TICS Pro Automation with LabView

By: Tam Tran (t-tran3@ti.com)

Purpose: To automate tests or experiments with TICS Pro software using LabView.

LabView Automation Requirements:

1) Software toolbox consisting of the following LabView subroutines (Virtual Instruments):

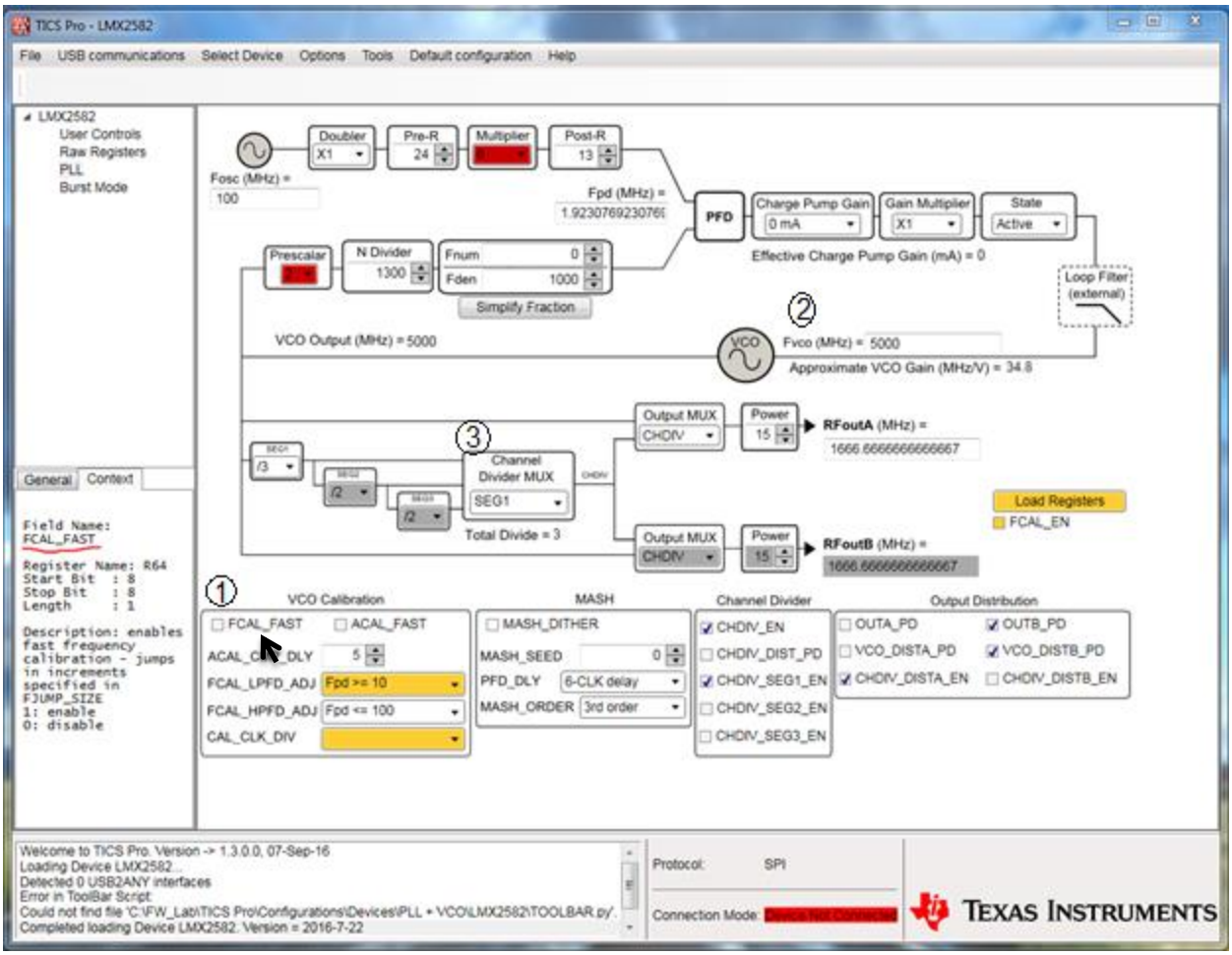
1) TicsPro_initialize.vi
2) TicsPro_close.vi
3) TicsPro_set_drop_down_index.vi
4) TicsPro_set_field_value.vi
5) TicsPro_set_text_only.vi
6) TicsPro_Example.vi

2) EVM Module

Software Overview

The LabView VI's included in this package allow the user to automate their tests or experiments with the TICS Pro software. This software package provides access to all user-accessible settings. The current limitation is that the user can not know if they are setting an invalid parameter or value during automation.

In order to control a setting on the TICS Pro software, the user will have to know the name of the parameter they want to control and how that parameter can be controlled. See below for details.

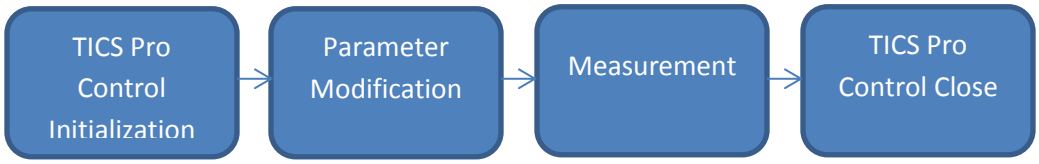


With the TICS Pro software is open, when the user hovers their cursor over a setting, the field name will show on the left. In the picture above, when the cursor is over the "FCAL_FAST" parameter, the field name will show on the left, as underlined in red.

1) For parameters like "FCAL_FAST", the "TicsPro_set_field_value.vi" has to be used to modify it.
2) For parameters like "FVCO (MHz)", the "TicsPro_set_text_only.vi" has to be used to modify it.
3) For parameters like "Channel Divider MUX", the "TicsPro_set_drop_down_index.vi" has to be used to modify it.

Software Structure

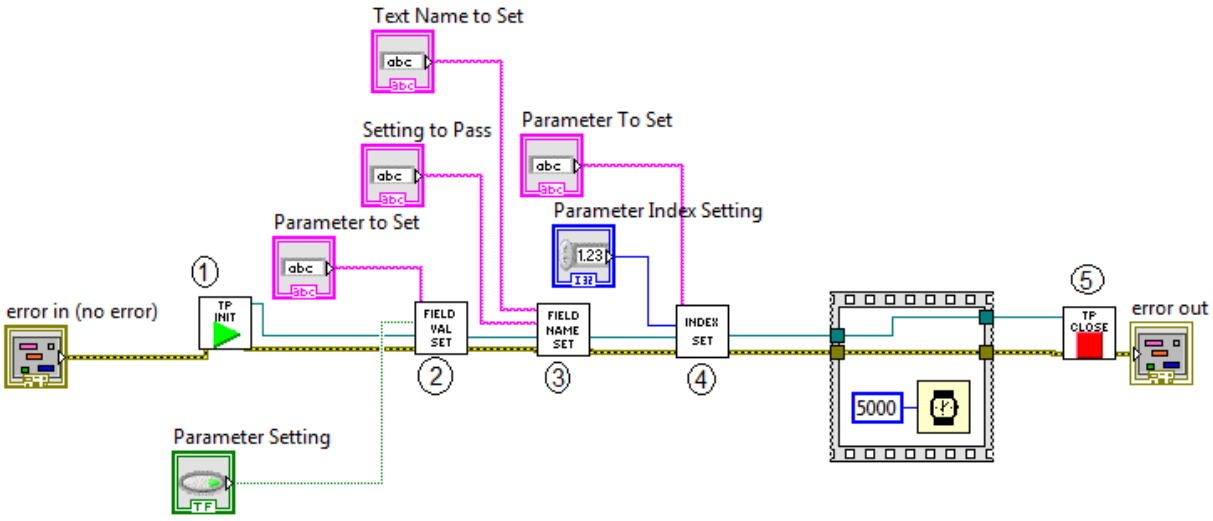
For a typical automation case, the software flow will go as followed:



Software Example

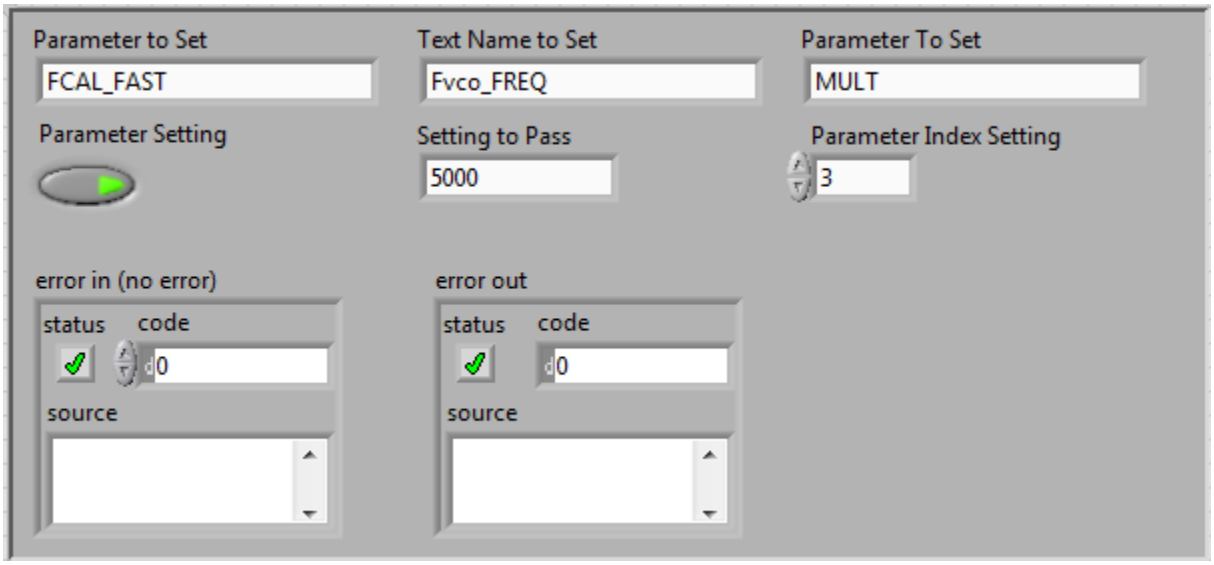
The included LabView example (TicsPro_Example.vi) will show how to toggle the FCAL option, how to select a FVCO frequency, and how to choose a multiplier.

The Block Diagram will look as follows:

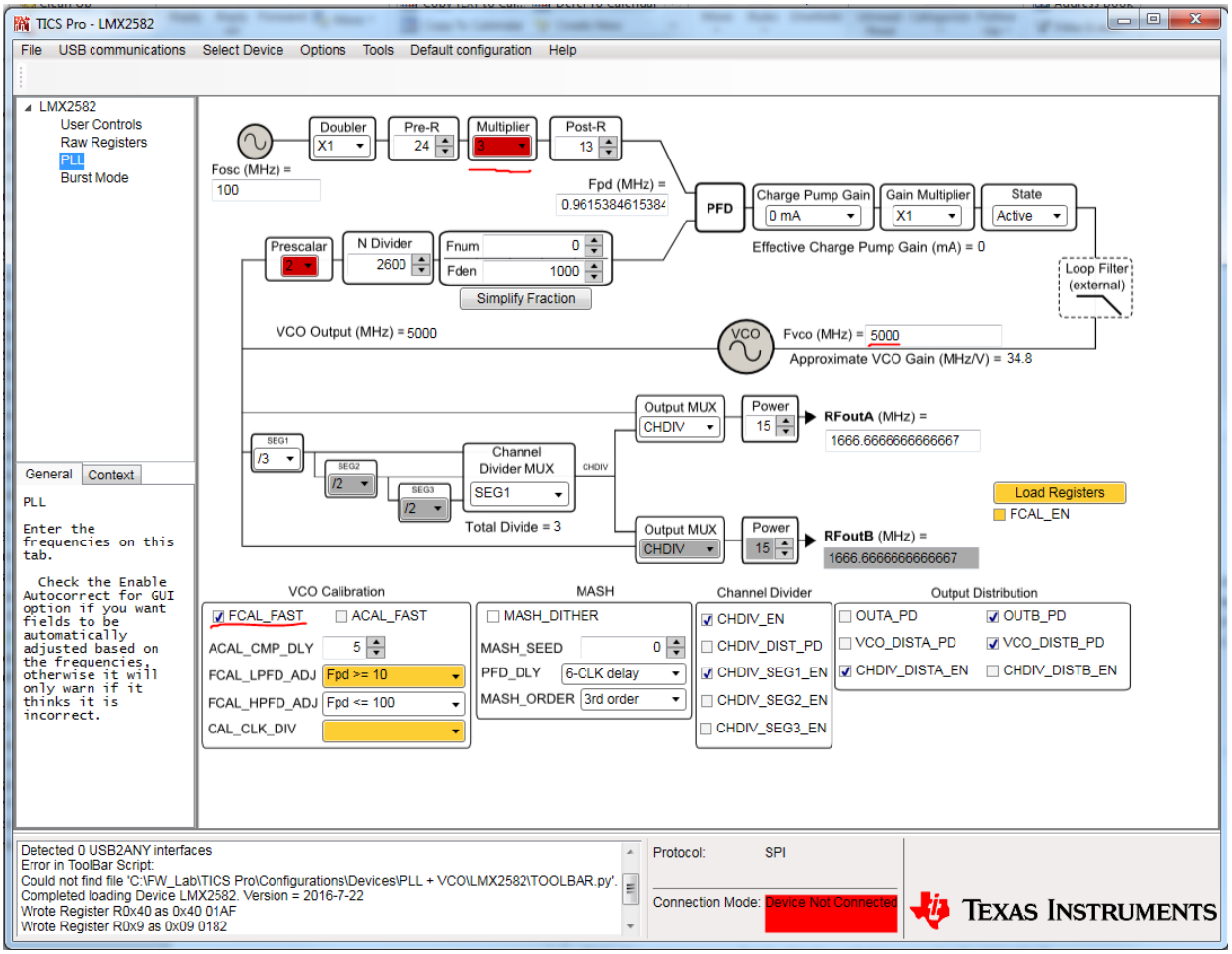


1) The first block, “TP INIT”, is the initialize block which will open up an instance of TICS Pro and pass along the reference of that instance to the rest of the blocks.
2) The second block, “Field Value Set”, is responsible for toggling the True/False settings.
3) The third block, “Field Name Set”, allows the user to modify any text-based setting of a parameter.
4) The fourth block, “Index Set”, sets the index of the user selected parameter drop down menu.
5) The fifth block, “TP Close”, closes the reference to that instance of the TICS Pro software and thus the instance of the TICS Pro software.

The Front Panel will look as follows:



When this LabView software executes, it will enable FCAL_FAST, set the Fvco_FREQ setting to 5000 MHz, and set the Multiplier to 3, as shown below.



Further Development

There are more functions that can be used to further automate the TICS Pro software, such as reading/writing registers, being able to select different devices, etc... The user can implement these features by creating their own LabView VI's by using the invoke nodes of an ActiveX library reference. See below for the full set of controls available in TICS Pro.

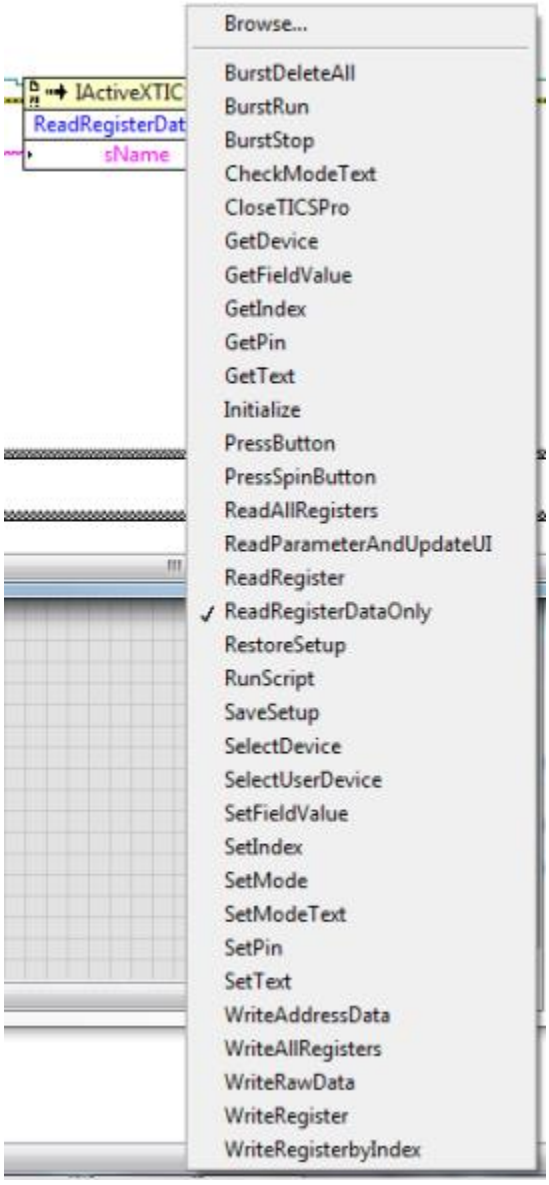
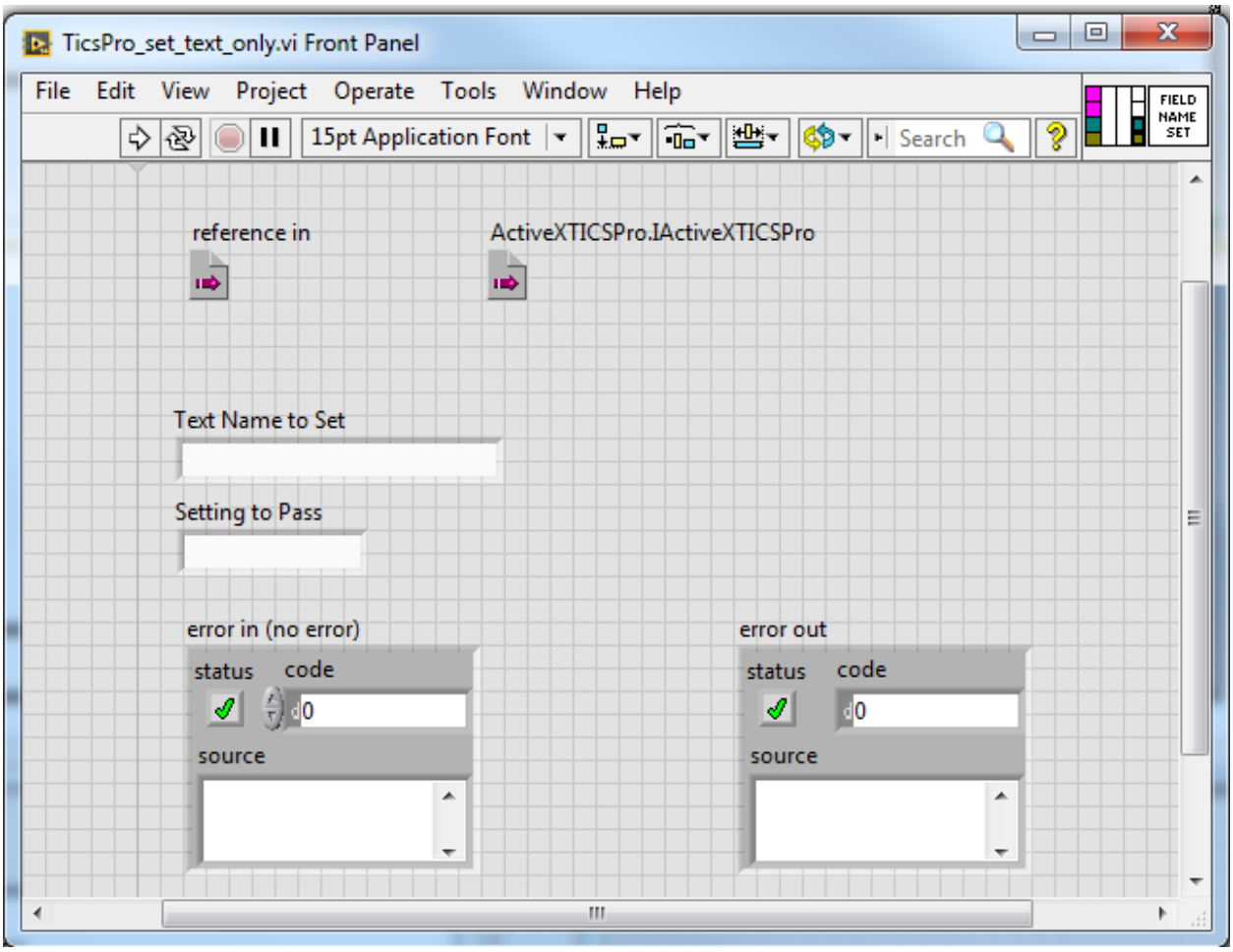


Figure 1

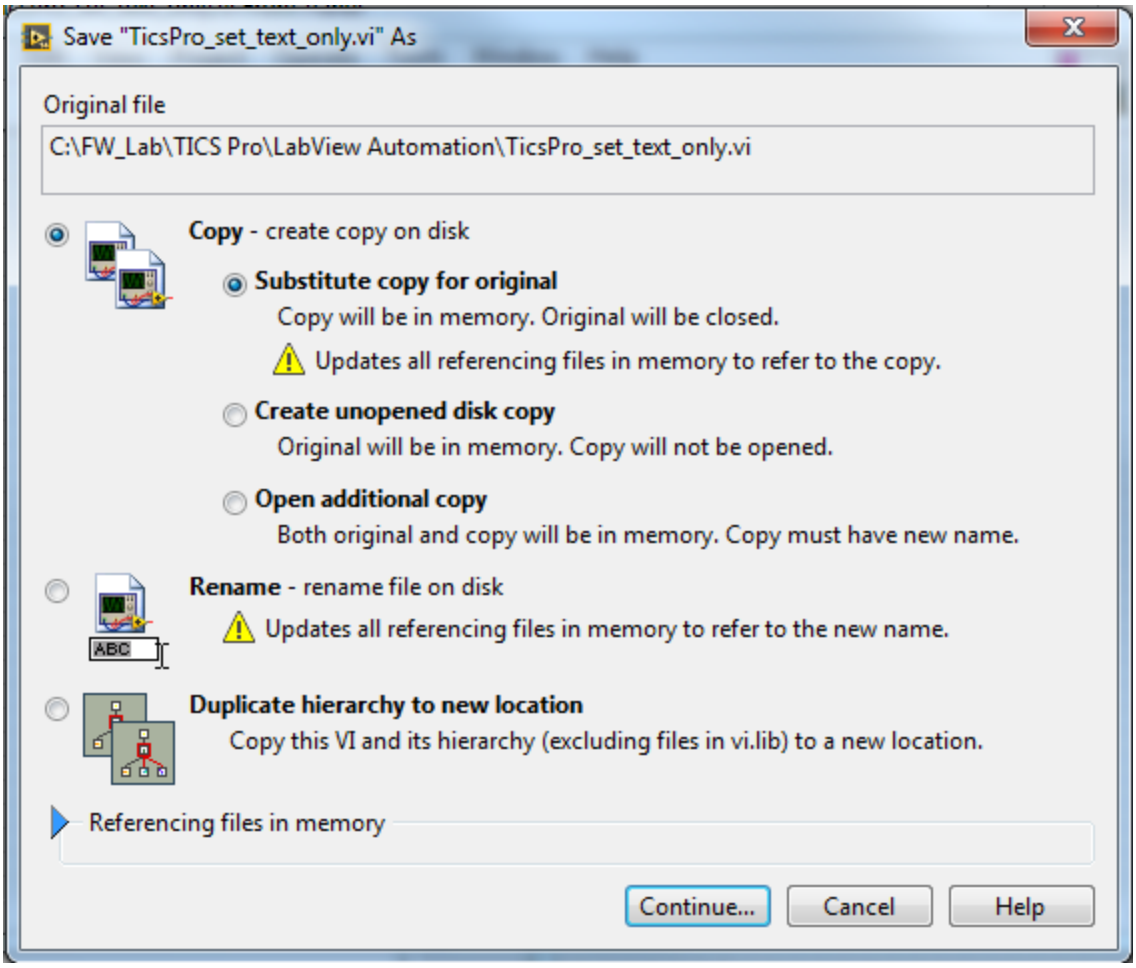
List of Controls Available to the User

Developing More Functionality

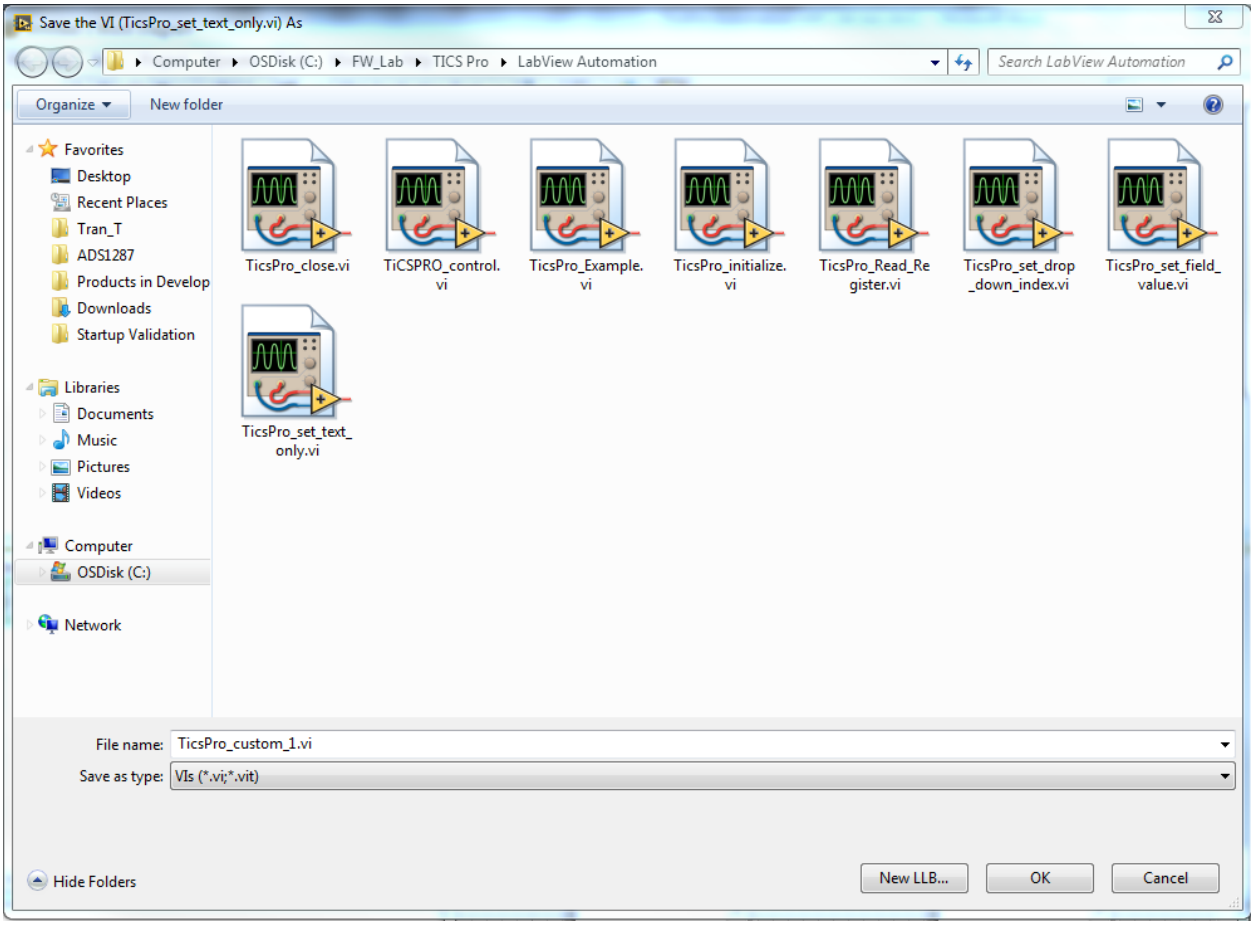
1) First open a preexisting VI, such as the 'TicsPro_set_text_only.vi'



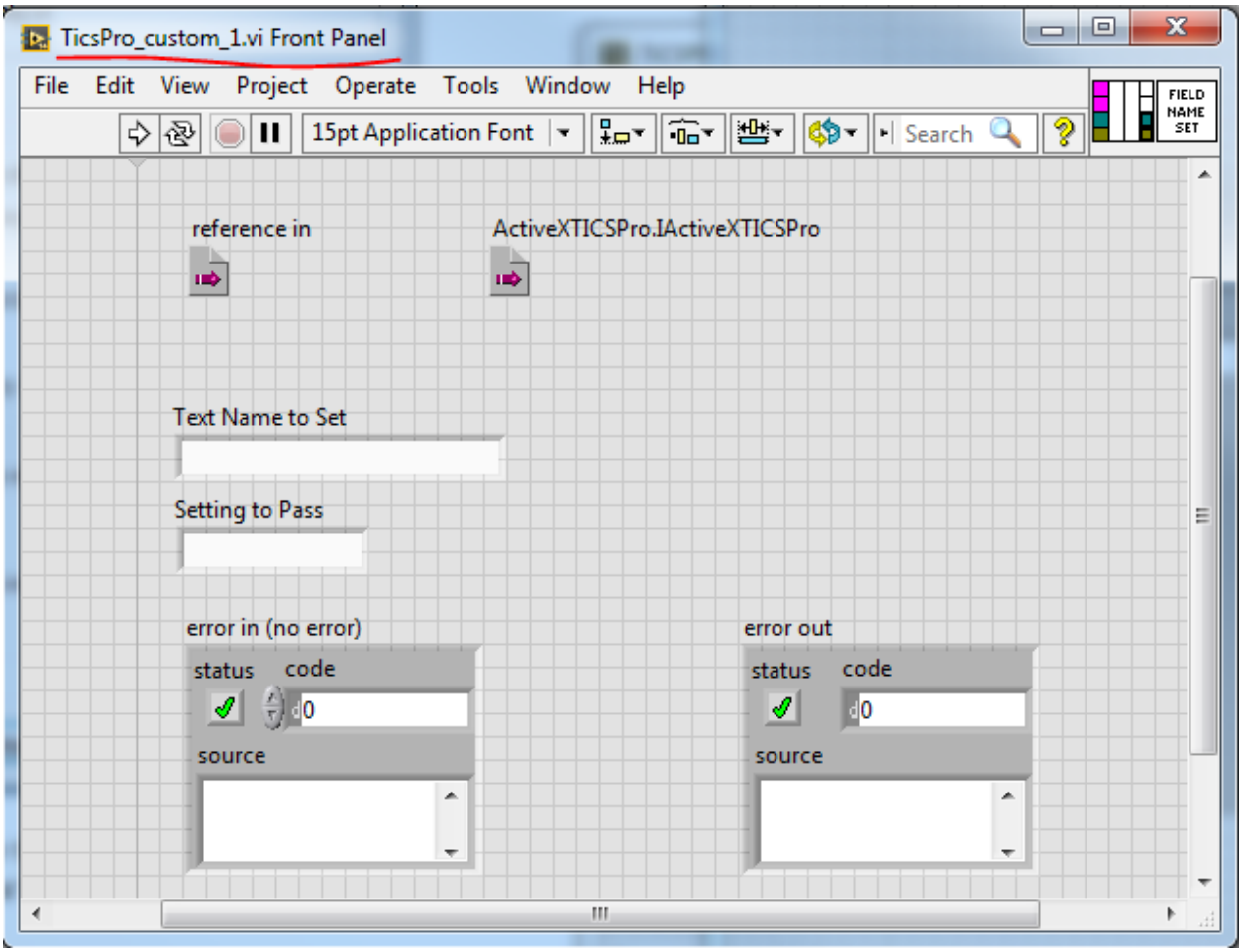
2) Go to File -> Save As -> Substitute Copy for original -> Continue



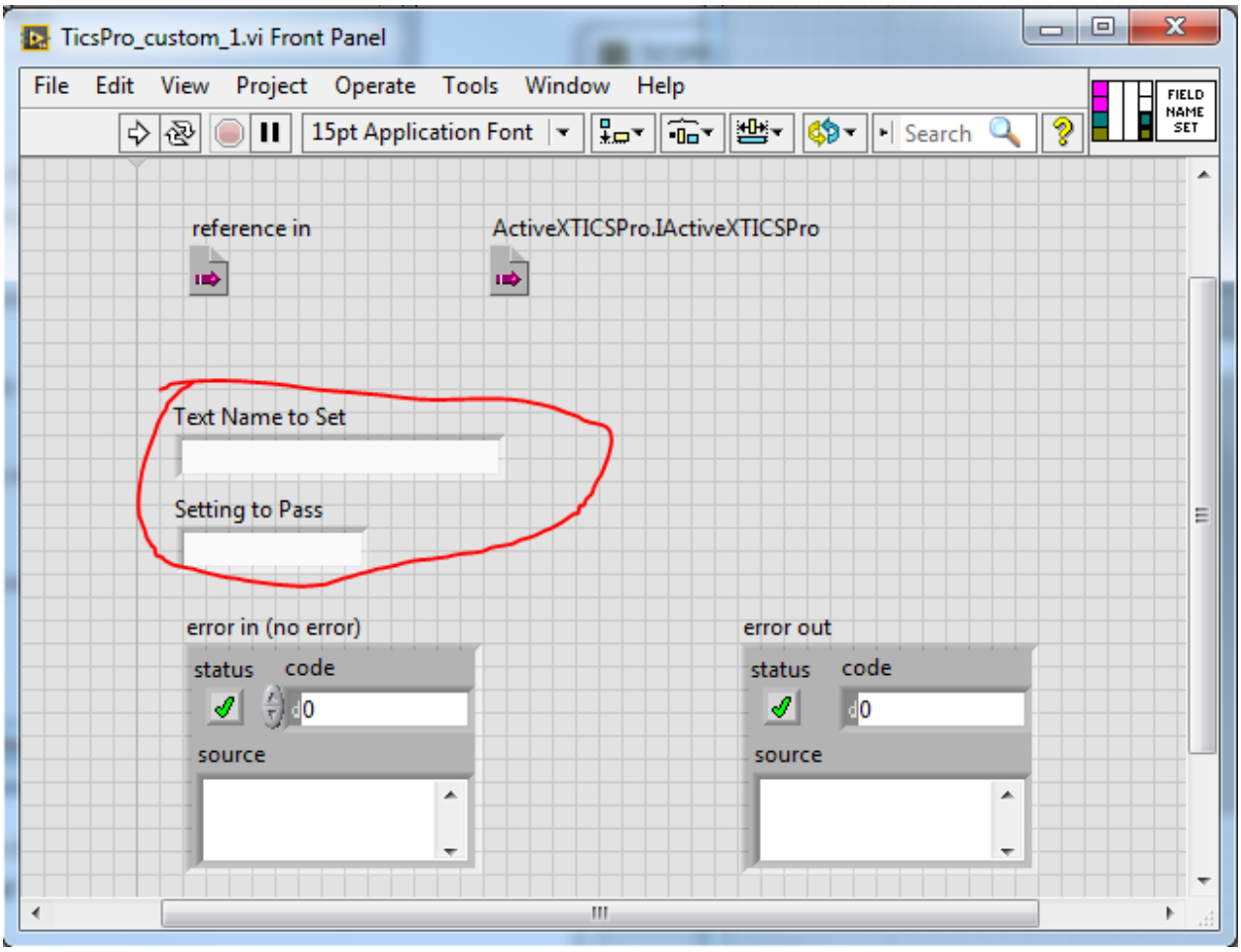
3) Set an appropriate name for the new VI, for the purposes of this example the VI will be named “TicsPro_custom_1.vi” -> then click ‘OK’

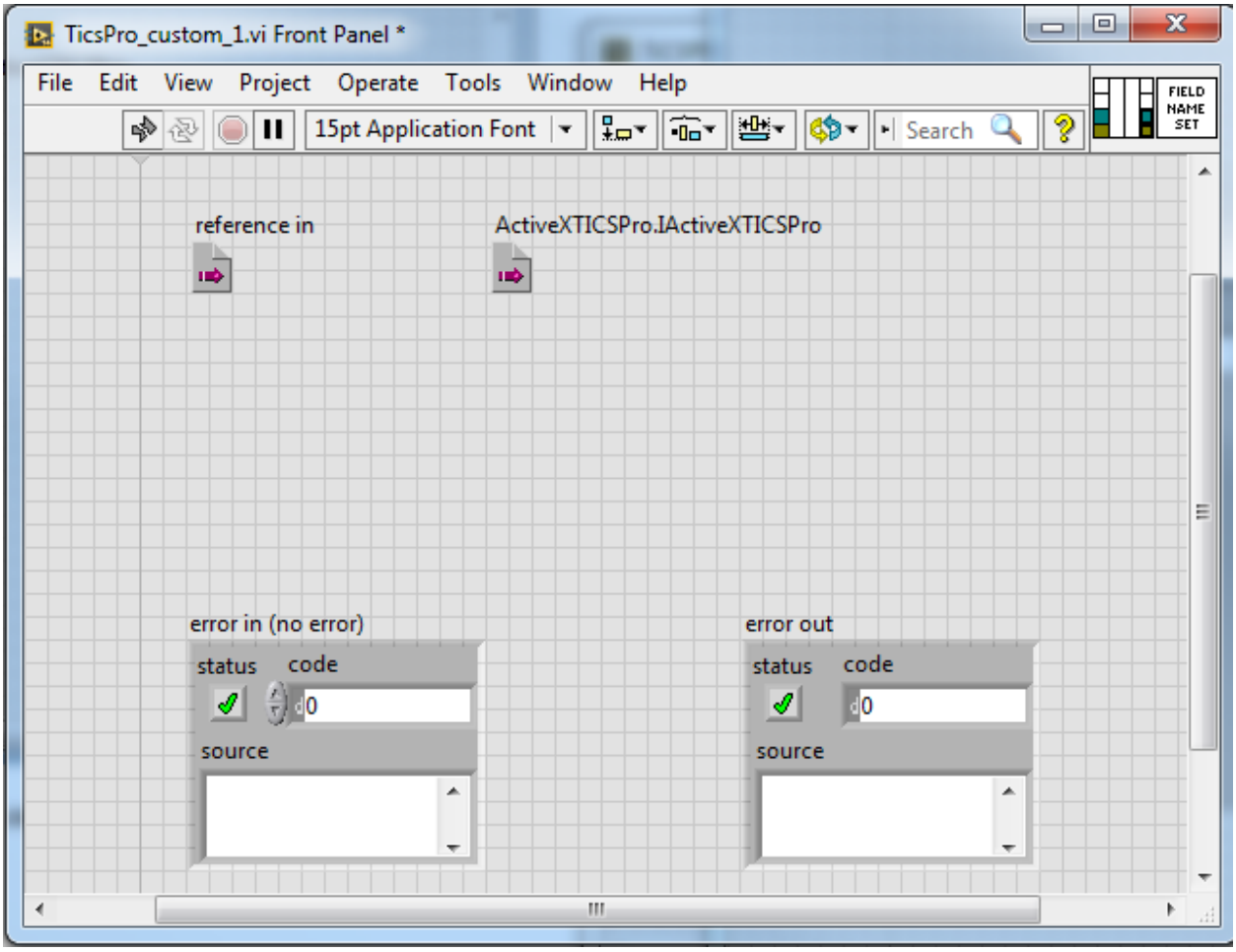


4) The new VI should have an updated name, see the red underline below.

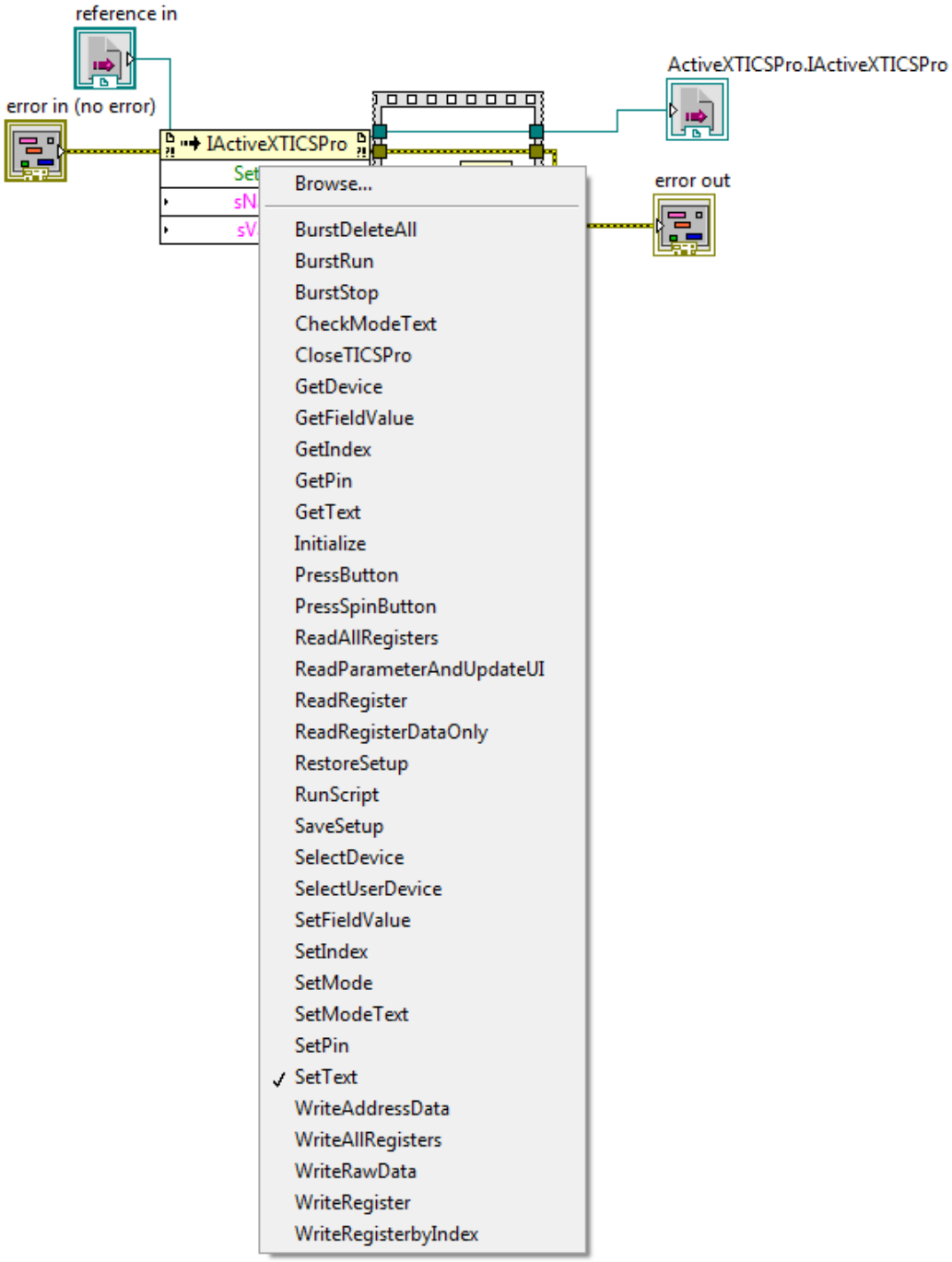


5) The two string controls will not be needed, so delete them from the front panel. Be sure to delete their wire connections in the block diagram as well.

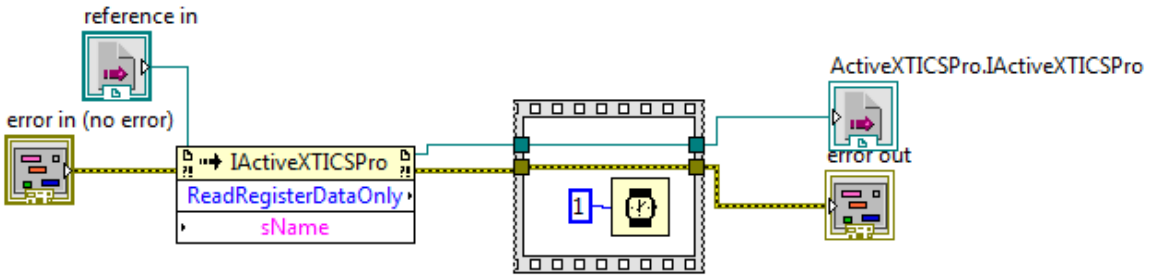

TicsPro_custom_1.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
FIELD NAME SET
reference in
ActiveXTICSPro.IActiveXTICSPro
Text Name to Set
Setting to Pass
error in (no error)
status
code
0
source
error out
status
code
0
source



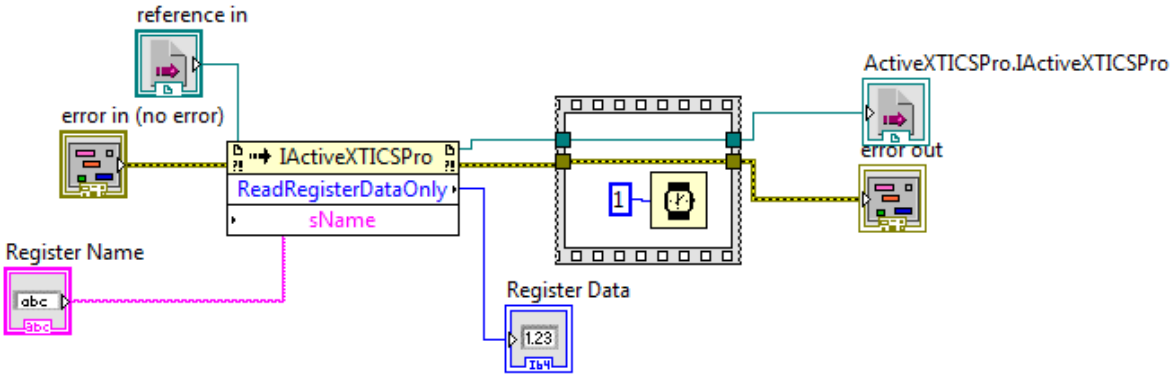
6) Go into the block diagram to modify the ActiveX invoke control, where any of the functions shown above in Figure 1, can be selected. Left click on where “Set Text” is in the invoke node block and the drop down menu should appear as follows

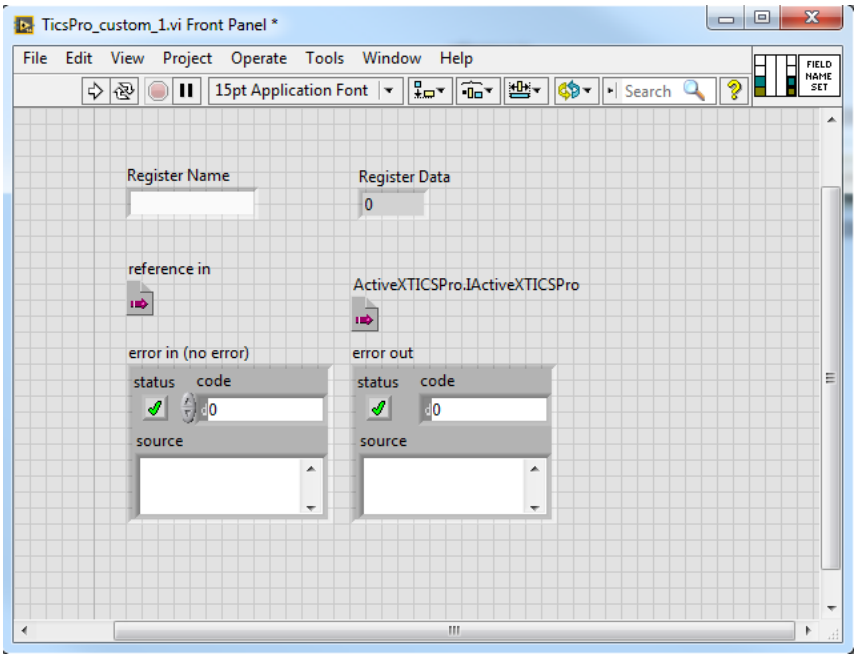


7) From here, any of these functions can be selected. For the purposes of this example, “ReadRegisterDataOnly” will be selected. Once it is selected, the invoke node block will update to show the controls a user can have with that function.

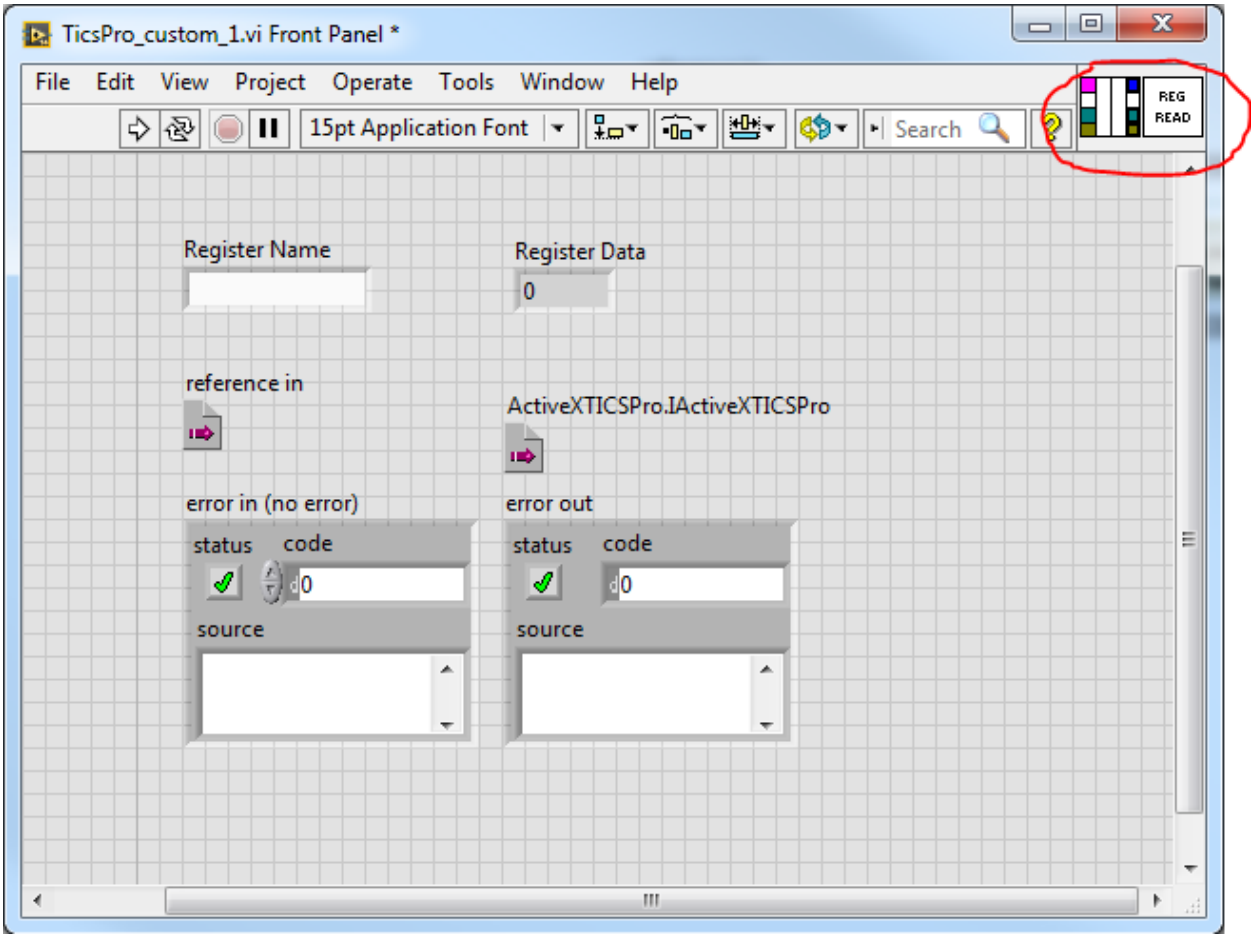


8) There is one string control and a number indicator that the "ReadRegisterDataOnly" function will have. Create a string control and a number indicator in the front panel or the block panel and make sure to connect them to the invoke node. Be sure to name the control and indicator appropriately to prevent confusion in the future.

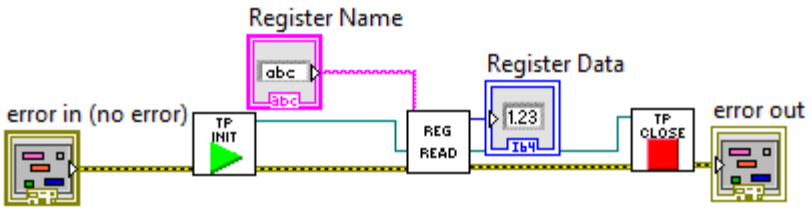


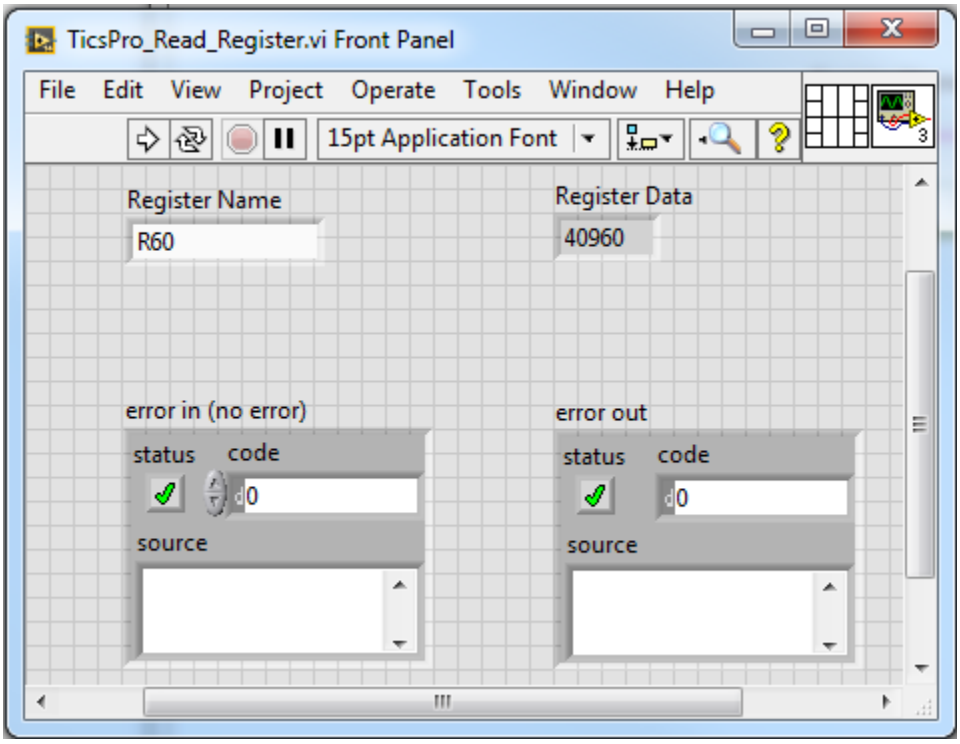


9) Now the VI Icon will have to be updated, as well as its wire terminals. For this example, I will connect the “Register Name” string control to the top left wire terminal and the “Register Data” number indicator to the top right wire terminal.



10) Now this VI is ready, follow the software flow as mentioned above in “Software Structure” to properly control TICS Pro by reading a register. In this example register “R60” is being read from, the register data is in decimal format. *Note: The hexadecimal value can be read by modifying the display format of the number indicator*


Register Name
abc
Register Data
1.23
error in (no error)
TP INIT
REG READ
TP CLOSE
error out


TicsPro_Read_Register.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Register Name
R60
Register Data
40960
error in (no error)
status
code
0
source
error out
status
code
0
source