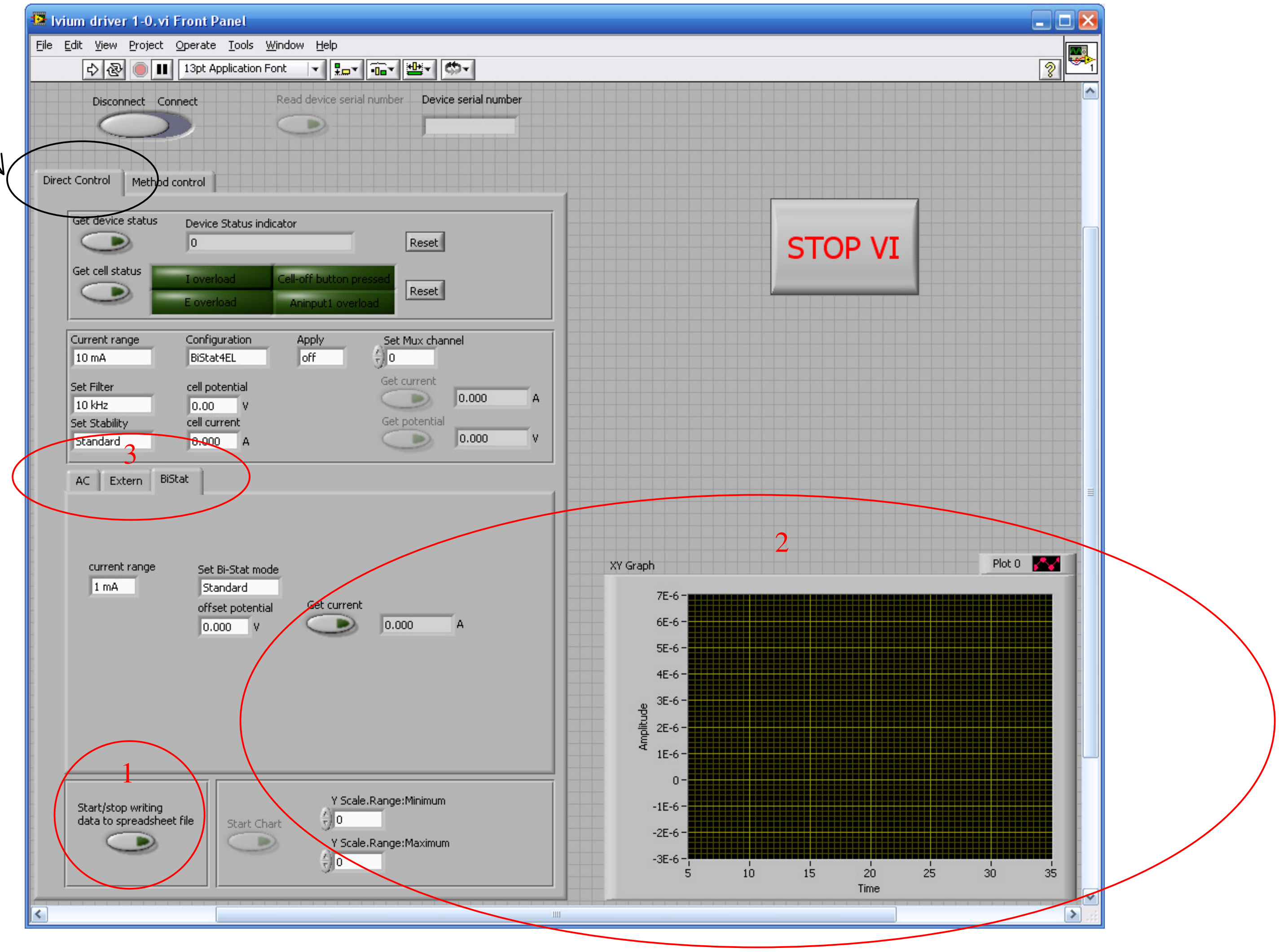
Prompt User for Input2

Please, start IviumSoft before running this VI

When connected the Ivium device can be simply operated by using the controls on the various tab pages.

Direct Mode

In the 'direct' mode tab page the device and cell status can be called and all parameters of the Ivium device can be controlled directly. For some of these parameters a property node is called to enable/disable the controls when relevant. This has been done to eliminate the possibility of sending illegal or conflicting commands that might result in an error and subsequent ceasing of Labview.



Ivium driver 1-0.vi Front Panel

Current range: 10 mA
Configuration: BiStat4EL
Apply: off
Set Mux channel: 0
Set Filter: 10 kHz
Set Stability: Standard
cell potential: 0.00 V
cell current: 0.000 A
Get current: 0.000 A
Get potential: 0.000 V
Device Status indicator: 0

current range: 1 mA
Set Bi-Stat mode: Standard
offset potential: 0.000 V
Get current: 0.000 A

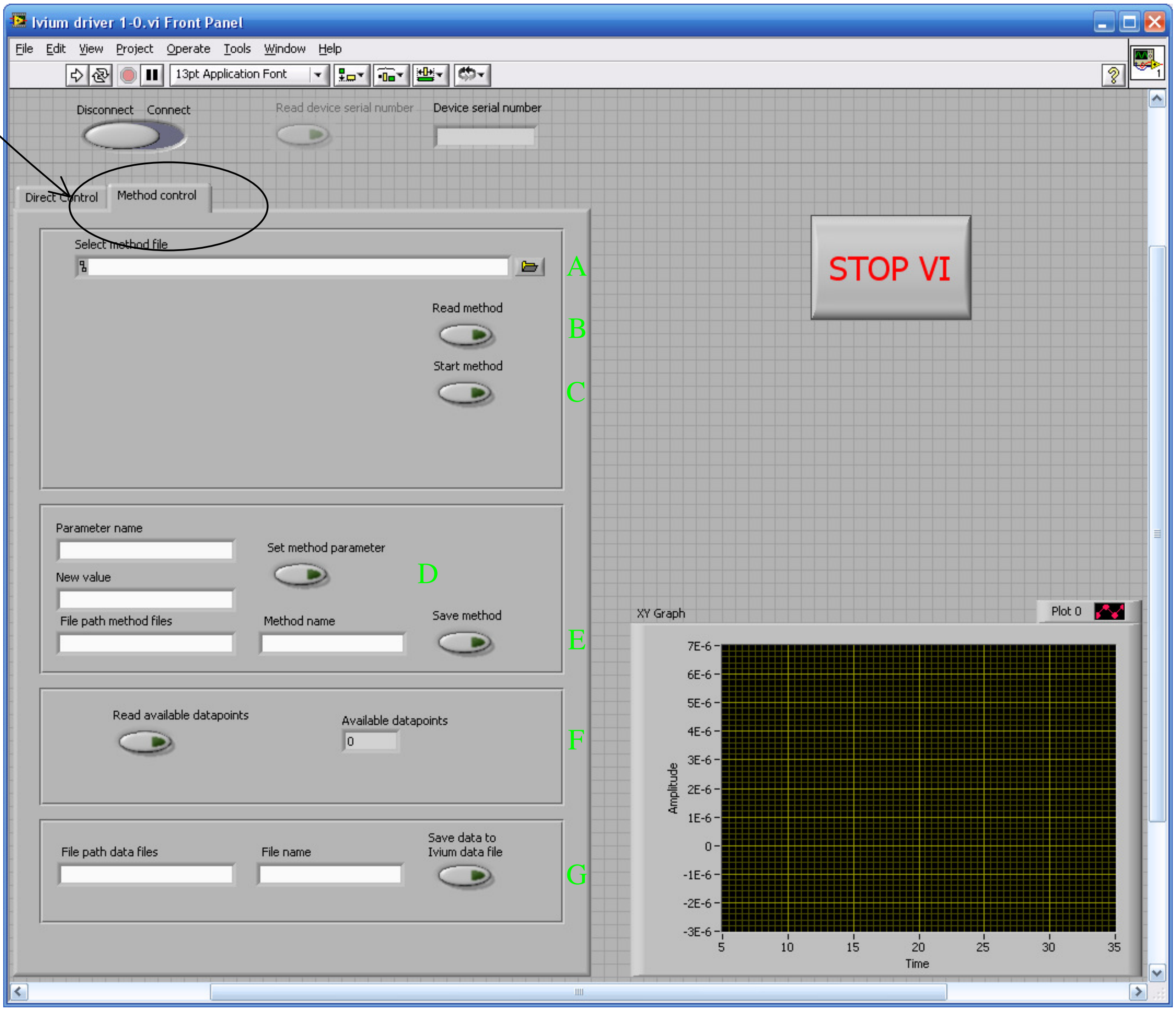
Y Scale.Range:Minimum: 0
Y Scale.Range:Maximum: 0

In the direct mode, as an example, an operation for writing data to spread-sheet file (1) and a simple chart (2) have been incorporated. The chart plots data from the data file and thus only operates when the data is being written to file. The name of the data file is automatically genrated as 'test+date/time'. This filename can be altered in the Block Diagram.

The 'direct' mode also shows an extra number of tab pages: AC/Extern/BiStat (3).

- The 'AC' page allows to control a simple AC experiment.
- The 'Extern' page allows to read and control all external ports.
- The 'BiStat' page only shows when the configuration 'BiStat4EL' or 'BiStat2EL' has been chosen. It then allows BiStat control.

<u>Method mode</u>
In the 'method' mode tab page a method can be selected (A). This method can then be read into the Ivium software and the device (B). Then the method can be started (C). When 'start method'is clicked and no method was read into the IviumSoft, the method currently selected in the IviumSoft is started.



Furthermore, separate parameters of method can be changed (D) and this new method can be saved (E). Care should be taken that the entered method-parameters to be changed are spelled exactly as in the IviumSoft, otherwise it will not register.

During running of a method the number of available data points can be requested (F). In Labview there is no possibility to abort a method once started.

Finally, upon completion of executing a method the then in the IviumSoft available data can be stored as a Ivium data file (.IDF) (G).