Software Pack

Phi6

680020E08

Created 07/09/08
Updated 21/03/14

This manual is related to the following product:

Product – P/N	**H6800 2000**	
Sofware version	**V314**	
Issued date	**21/01/2010**	

Document History

Version	Issued date	Reason
V1	09/2006	Preliminary version
V2	03/2010	Updated by RPH
V3	11/2010	Updated by RPH
V4	28/01/11	Updated by RPH
V5	21/10/11	Reflect Phi6 **Version 3.0** changes
E6		Included soft panels: H710060, H710061, H730050, YAV90096, YAV90128, YAV 90132, YAV 90304, YAV904X8, YAV90832, YAV90CLR, YAV90HVT, YAV90MMU, YAV90PIN (YAV90CIN), YAV90PNE, YAV90321, YAV90048, YAV91616, YAV91775, YAV90CANCON
E07		RPH. Included Phi6 Operator Interface
E08	19/03/14	Added more info on chapter 3

Software History

Version	Issued date	Reason
3.14	09/2006	Preliminary commercial version

TABLE OF CONTENTS

0. About Phi6

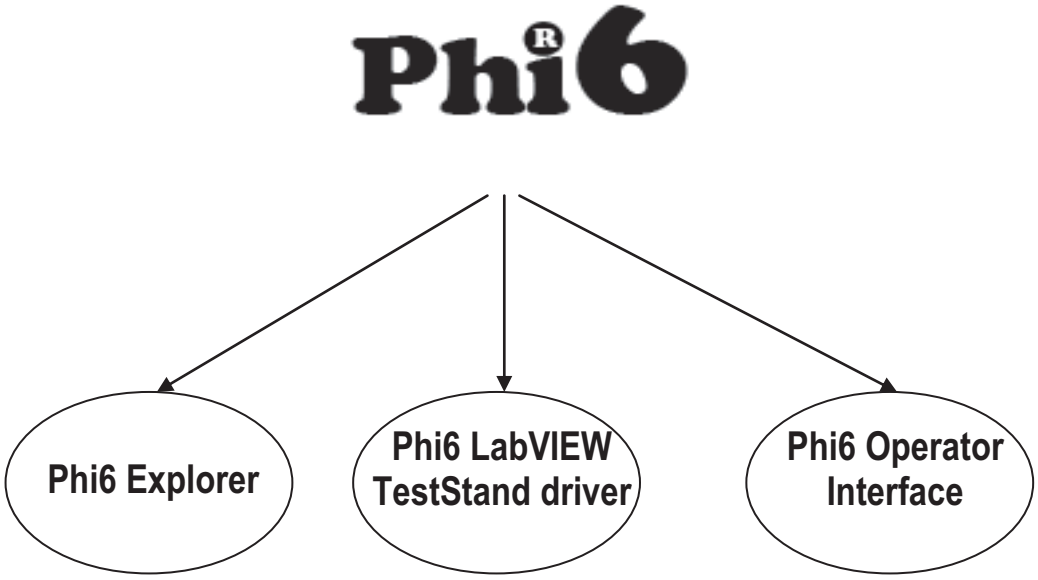
Phi6

is a **registered trademark** of SASistel in classes 9 and 42, Number 6766398

The disclosed technical descriptions in this document do not imply resignation to any right protected by patents and models of utility, or rights of property of software.

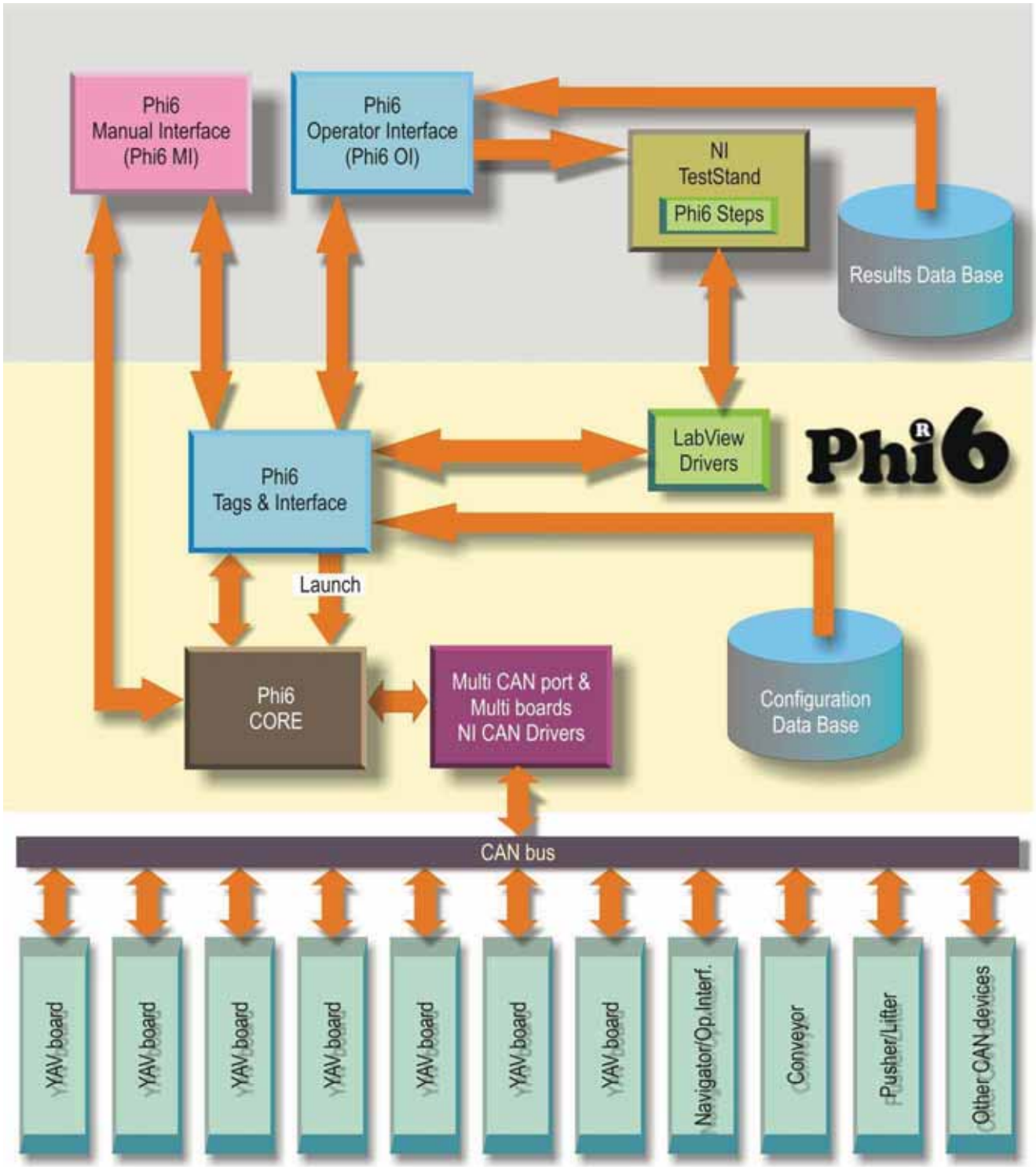
Phi6 is the software environment to manage, control and program 6TL devices and systems.

There are 3 main Phi6 components:



- **Phi6 Explorer**: Configure, calibrate, and quickly operate YAV boards. (Free)
- **Phi6 LabVIEW and TestStand driver**: LabVIEW and TestStand functions necessary to interface with the YAV boards. (Free)
- **Phi6 Operator Interface**: Operation and management of 6TL platforms. (OPTION)

The LabVIEW driver is compatible with LabVIEW versions 10.0 and higher. For compatibility with previous versions please contact 6TL Engineering.



1. YAV modules overview

The use of YAV modules provides a real technical, economical and logistic advantage against all other classic instrumentation solutions. YAV modules are the solution for typical practical problems that occur while designing test platforms. They provide a considerably test system performance increase. The minimal amount of connections and wiring length reduction maintains the best quality and integrity of the signals and provides a short assembly and wiring time.

Being a combination of hardware modules and software drivers, the implementation of the YAV modules in your test system is very fast. When connected, the modules are immediately operative. YAV modules make your test system very flexible, re-configurable, easy to expand and maintain. Each modules can work independently form the tester so you can just plug each of them directly into the ITA frame of a fixture to start debugging the software without the need of using the test platform. While the tester is being used to test PCBAs you have the advantage to debug or build any new configuration in a few minutes.

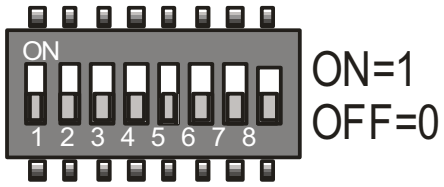
Can bus control has a big advantage; it provides all the control and power supply signals to power the YAV modules with just 4 wires. Therefore the modules can be mounted in any part of the test system or test fixture. The Can bus transmission speed might be lower than the one available in parallel buses, but it is at least 100 times faster than the speed needed to control all the available functions in each of our YAV modules.

1.1 YAV modules common information

1.1.1 SW1 DIP switch functionality

The eight switches contained in SW1 are covering the following functionalies:

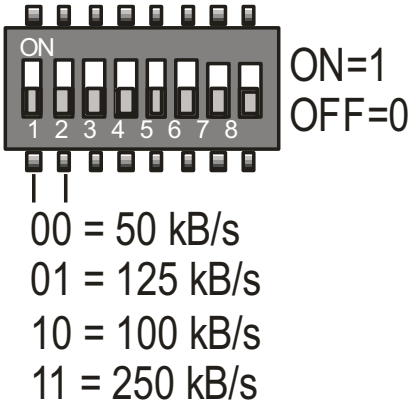
- Set CAN bus speed
- Set module address



1.1.2 CAN bus communication speed.

Switches 1 and 2 set the CAN communication speed:

Select 50Kb/s to communicate with Phi6 and all 6TL modules. All devices communicating through the CAN network must have the same speed configured in order to communicate properly.

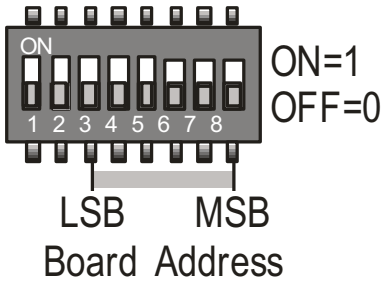


1.1.3 YAV module address

The logic address is composed by 2 elements: the module identification and the hardware address.

The hardware address is selected by switches 3 to 8 of SW1, being 3 the least significant bit (LSB). The module identification is contained in the module firmware and cannot be changed. There is a different identification for each YAV module type. Therefore, modules of different module identification can have the same hardware address.

Since there are 6 bits there are up to 64 possible addresses (from 0 to 63).



1.2 YAV modules standard addressing & Virginia Panel Receivers

Although the user can set any address for his modules, 6TL defines standard addressing for the modules that are installed by default in 6TL testing platforms.

The motivation is to ease compatibility between platforms and systems and 6TL strongly recommends following this standard in order to facilitate platforms setup and maintenance.

The following table shows the standard addressing for the YAV modules and 6TL products installed in the different slots of the Virginia Panel receivers:

SW1 Address (Binary code) MSB LSB	Phi6 Address	Skeeter S6	Gemini 10	Gemini 12	Gemini 12X	Gemini 25	Gemini 50
00 0001	**1**	A1	1	A1	A1	1	1 UP
00 0010	**2**	A2	2	A2	A2	2	2 UP
00 0011	**3**	B1	3	A3	A3	3	3 UP
00 0100	**4**	B2	4	A4	A4	4	4 UP
00 0101	**5**	C1	5	B1	B1	5	5 UP
00 0110	**6**	C2	6	B2	B2	6	6 UP
00 0111	**7**		7	B3	B3	7	7 UP
00 1000	**8**		8	B4	B4	8	8 UP
00 1001	**9**		9	C1	C1	9	9 UP
00 1010	**10**		10	C2	C2	10	10 UP
00 1011	**11**			C3	C3	11	11 UP
00 1100	**12**			C4	C4	12	12 UP
00 1101	**13**				A5	13	13 UP
00 1110	**14**				A6	14	14 UP
00 1111	**15**				B5	15	15 UP
01 0000	**16**				B6	16	16 UP
01 0001	**17**				C5	17	17 UP
01 0010	**18**				C6	18	18 UP
01 0011	**19**					19	19 UP
01 0100	**20**					20	20 UP
01 0101	**21**					21	21 UP
01 0110	**22**					22	22 UP
01 0111	**23**					23	23 UP
01 1000	**24**					24	24 UP
01 1001	**25**					25	25 UP
01 1010	**26**						1 DOWN
01 1011	**27**						2 DOWN
01 1100	**28**						3 DOWN
01 1101	**29**						4 DOWN
01 1110	**30**						5 DOWN
01 1111	**31**						6 DOWN
10 0000	**32**						7 DOWN
10 0001	**33**						8 DOWN
10 0010	**34**						9 DOWN
10 0011	**35**						10 DOWN
10 0100	**36**						11 DOWN
10 0101	**37**						12 DOWN
10 0110	**38**						13 DOWN
10 0111	**39**						14 DOWN
10 1000	**40**						15 DOWN
10 1001	**41**						16 DOWN
10 1010	**42**						17 DOWN
10 1011	**43**						18 DOWN
10 1100	**44**						19 DOWN
10 1101	**45**						20 DOWN
10 1110	**46**						21 DOWN
10 1111	**47**						22 DOWN
11 0000	**48**						23 DOWN
11 0001	**49**						24 DOWN
11 0010	**50**						25 DOWN

Next table shows the list of YAV modules that are installed in each of the 6TL platforms. Note that the “Platform” column indicates whether the module is actually installed in the platform or not:

YAV Module Identificator	Phi6 Address	SW1 Address (Bin Code) MSB…..LSB	YAV Board	Firmware Name	Function (Board Alias)	Platform TTT-1	Platform 6TL-22	Platform 6TL-32	Module P/N
YAV90304	0	00 0000	MW0304	YAV-90304	6TL_Platform_Status	X	X		8TTT2BOX
YAV90304	63	11 1111	H5400 0100	MMI5400	6TL_MMI		X	X	H7300 03 XX
YAV90304	62	11 1110	YAV90304	YAV-90304	6TL_Beacon	-/X	-/X	-/X	H7800 10 XX
YAV90304	61	11 1101	MW0304	YAV-90304	6TL_Rejection_Channel	-/X	-/X		H7300 06 00
YAV90304	60	11 1100	87100600	R300K	6TL_Conveyor_Control			X	H710061
YAV90304	59	11 1011	8710061E02	R300K	6TL_Feed_Control			-/X	H710061
YAV90304	58	11 1010	8710061E02	R300K	6TL_Outp_Control			-/X	H710061
YAV90304	57	11 1001	YAV90304	YAV-90304	6TL_Apil_Control			-/X	H7800 10 XX
YAV90304	56		YAV90304	YAV-90304	6TL_Power_Triphasic				
YAV90304	55		YAV90304	YAV-90304	6TL_Power_Rack				
YAV90304	54		YAV90304	YAV-90304	6TL_Soft_Start				
YAV90304	53		YAV90304	YAV-90304	6TL_CAN_Power				
YAV90304	52		YAV90304	YAV-90304	6TL_PDU2				
YAV90304	51		YAV90304	YAV-90304	6TL_PDU1				
YAV90304	50		YAV90304	YAV-90304	6TL_RF_pusher				
H710061	0	00 0000	8710061E02	H7100-61	6TL_Conveyor_Belt			X	H710061
H710061	1	00 0001	8710061E02	H7100-61	6TL_Conveyor_Wide			X	H710061
H710061	2	00 0010	8710061E02	H7100-61	6TL_Feed_Belt			-/X	H710061
H710061	3	00 0011	8710061E02	H7100-61	6TL_Feed_Wide			-/X	H710061
H710061	4	00 0100	8710061E02	H7100-61	6TL_Outp_Belt			-/X	H710061
H710061	5	00 0101	8710061E02	H7100-61	6TL_Outp_Wide			-/X	H710061
H710061	6	00 0110	8710061E02	H7100-61(V02.xx)	6TL_Apil_Belt			-/X	H710061
H710061	7	00 0111	8710061E02	H7100-61(V02.xx)	6TL_Apil_Wide			-/X	H710061
H710061	8			H7100-61	6TL_Base_Lifter				H710061
H710060	0	00 0000	87100600	H7100-60	6TL_Pusher		-/X	X	H710060
H710060	1	00 0001	87100600	H7100-60	6TL_Lifter			X	H710060
H710060	7	00 0111	87100600	H7100-60	6TL_Selector			-/X	H710060
YAVCANCON	0	00 0000	YAVCANCON	YAV-CANCON	6TL_Fixture_ID_Master	X	X	X	YAVCANCON
YAVCANCON	1	00 0001	YAVCANCON	YAV-CANCON	6TL_Fixture_ID_Slave		-/X	X	YAVCANCON
MMI	0	00 0000	MMI	MMI	6TL_MMI		X	X	H7300 0700
YAV90PNE*	25	01 1001	YAV90PNE	YAV-90PNE	6TL_Pneumatic			X	YAV90PNE

* Installed in

First (YAV Module Identification) and second (Phi6 Address) columns are composing the address of the module when it is part of a 6TL testing platform.

There are some modules that do not feature SW1, therefore, user can not change the address. This modules are YAVCANCON’s and the MMI, P/N H730003xx, and their address is programmed in the factory with the address shown in the table.

Fourth column (YAV module) refers to 6TL’s internal code for the PCB HW used by the module.

Fifth column (Firmware name) is referring to 6TL’s internal name given to the firmware running in the PCB HW of the module.

Sixth column (Function; Module Alias) refers to the default (and standard 6TL) name given by Phi6 to each module, depending on his function in the 6TL platform. These alias could be changed through Phi6 Explorer, edit Tags.

Seventh column (Platform) is informing about what platform uses what modules. A cross [X] is indicating that the module is always installed as standard delivery. A cross with bar [-/X] is indicating that that module is a potential option for that platform.

Eighth column (Module P/N) is showing the module commercial P/N of the module (Typically 8 or 9 digit number. If less numbers in the table, complete with 0's).

2. Installing the Software

To install the software, complete the following steps:

1. Install NI LabVIEW (only if you will use the Phi6 LabVIEW driver).

2. Install the NI-CAN driver.

3. Install TestStand (only if you will use the Phi6 Operator Interface).

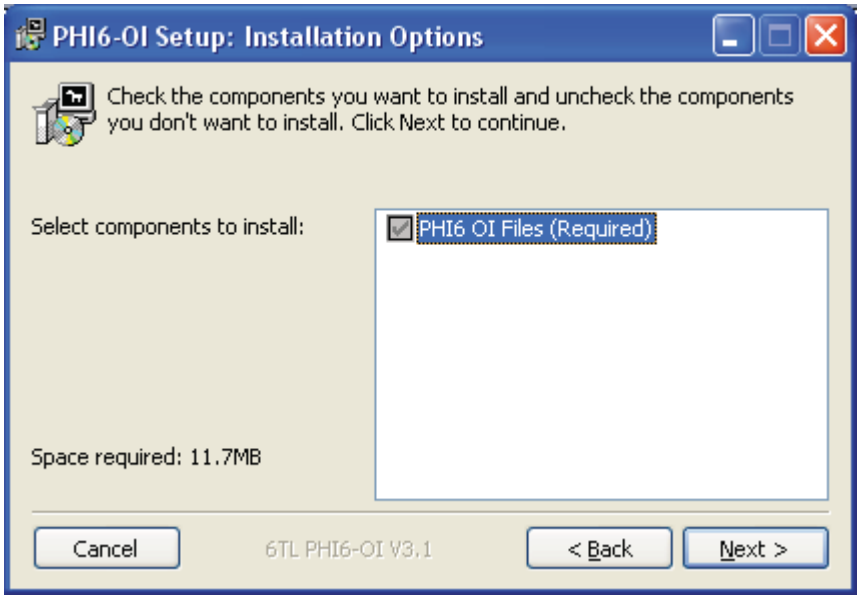
4. Install Phi6 Driver by executing SETUP-PHI6-3.X.exe. Select the appropriate components:

1. *PHI6 Files*: installs the Phi6 Explorer (always required).
2. *LabVIEW Phi6 Drivers*: creates the LabVIEW palette to interface with the YAV boards.
3. *LabVIEW RunTime Engine 2010*: only needed if LabVIEW has not been installed previously (installation of Phi6 Explorer only).

To continue with install process follow the steps of the displayed windows.

IMPORTANT NOTE: this procedure must be repeated every time a new LabVIEW installation is performed. The Phi6 driver uses the 'NI-CAN' library located in the vi.lib of each LabVIEW installation. Same wise, it stores the Phi6 driver inside the instr.lib folder of each LabVIEW installation.

4. Install Phi6 Operator Interface (Option) by executing SETUP-PHI6-OI-3.X.exe:



To continue with install process follow the steps of the displayed windows.

3. Phi6 files organization

3.1 Phi6 Explorer

Phi6-Explorer.exe is installed by default into the following location: C:\Program files\6TL

You can also launch Phi6 Explorer from the Programs Menu or by creating a shortcut in your desktop.

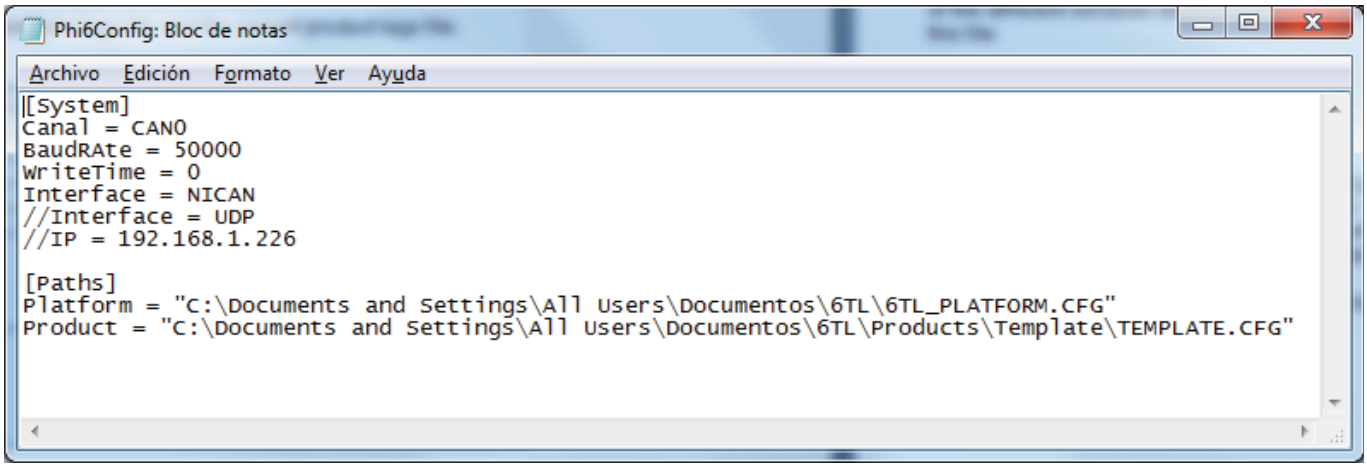
3.2 Phi6 configuration files

Phi6 configuration files are installed in C:\Documents and Settings\All Users\Documents\6TL folder. These are the files and their function:

Phi6Config.ini	Phi6 configuration file. CAN bus configuration
6TL_PLATFORM.cfg	Configuration of the platform tags
Phi6-Platform.cfg	Platform settings. Only for Phi6 Operator Interface
Position.cfg	Phi6 Operator Interface GUI configuration
\Products	Info about YAV boards installed on the fixture. A subfolder for every product (fixture).

3.2.1 Phi6Config.ini

Phi6Config.ini is the Phi6 configuration file and it is generated automatically after the install process. In this file, you can configure the interface for the CAN connection and set the CAN parameters. You can also select the paths of the platform tags file and the current product tags file



Field	Description	Default value
Canal	Select the CAN channel if using NICAN interface, check in NI MAX. (Does not apply with UDP interface)	CAN0
BaudRate	Baud Rate. To interface with YAV Boards must be set at 50000	50000
WriteTime	Delay on CAN transmissions. [For 6TL internal use only]	0
Interface	NICAN: Connection to CAN Bus through a NI-CAN High Speed board (PCI, PXI or USB). UDP: 6TL CAN Gateway through an Ethernet connection, usually YAV90MMU	NICAN
IP	IP of Ethernet CAN gateway device	192.168.1.226
Platform	Platform tags file path, includes tags for all the platforms devices. Only needed when interfacing with a 6TL platform.	"...\Documents\6TL\6TL_PLATFORM.CFG"
Product	Product tags file path. Template product installed as reference. See section 3.3	"...\Documents\6TL\Products\Template\TEMPLATE.CFG"

3.2.2 Platform tags file

A default platform configuration file "6TL_PLATFORM.cfg" is installed. The file contains information about the YAV boards installed in a platform. The file is needed only when interfacing with the YAV boards that control the different components of a 6TL platform. The file includes all possible system YAVs; usually the actual system has only a subset of this configuration.

Although they are fixture devices (not platform devices), the file also includes the YAVCANCON identification modules, since they are a 6TL standard device:

YAVCANCON#00 = 6TL_Fixture_ID_Master
YAVCANCON#01 = 6TL_Fixture_ID_Slave

3.2.3 Phi6-Platform.ini (Phi6 Operator Interface option)

This file is installed with the Phi6 Operator Interface. It configures a set of 6TL platform settings. For a detailed description see Phi6 Operator Interface chapter on this manual (7.1.2).

3.2.4 Position.ini (Phi6 Operator Interface option)

This file is installed with the Phi6 Operator Interface. It configures the window position and size of the different windows displayed in the Operator Interface GUI. It is not recommended to edit this file.

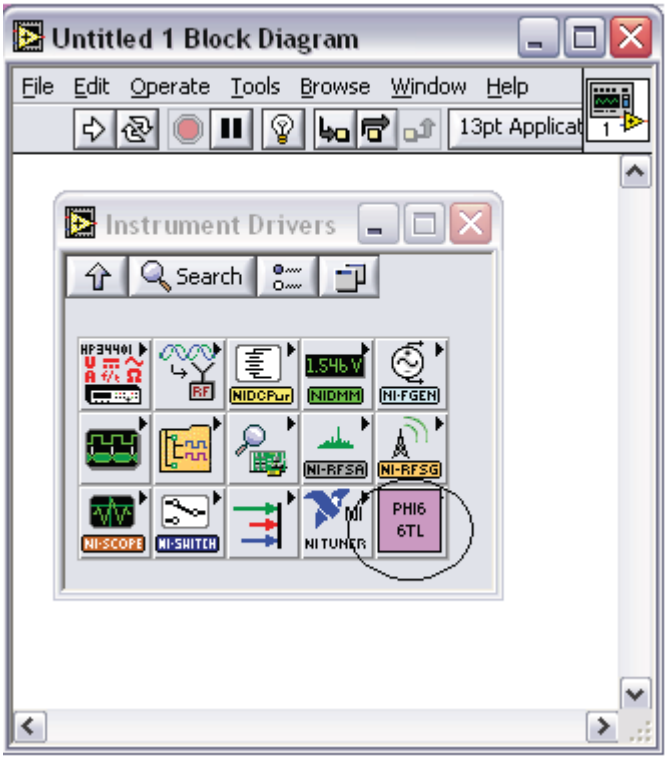
3.3 Products folder

6TL test standard divides projects by product. Each product is a different fixture. When installing Phi6 the default product 'Template' is created. The product contains a configuration file (Template.cfg), which has to be edited to contain all information of the YAV boards installed in the product fixture (this can be done from Phi6 Explorer, see section 4.2).

The products folder is also located at the C:\Documents and Settings\All Users\Documents\6TL path.

3.4 LabVIEW Phi6 Driver

The LabVIEW driver is installed in the instr.lib folder of LabVIEW 2010 and all LabVIEW versions above. It appears as a sub-palette in the Instrument Drivers section of the Instrument I/O palette.



For example for a computer with LabVIEW 2010, the driver is installed in the following folder:

C:\Program Files\National Instruments\LabVIEW 2010\instr.lib\Phi6\Common\Instrum

It is recommended to perform a Mass Compile action of the folder to save all VIs in the LabVIEW version you are using.

3.5 Phi6 Operator Interface

The Operator Interface is contained in the executable Phi6-Platform.exe, which controls and manages the operations of all 6TL platforms: 6TL-19; 6TL-22; 6TL-29; 6TL-31; 6TL-32; 6TL-33. The executable is located in the C:\Program files\6TL folder.

4. Configuring and Testing in Phi6 Explorer

Phi6 Explorer is the appropriate tool to quickly operate the YAV boards. When launched it will automatically detect all YAV boards connected in the system:

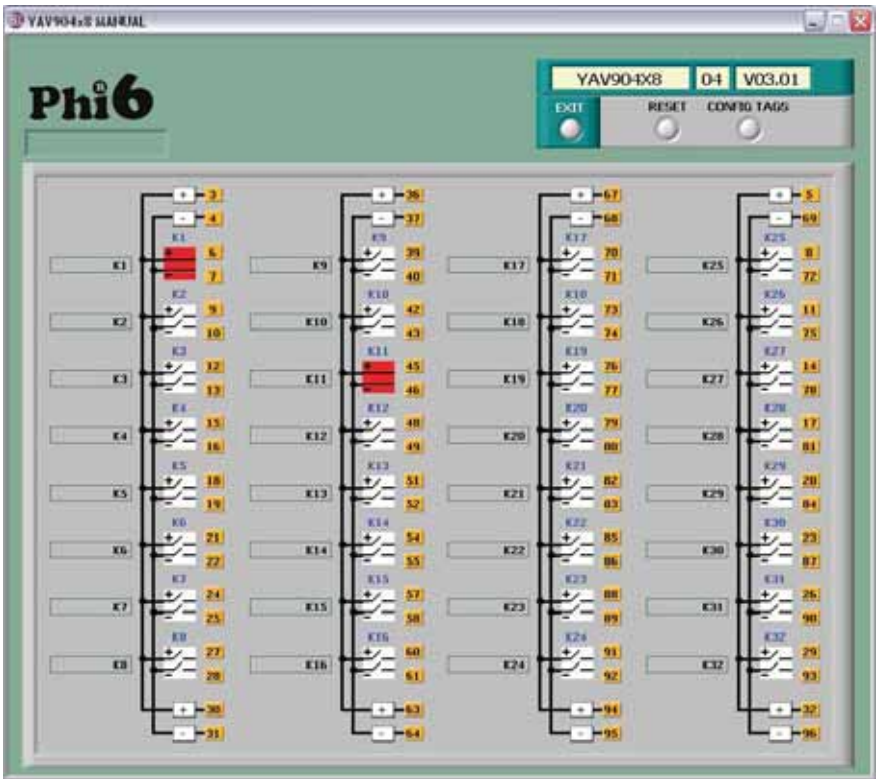
In the example above 3 YAV boards have been detected. The first is the YAV90832 (Multifunction) at address 04. The second and the third are YAV90304 that belong to the Platform YAVS group; therefore, the system has already identified its name and the tags are already configured.

The product loaded by default is “Template”. This means that all changes we make in the configuration parameters of the boards will be reflected in the Template.cfg file, located in the C:\Documents and Settings\All Users\Shared Documents\6TL\Products\Template folder. Once different products have been created the user is able to select among the different products. The path of the current product is stored in the Phi6Config.ini file.

Note the green led indicator at the top right side of the panel. This means that the CAN bus is working properly.

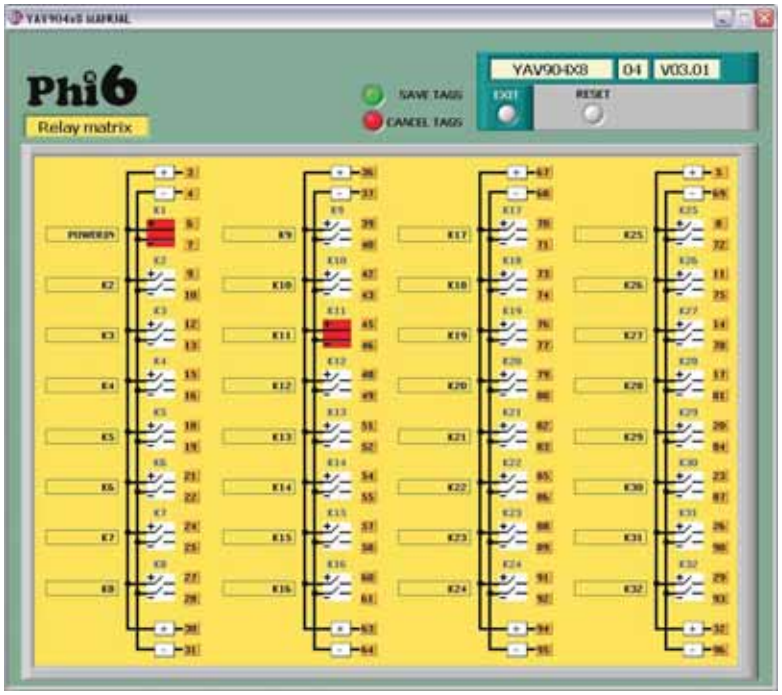
4.1 Operate the YAV boards

To operate the YAV boards simply click the appropriate line in the panel and the YAV’s specific screen will pop-up. In the following example, relays K1 and K11 of a YAV904X8 have been activated:



4.2 Configure YAV channels

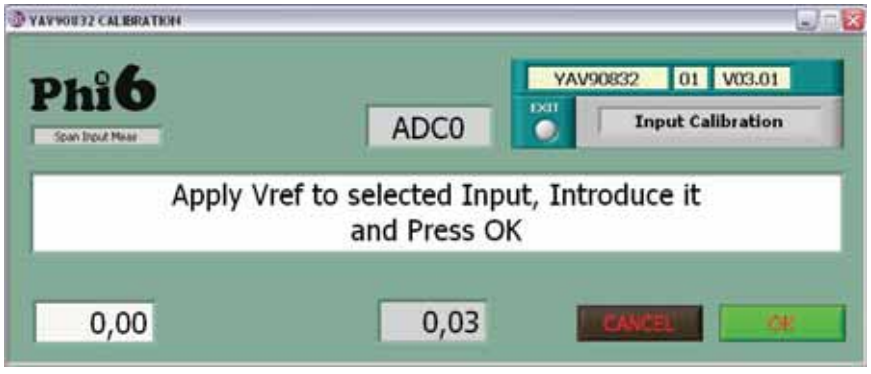
To interface with the channels of the YAVs, each channel needs to have a tag name. From Phi6 Explorer we can easily edit the tag names of each channel. Click the ‘Config Tags’ button at the top right side. Then an alias name for the board and a tag for each channel can be added. After making all changes click the Save Tags button to reflect the changes and the product configuration file will be automatically edited.



There is no need to change the tag names of the Platform YAVs, since they come already defined in the 6TL_PLATFORM.cfg file. It is recommended to edit the tag names of only the YAVs specific to the product you are developing.

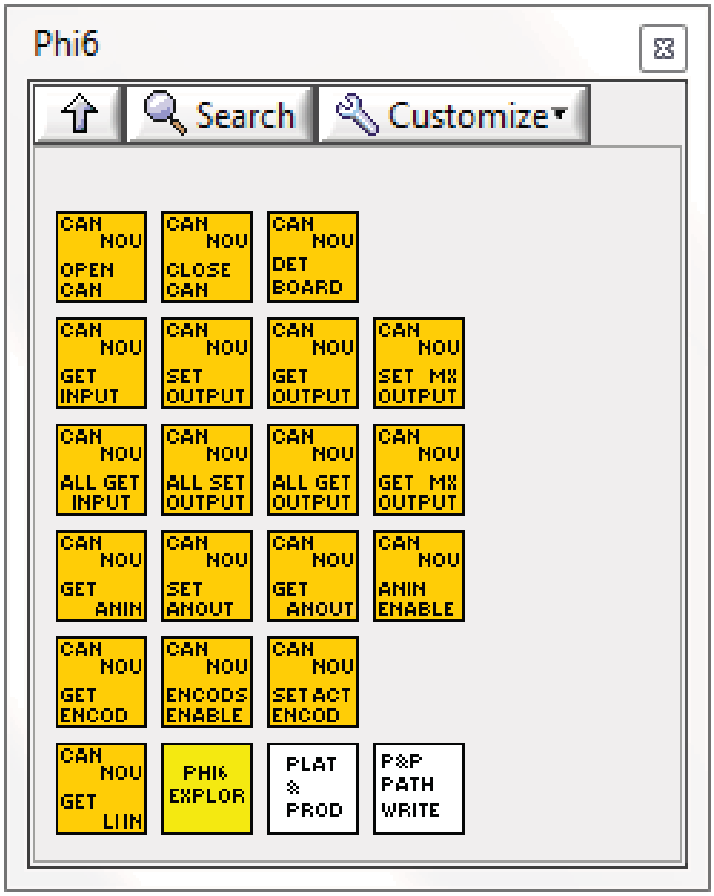
4.3 Calibrate analog channels

From Phi6 Explorer we can also calibrate the analog channels of the YAV boards. Click the 'Calibration' button at the top right side and select 'Input Calibration' for AI channels and 'Output Calibration' for AO channels. Follow the steps of the displayed windows to complete the calibration.



5. Using the LabVIEW driver

The Phi6 palette contains the following functions:



4. **Open Phi6**: Opens CAN communication to the YAV boards. Needed at the beginning of an application before any YAV operation is executed (See Phi6 Template).

5. **Close Phi6**: Closes CAN communication to the YAV boards. Needed at the end of an application to properly terminate CAN communication (See Phi6 Template).

6. **Detect YAV Boards**: sends a request to all YAV boards to identify themselves. Used to force a detection of all boards connected to the system. The output is the number of boards detected. Usually used to verify the number of boards in the system.

7. **PHI6-6200 P&P Path File Write**: used to edit to the platform and product tag files path in the Phi6Config.ini file. (this VI is located in the C:\Program files\National Instruments\LabVIEW 2010\instr.lib\Phi6\Common\Instrum\Phi6 MI\PHI6-6200.llb folder).

8. **Phi6 Get Platform & Product**: used to load the product and platform tag files indicated by the Phi6Config.ini file. This operation programmatically changes the product and platform tags available for the YAV operation functions.

9. **PHI6-Explorer Starter**: used to programmatically start the PHI6-Explorer interface.

10. **YAV SET OUTPUT**: sets an output channel of a YAV board.
    The 'Label' input specifies which tag (channel) to set. The labels are stored in the Product and Platform configuration files.

    The 'Action' input specifies what action to set; depending on the YAV board there are different actions available:

    - YAV90832, YAV904X8, YAV90132, YAV90PIN, YAV90128, YAV90304:
      - ON: sets a digital output channel true.
      - OFF: sets a digital output channel false.
      - BLINK: blinks the digital output channel at a frequency of 2Hz.

    - YAV90PNE:
      - ON: sets a digital or pneumatic output channel true.
      - OFF: sets a digital or pneumatic output channel false.

    - YAV90CLR:
      - GET SAMPLE: asks the board to send a single colour measurement for each channel.
      - GET DATA: asks the board to send continuously colour measurements for all channels.

    The VI has a PASS/FAIL flag output indicating whether the operation has been performed successfully or not.

11. **YAV ALL SET OUTPUT**: sets multiple output channels of a YAV board.
    - The 'Board' input specifies the YAV card to use. The name has to be followed by the address (Ex: YAV904X8#04).
    - The 'Action' input specifies what action to set, ON or OFF.
    - The 'Bits' input specifies which channels to set. Format is hexadecimal, each digit controls four channels (four bits). There are 8 digits (up to 32 channels).
      Example: 00000021hex actuates on the first and sixth relay of the card YAV90132. If 'Action' was set to ON, it will set the relays to True. If 'Action' was set to OFF, it will set the relays to False
    - The 'Block' input is only used for the cards that have more than 32 channels, like the YAV90128 and the YAV90HVT. Block 1, Bits 00000001hex means channel 33.

12. **YAV GET INPUT**: gets the status of a digital input channel of a YAV board.
The 'Label' input specifies which tag (channel) to set. The labels are stored in the Product and Platform configuration files.

The 'Status' output is a boolean indicating the status of the digital input channel.

The VI has a PASS/FAIL flag output indicating whether the operation has been performed successfully or not.

13. **YAV ALL GET INPUT**: gets the status of multiple digital input channels of a YAV board.
- The 'Board' input specifies the YAV card to use. The name has to be followed by the address (Ex: YAV90832#01).
- The 'Bits' input specifies which channels to set. Format is hexadecimal, each digit controls four channels (four bits). There are 8 digits (up to 32 channels).
  Example: 00000021hex gets the status of the first and sixth digital input of the card YAV90832.
- The 'Block' input is only used for the cards that have more than 32 channels, like the YAV90096.

14. **YAV GET OUTPUT**: gets the status of a digital output channel of a YAV board.
The 'Label' input specifies which tag (channel) to get. The labels are stored in the Product and Platform configuration files.

The 'Status' output is a boolean indicating the status of the digital output channel.

15. **YAV ALL GET OUTPUT**: gets the status of multiple digital output channels of a YAV board.
- The 'Board' input specifies the YAV card to use. The name has to be followed by the address (Ex: YAV90832#01).
- The 'Bits' input specifies which channels to set. Format is hexadecimal, each digit controls four channels (four bits). There are 8 digits (up to 32 channels).
  Example: 00000021hex gets the status of the first and sixth digital input of the card YAV90832.
- The 'Block' input is only used for the cards that have more than 32 channels, like the YAV90096.

16. **YAV SET ANALOG OUTPUT**: sets an analog output channel of a YAV board.

The 'Label' input specifies which tag (channel) to set. The labels are stored in the Product and Platform configuration files.

The 'Value' input is regularly used to specify the output level to set:

- YAV90PNE: analog output voltage level from 0 to 5V.
- YAV90832: this card has eight analog outputs. The first four (channels 0 to 3) are used as a regular analog output, where the 'value' input set a voltage level from 0 to 15V. The last four outputs (channels 4 to 7) are PWM outputs; the 'value' input set the duty cycle, from 0 to 100 (percentage of time the level is high). The PWM outputs period is 102,4 µs (frequency 9,765 kHz) and the voltage level is fixed to 5V.

17. **YAV GET ANALOG INPUT**: gets the value of an analog input channel of a YAV board.
The 'Label' input specifies which tag (channel) to get. The labels are stored in the Product and Platform configuration files.

The 'Volts' output provides the value of the channel:

- YAV90832: analog input voltage level from 0 to 30V.
- YAV90PNE: readout of the pressure sensor (kPa).

18. **YAV GET LIGHT INPUT**: gets the RGB and XYZ coordinates of a YAV90CLR (Colorimeter) channel.
The 'Label' input specifies which tag (channel) to get. The labels are stored in the Product and Platform configuration files.

The output is a vector of six components (R,G,B,X,Y,Z).

19. **YAV ENABLE ANALOG INPUTS**: used for the YAV cards that have the same pin used for multiple functionality. The pins of the Analog Inputs 0 to 15 of the YAV90832 are shared with digital inputs. To use them as Analog Inputs they need to be enabled first with this function.
    - The 'Board' input specifies the YAV card to use. The name has to be followed by the address (Ex: YAV90832#01).
    - The 'Enables' input specifies which inputs to enable. Format is hexadecimal, each digit controls four channels (four bits). For example 1100hex enables analog inputs 8 and 12.

The VI has a PASS/FAIL flag output indicating whether the operation has been performed successfully or not.

20. **YAV SET ENCODER ACTION**: sets an action to one of the encoders of the YAV90832 card.
    The 'Label' input specifies which tag (channel) to set. The labels are stored in the Product and Platform configuration files.

    The 'Action' input specifies what action to set. The available actions are:
    - ASK: asks an encoder value. Forces an update of the encoder value.
    - START: starts the encoder readout.
    - STOP: stops the encoder readout.
    - RESET: resets the encoder value.

21. **YAV GET ENCODER VALUE**: gets the value of one of the encoders of the YAV90832 card.
    The 'Label' input specifies which tag (channel) to set. The labels are stored in the Product and Platform configuration files.

    The 'Encoder' output is the numeric value of encoder states.

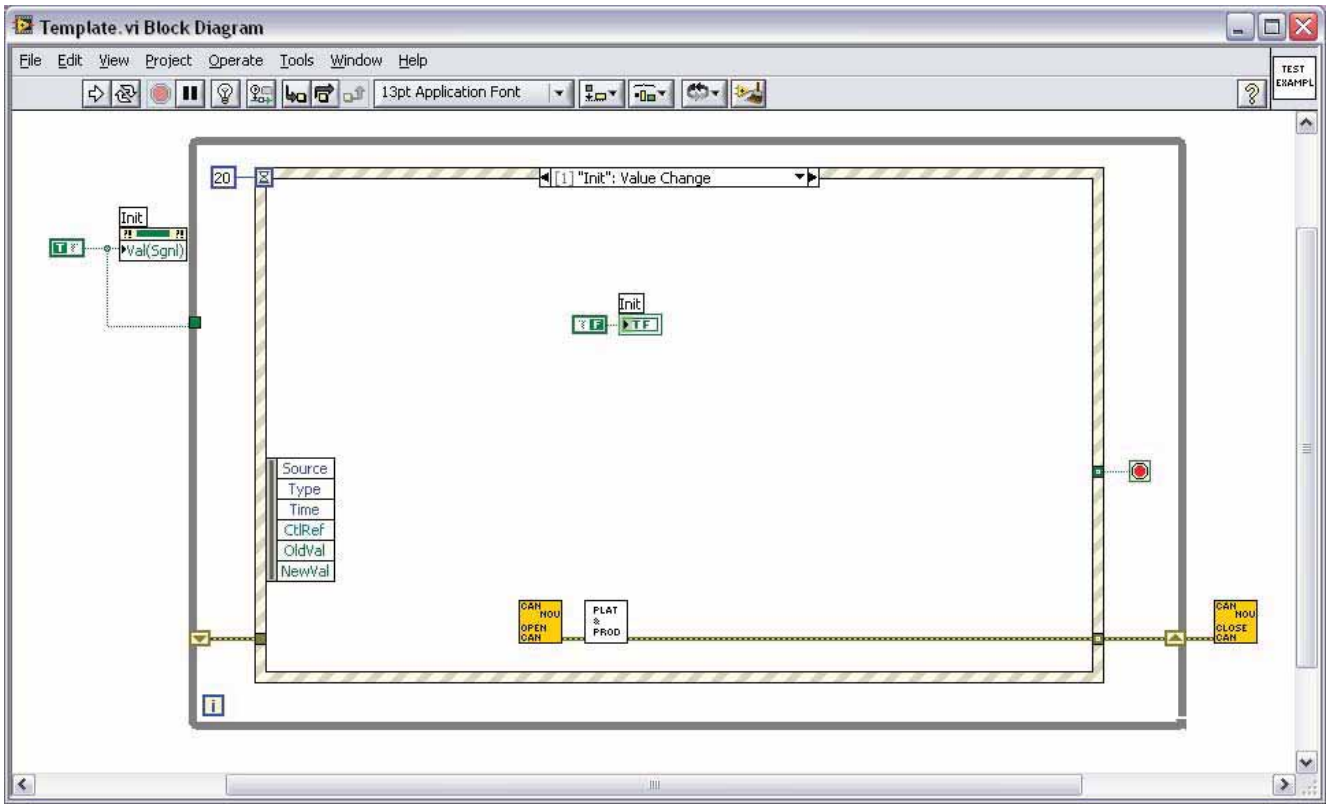
22. **YAV ENABLE ENCODERS**: used for the YAV cards where the same pin is used for multiple functionality. The pins of the three encoders of the YAV90832 are shared with analog and digital inputs. To use them as encoders they need to be enabled first with this function.
    - The 'Board' input specifies the YAV card to use. The name has to be followed by the address (Ex: YAV90832#01).
    - The 'Encoders' input specifies which encoders to enable. Format is hexadecimal, each digit controls four channels (four bits).

    The VI has a PASS/FAIL flag output indicating whether the operation has been performed successfully or not.

5.1 Example of Phi6 operation

With the Phi6 driver installation comes an example application, the Template.vi contained in the Template product folder. It can serve as a starting point for developing your application.

The following screenshot shows the basic steps to start operating with Phi6:



1. Run the Open Phi6 VI.
2. Run the Phi6 Get Platform & Product VI. These will load all channels of the system, with its tag names.
3. Use the different Phi6 functions to interface with the YAV channels.
4. At the end Close Phi6.

6. TestStand driver (Phi6 Operator Interface only)

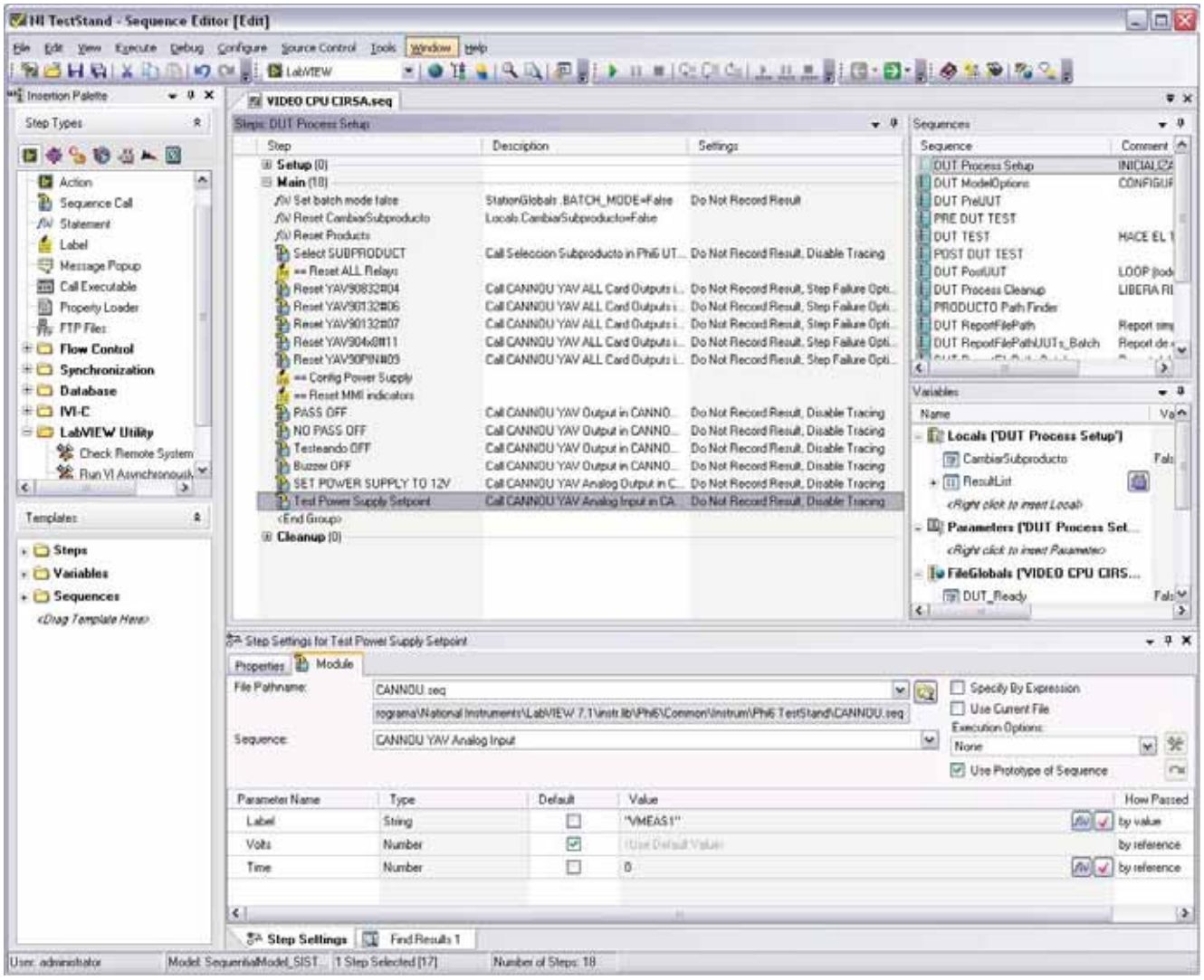
With the installation of the Phi6 Operator Interface the necessary tools to interface the YAV boards from TestStand are also installed.

To use the YAV boards from TestStand load the 'CANNOU.seq' sequence located in the following folder:

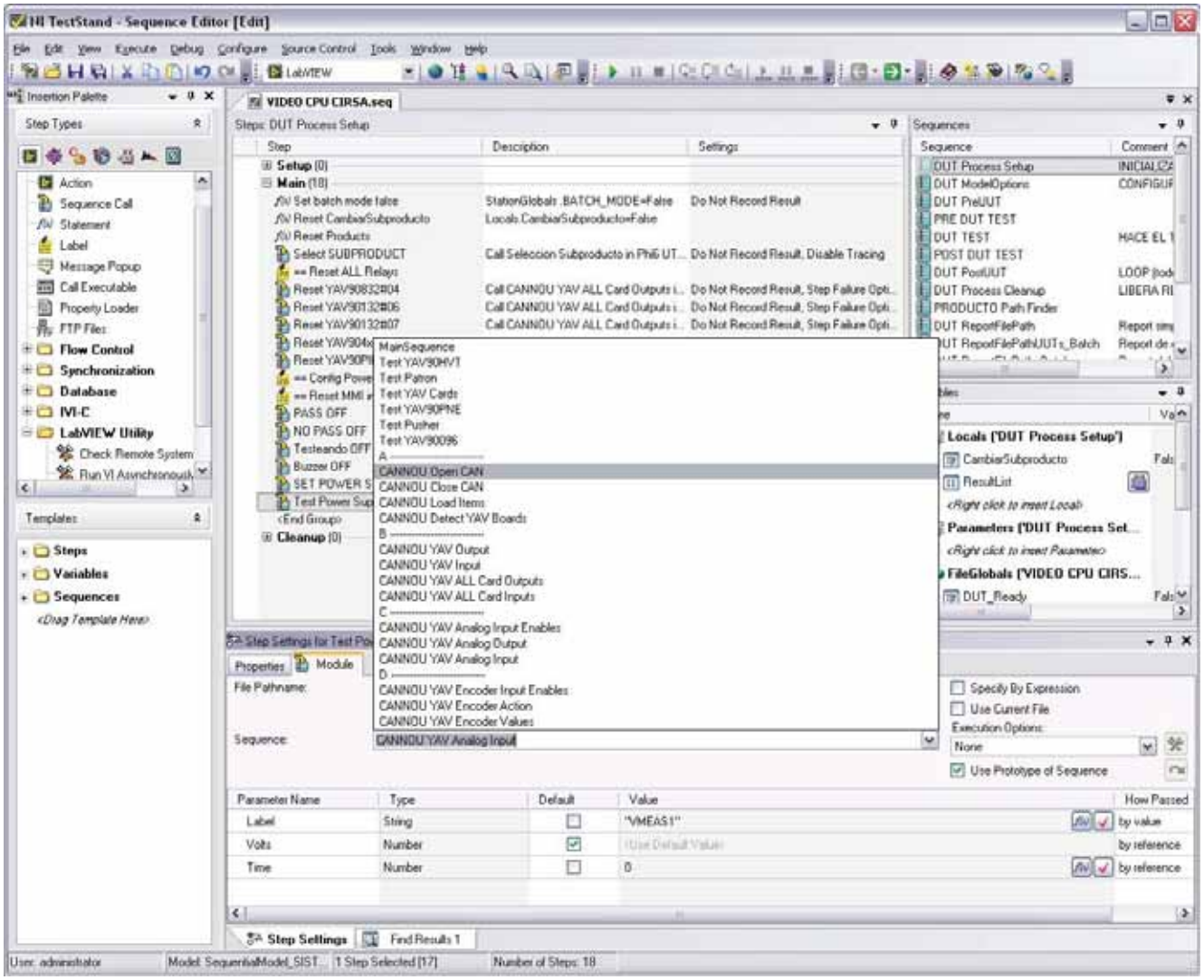
C:\Documents and Settings\All Users\Documents\6TL\CommonDrivers\Phi6 TestStand

Within the CANNOU.seq there is a sequence for each of the functions available to the YAV boards.

In the following example the 'CANNOU YAV Analog Input ' sequence is used to read an analog signal. This sequence is equivalent to the 'YAV GET ANALOG INPUT.vi' of the LabVIEW Phi6 drivers seen in section 5:



Samewise, any of the other Phi6 functions can be selected:



7. Phi6 Operator Interface

The Operator Interface is contained in the executable Phi6-Platform.exe, which controls and manages the operations of all 6TL platforms.

7.1 Set-up and configuration

7.1.1 TestStand configuration

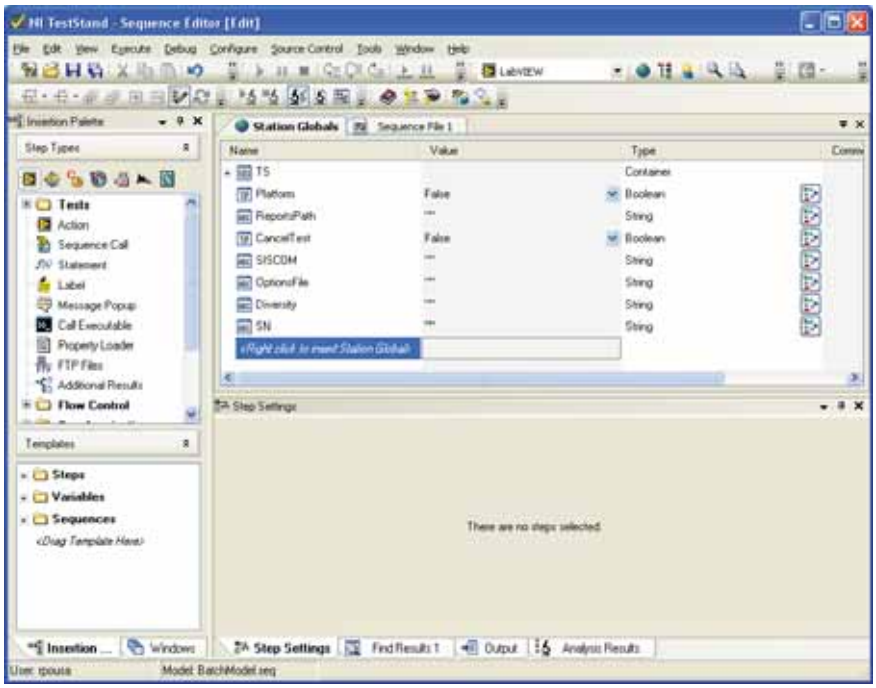
Follow these steps to configure TestStand to be ready to work with the Phi6 Operator Interface:

7.1.1.1 Add StationGlobals variables

1. Off-line Test Platforms (6TL-19, 22, 29): Platform (boolean), ReportsPath (string), CancelTest (boolean):

o

2. Online Test Platforms (6TL-31, 32): Platform (boolean), ReportsPath (string), CancelTest (boolean), SISCOM (string), OptionsFile (string), Diversity (string), SN (string):



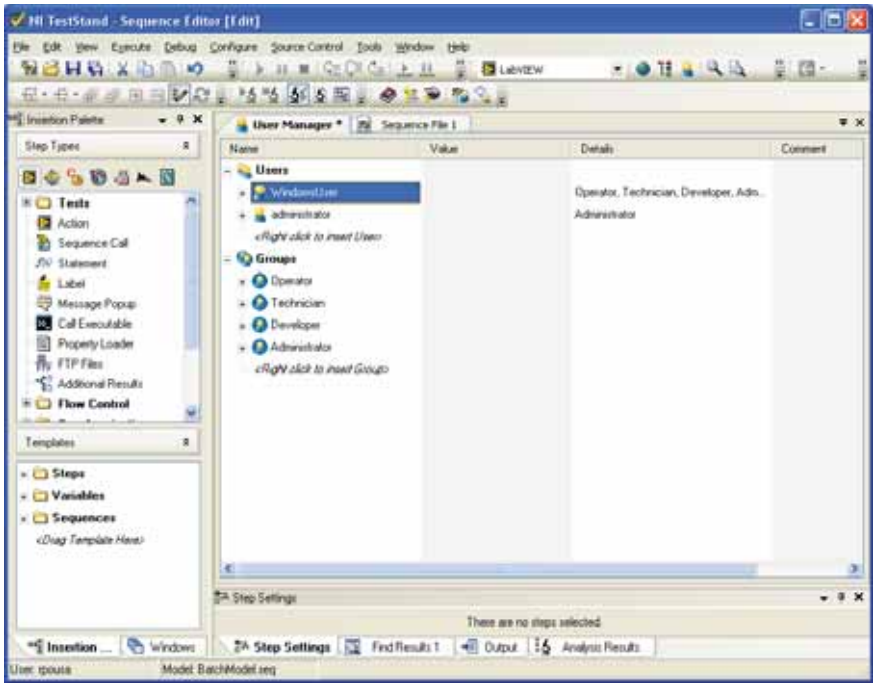
7.1.1.2 Add search directories

Add the following search directories (Configure -> Search Directories):

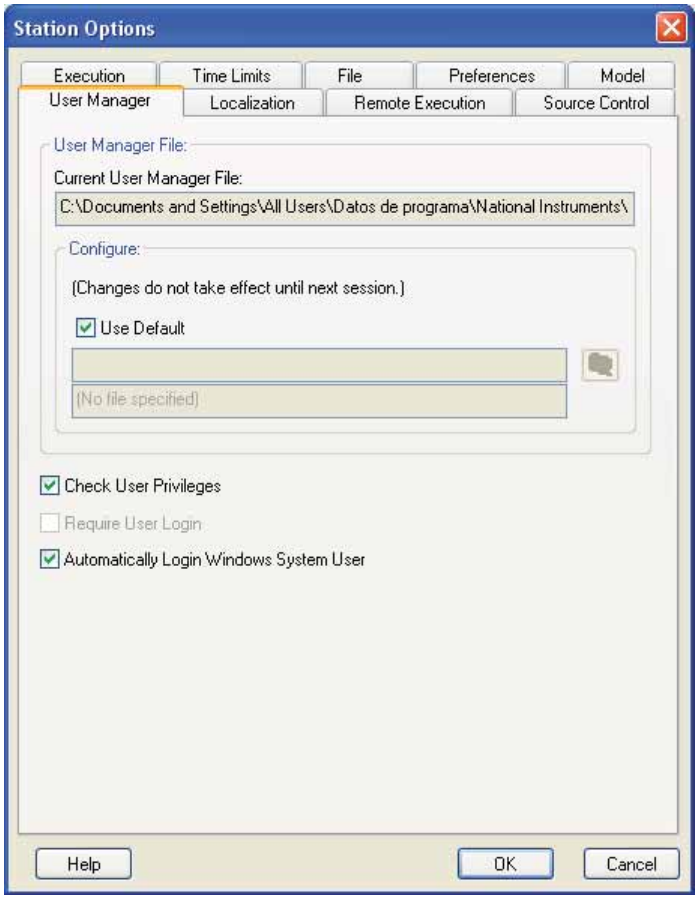
1. C:\Documents and Settings\All Users\Documents\6TL\CommonDrivers\OIUtil
2. C:\Documents and Settings\All Users\Documents\6TL\CommonDrivers\Phi6 TestStand

7.1.1.3 User configuration

In the User Manager palette create a user with the same name as the Windows user:

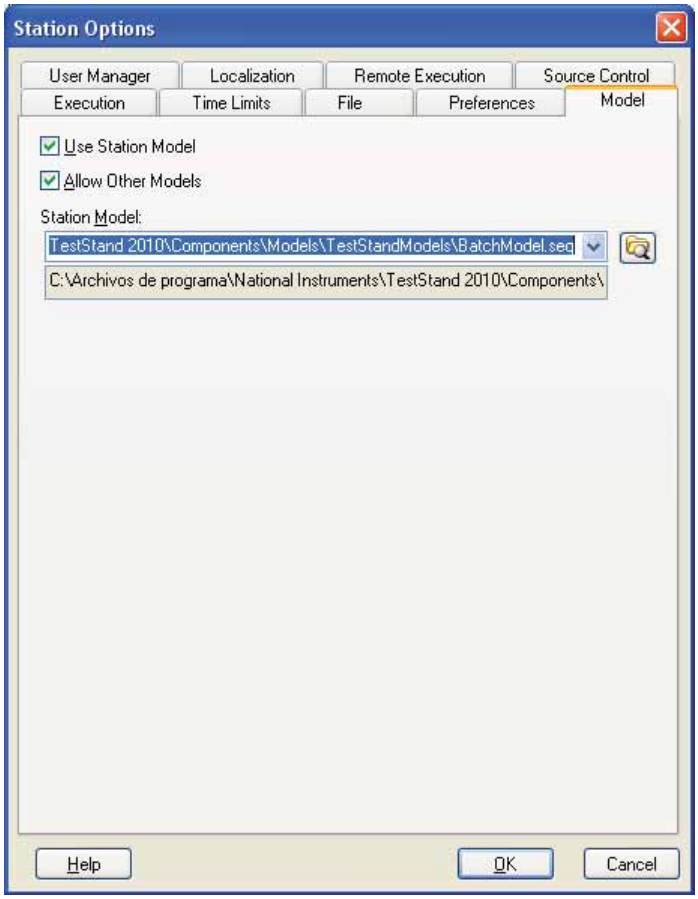


In the Station Options menu, User Manager tab, select the option “Automatically Login Windows System User”:



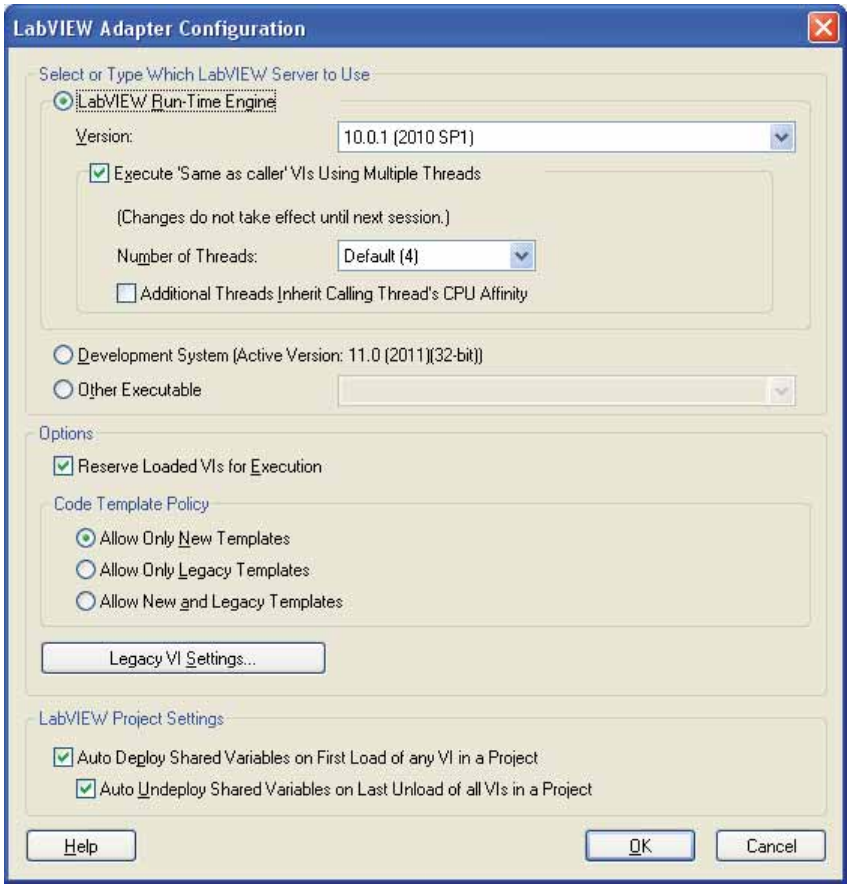
7.1.1.4 Select Batch model

In the Station Options menu, Model tab, select the default TestStand Batch model:



7.1.1.5 Configure LabVIEW Adapter

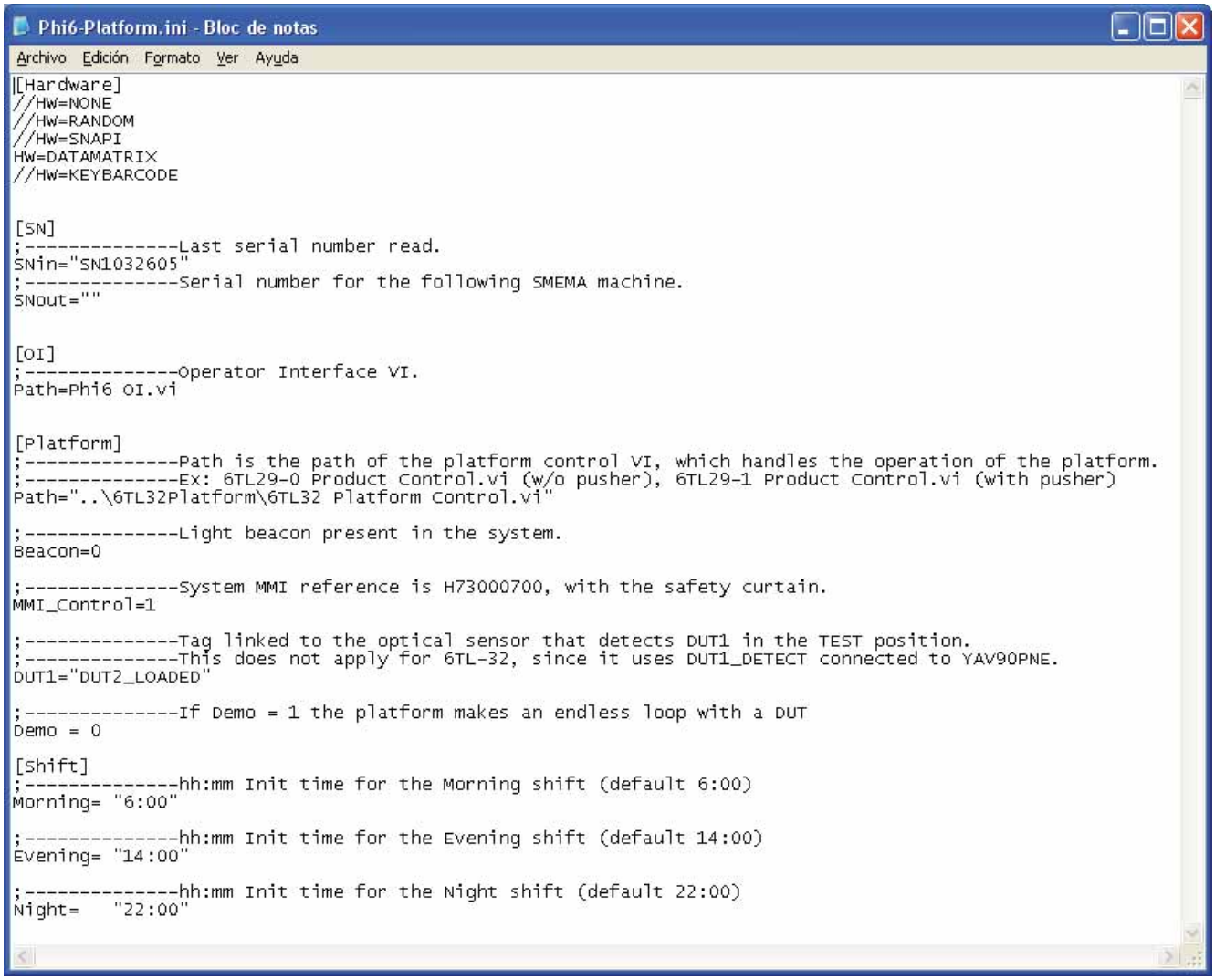
In the Adapters menu, select the LabVIEW Adapter Configuration and choose LabVIEW Run-Time Engine, 2010 SP1 option:



7.1.2 Platform configuration

The Phi6-Platform.ini file is installed with the Phi6 Operator Interface. It configures some important 6TL platform settings.

Example of configuration for a 6TL-32 platform:



[Hardware]: Specifies which hardware is used to read the serial number. Currently 4 options are supported:

1. NONE: no hardware is installed to read the serial number. The application will use the current string of the [SN] → SNin field.

2. RANDOM: no hardware is installed to read the serial number. The application will generate a random number and store it in the [SN] → SNin field and use it afterwards as the DUT's serial number.

3. SNAPI: system uses a Symbol Motorola datamatrix scanner. The application will run a process in the background which reads the serial number and stores it in the [SN] → SNin field. The drivers for this reader are installed in the CommonDrivers\SNAPI folder.

4. DATAMATRIX: an indeterminate datamatrix reader is installed. The end user will add the appropriate code in the TestStand sequence to read from the datamatrix.

5. KEYBARCODE: system uses a barcode reader or the keyboard to enter the serial number. The application will run a process in the background which reads the serial number and stores it in the [SN] → SNin field.

[SN]:

1. SNin: used to store the serial number read for the NONE, RANDOM, SNAPI, KEYBARCODE cases.
2. SNout: used in 2 unit 6TL-32 systems. Since the serial number reader is in the first unit the second one gets this information through this field.

[OI]: path to the Operator Interface VI inside the Phi6-Platform executable. Currently only Phi6 OI.vi is supported.

[Platform]:

3. Path: path of the platform control VI inside the executable, which handles the operation of the platform. Examples:
   - **6TL32**: "..\6TL32Platform\6TL32 Platform Control.vi"
   - **6TL29:** "..\6TL29-0Platform\6TL29-0 Platform Control.vi" → without pusher.
     "..\6TL29-1Platform\6TL29-1 Product Control.vi" → with pusher.
   - **6TL22:** "..\6TL22-0Platform\6TL22-0 Platform Control.vi" → no VPC Receiver, no Servo-Pusher.
     "..\6TL22-1Platform\6TL22-1 Platform Control.vi" → with VPC Receiver, no Servo-Pusher.
     "..\6TL22-2Platform\6TL22-2 Platform Control.vi" → with VPC Receiver and Servo-Pusher.
   - **6TL19:** "..\6TL22-0Platform\6TL22-0 Platform Control.vi".

4. Beacon: specifies if a 6TL light beacon is installed in the system.

5. MMI_Control: specifies which version of the MMI is present in the system. For H73000700 the value should be set to 1. For H73000300 the value should be set to 0. This parameter only affects the display of the MMI, it does not affect the functionality of the MMI.

6. DUT in test position sensor: this is used only in 6TL-29 systems. It refers to the tag of the optical sensor that indicates that a DUT is in the position to be tested. If the field is left blanked no sensor is being used to signal DUT presence. Otherwise, the program will use the digital signal of the sensor as a START signal to run the test sequence.

7. Demo: for in-line test platforms only. If demo = 1 the platform makes an endless loop with the DUT, executing the complete test cycle continuously until the CANCEL button is pressed.

[Shift]: defines the 3 production shifts starting time.

7.1.3 Product configuration

Configure the ProductName.ini file located inside the product folder.

Name: name of the product (fixture). It MUST match the product folder name and the product configuration file (.cfg).

Description: what is written here will appear in the Description field of the fixture detection page of the Operator Interface.

TestSequence: exact path to the TestStand test sequence that the Operator Interface must load.

TS_Version: TestStand version being used. TS 2010 corresponds to version 4.5. Currently only version 4.5 is supported.

Station_Model: Path to the TestStand model being used. This is located inside the TestStand folder version specified by 'TS_Version'. Currently only Batch Model is supported.

Sockets: Number of DUTs to be tested in parallel. The only allowed values are 1 and 2, since the Operator Interface allows a maximum of 2 DUTs in parallel.

ITACode_Master: Main fixture ID, stored in the fixture YAVCANCON.

ITACode_Slave: Servo-Pusher ID, stored in the servo-pusher YAVCANCON. This is only used for platforms with servo-pusher.

Pusher_Level: specifies the position at which the Servo-Pusher returns after performing a DUT test. In most applications it returns to the highest (initial) position, which in the standard 6TL_PLATFORM.cfg file is defined as RETURN; in some applications it must return to some intermediate position in order to shorten the cycle time.

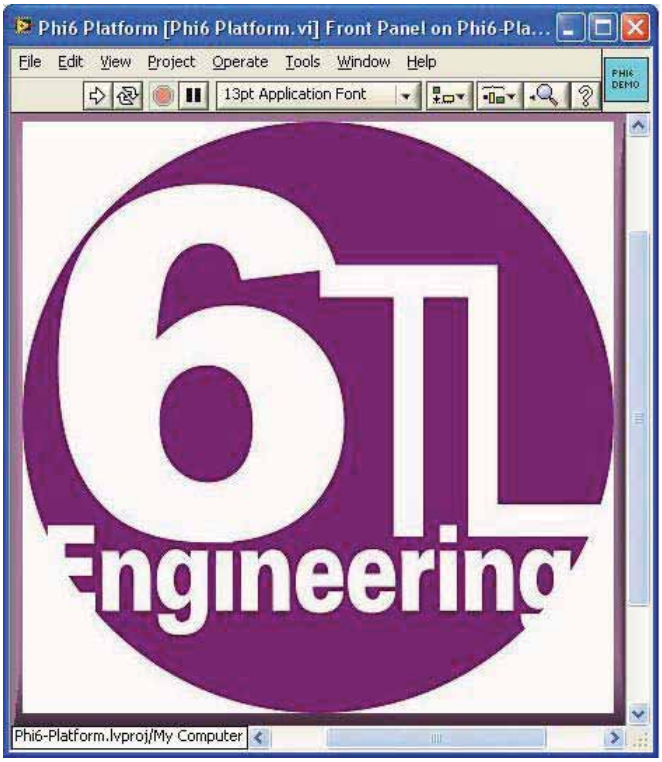
Reader: this variable applies only for systems with 2 SNAPI Datamatrix readers. In those systems typically there is one reader on top and one at the bottom of the platform. To use the bottom reader set this value to 0; to use the top reader set this value to 1.

The example product 'Product1' shows an example of correct configuration:

```
[PRODUCT]
Name=Product1
Description="Product1"
TestSequence="C:\Documents and Settings\All Uers\Documents\6TL\Products\Product1\Sequences\Product1.seq"
TS_Version=4.5
Station_Model="Components\Models\TestStandModels\BatchModel.seq"
Sockets=1
ITACode_Master= 0000100101
ITACode_Slave=  0000100110
Pusher_Level=RETURN
Reader=0
```

7.2 Graphical User Interface

After running the executable the Phi6 Platform engine will start up. It will open the Operator Interface and the platform control program specified in the Phi6-Platform.ini file. Afterwards this engine runs in the background:



The initial screen of the Operator Interface is the fixture detection page:

The first operation is to set the platform in its init conditions. Depending on the hardware installed the following actions will be performed:

1. Move the Servo-Pusher to the Return position.
2. Move the Lifter to the Exchange position.
3. Set the Conveyor to its initial conditions.

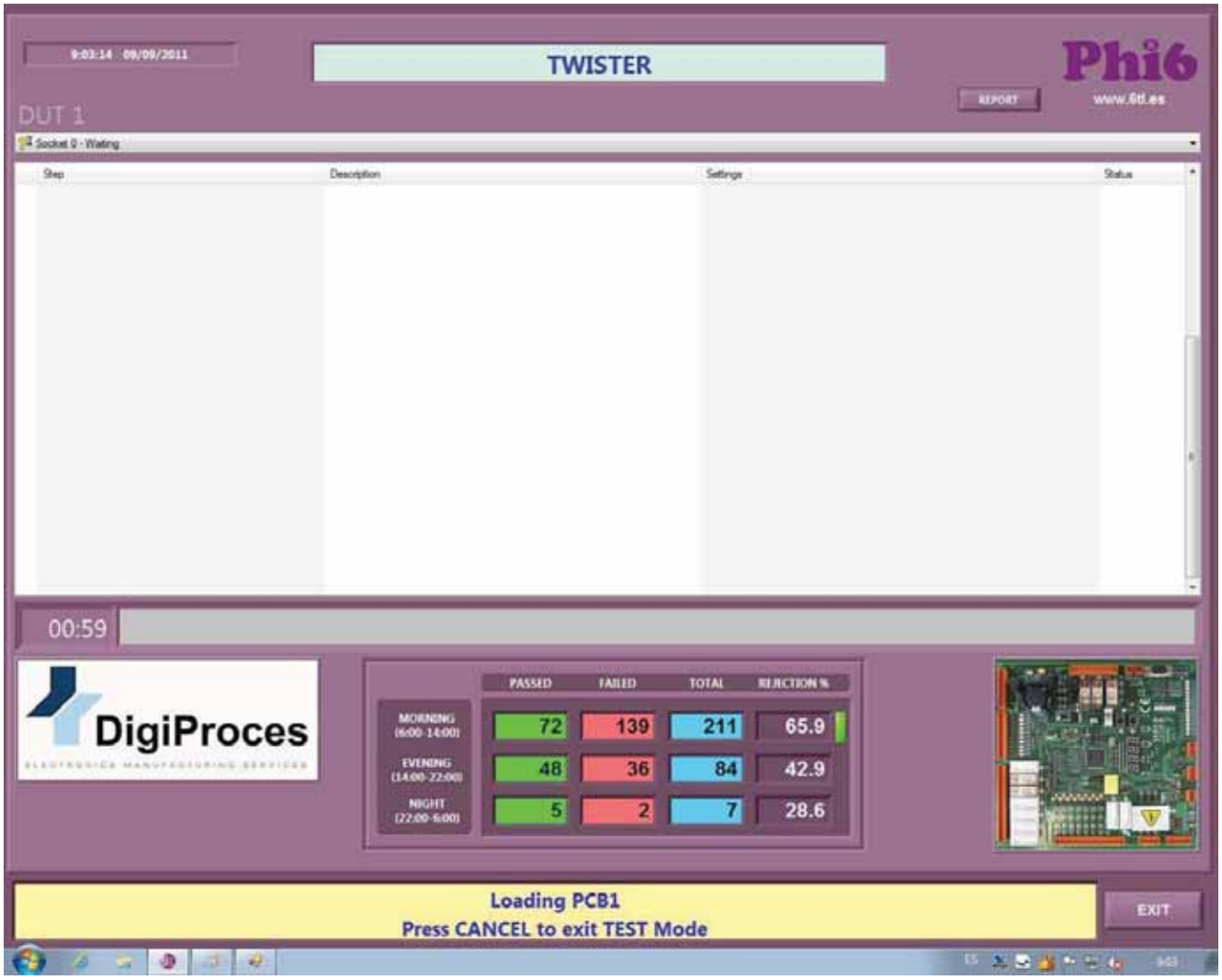
In the platforms with Virginia Panel Receiver, the program first detects that the MASTER_IN tag is 'True'; the tag is connected to the micro-switch of the Receiver, so it is an indication that a fixture has been inserted.

Next the program reads the reference ID of the fixture by reading the ID_MASTER of the 6TL_Fixture_ID_Master board (YAVCANCON). Then it performs a search in all the products available stored in the 6TL\Products folder to find a Product.ini file with the reference ID that has been previously read.

At this point the program knows which product to test and displays all the Product.ini information. Also, if an image named ProductX.jpg is present in the product folder it will also display the picture of the device under test.

Note also that an image of the clients' logo can also be included; in the above example the client is DigiProces. That image file has a fixed non-editable name 'Manufacturer.jpg' and is located in the same folder as the executable (C:\Program files\6TL).

After pressing START button on the MMI the program switches to the TestStand GUI screen.



The sequence execution is displayed as well as metrics of passed, failed and total devices been tested.

The start of the product TestStand sequence depends on the system platform:

1. 6TL-19, 6TL-22: a window will show up asking the operator to press the START button of the MMI. When the test sequence is finished the window will appear again to proceed with the test of the following DUT:

2. 6TL-29 without Servo-Pusher: if DUT in test position sensor is present it will first make sure no DUT is present. Then it checks that a device has been inserted into the test position (CURTAIN tag is on, meaning light barrier is free, and DUT in test position tag is on). Finally, it makes sure the tag DUT1_LOADED is on, which assures there is good contact between the DUT and the test needles. When the test finishes it waits until the device has been removed from the test position (CURTAIN tag is off, meaning light barrier is not free, and DUT in test position tag is off).

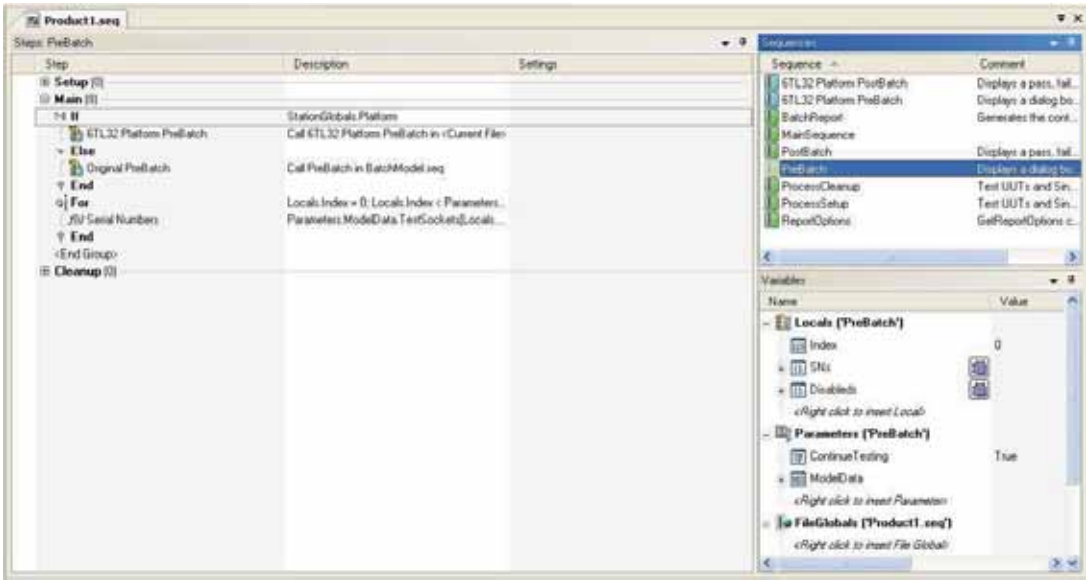
3. 6TL-29 with Servo-Pusher: if DUT in test position sensor is present it will first make sure no DUT is present. Then it waits until the external bi-manual start push-buttons have been pressed and checks that a device has been inserted into the test position (CURTAIN tag is on, meaning light barrier is free, and DUT in test position tag is on). Afterwards, it moves the servo-pusher to the test (lower) level. Finally, it makes sure the tag DUT1_LOADED is on, which assures there is good contact between the DUT and the test needles. When the test finishes it checks that the light barrier is free and moves the servo-pusher to the Return position. Finally it waits until the device has been removed from the test position (CURTAIN tag is off, meaning light barrier is not free, and DUT in test position tag is off).

4. 6TL-32: once a board has been detected at the entrance of the conveyor the program will handle all necessary operations to place the board in the test position and start the product TestStand sequence.

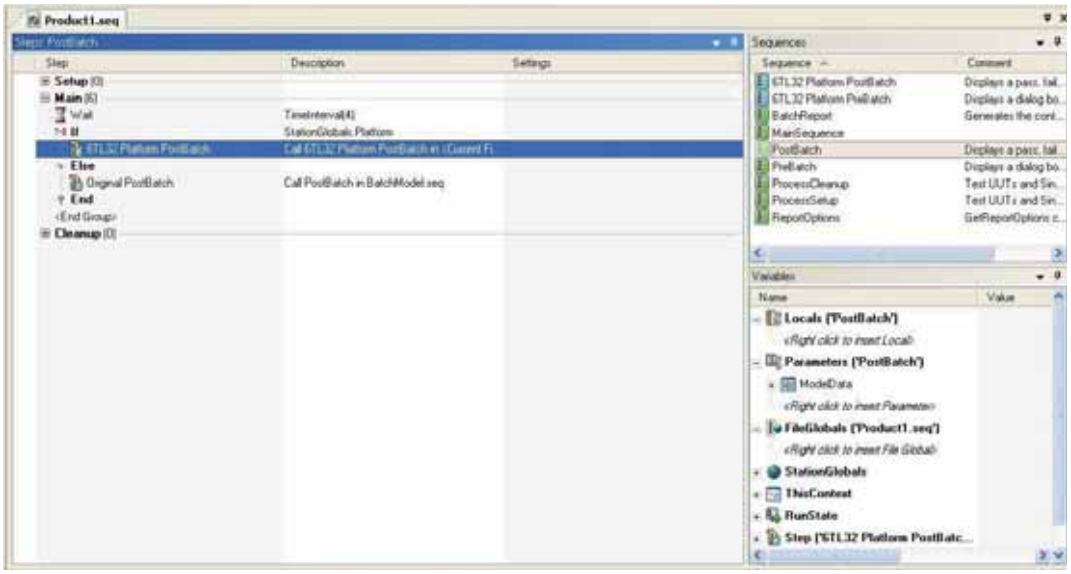
7.3 Product TestStand sequence specification

In order to be able to communicate with the Operator Interface the sequence must include some 6TL defined steps. These steps contain shared variables that are used for this communication.

1. Use the 'Phi6-OI PreBatch.vi' in the PreBatch sequence. The VI is located in the C:\Documents and Settings\All Users\Documents\6TL\CommonDrivers\OIUtil folder.



2. Use the 'Phi6-OI PostBatch.vi' in the PostBatch sequence. The VI is located in the C:\Documents and Settings\All Users\Documents\6TL\CommonDrivers\OIUtil folder.



3. Use the 'Phi6-OI After Cleanup.vi' in the Process Cleanup sequence. The VI is located in the C:\Documents and Settings\All Users\Documents\6TL\CommonDrivers\OIUtil folder.

As a starting point to develop the application it is recommended to use the Product1.seq located in the Product1 folder, which serves as an example application.

8. Packaging

El producto, protegido en bolsa antiestática P/N C5006105, se suministra en caja de embalaje P/N 695023 (250x300x80 mm), con la etiqueta colocada en la posición de la fotografía. La documentación técnica en formato A4 se sitúa en la parte inferior de la caja.
La caja de cartón se cerrará y sellará con cinta adhesiva transparente solo por la parte inferior.

The product, protected with an antistatic bag P/N C5006105 will be supplied into packaging box P/N 695023 (250x300x80 mm), with label placed on the position shown in the photo. The technical documentation in A3 format will be placed on the bottom part of the box.
The cardboard box will be closed and sealed with transparent self-adhesive tape only in the bottom part.