User's Manual

TICS PRO

Revision History

Revision	Date	Descriptions/Comments
1.0	8-April-2015	Initial version
2.0	19-June-2015	Updated the Color compliant screen shots

The TICS PRO User's Manual provides a general overview of the Evaluation Module (EVM) GUI which includes,

- Description of the features of the GUI
- Functions to be considered while using the GUI

Table of Contents

Table of Figures

Pre-requisites

- Iron Python 2.7.5
- .Net Framework 4.5

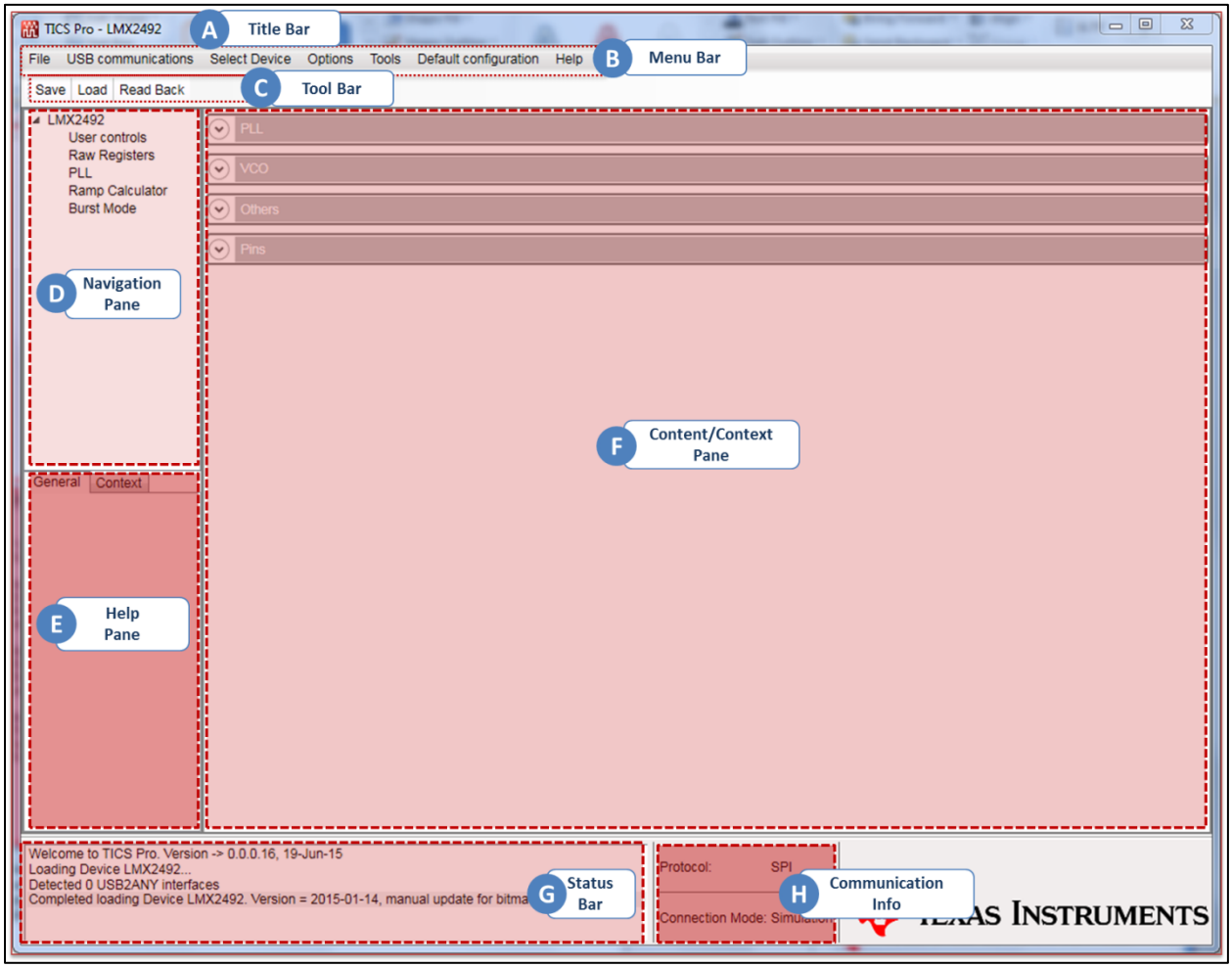
1. TICS PRO GUI Overview

The TICS Pro software provides a generic EVM GUI framework for the LMX, LMK and the CDC timing devices.

1.1. TICS-PRO GUI Layout

In this section, the layout of the TICS Pro GUI is explained.

Figure 1: TICS Pro GUI Layout



A. Title Bar

The TICS Pro GUI provides an option to select the required device. The name of the selected device is displayed in the title bar.

B. Menu Bar

The menu bar provides the option to perform the various operations on the GUI like selecting the device, communication mode, loading or saving configurations etc.

C. Tool Bar

The Tool bar provides the options to perform device specific operations that are defined as Iron Python script. The display of the tool bar is optional in the UI and the menu present varies for different devices. It will be loaded only if there is an entry [TOOLBAR] in the device INI and the scripts specified in the section are displayed as the options in the tool bar.

D. Navigation Pane

The navigation pane lists all the available pages in the GUI. Clicking on an item in the pane will load the corresponding page in the GUI.

E. Help Pane

When you mouse over the controls on the GUI pages, the help pane displays the information relevant to the controls. It has two tabs – General & Context. The general tab provides information about the selected page. The context tab explains the control-specific information.

F. Content Pane

When you select any page in the navigation pane, the content pane is loaded with the corresponding page

G. Status Bar

The status bar displays the status with respect to the different operations being performed on the page.

H. Communication Info

This pane displays the selected communication protocol and whether the GUI is connected to a device or is being used in simulation mode.

2. TICS PRO GUI Menu Options

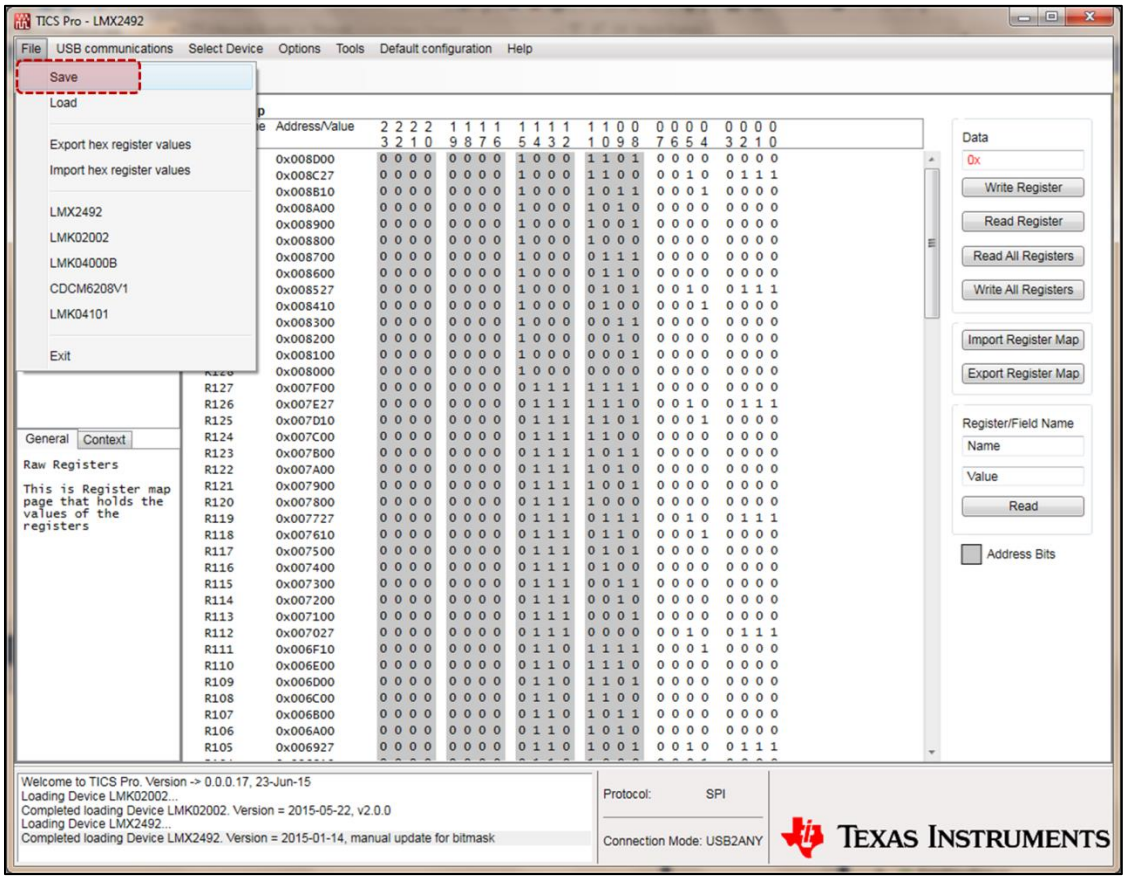
In this section, we will describe the use of the listed menu options

2.1. File Menu

2.1.1. Save

The Save option allows you to save the current configuration of the GUI in a macro file(.mac file). This option can be accessed by traversing through the file menu

Figure 2 : File Menu - Save Option



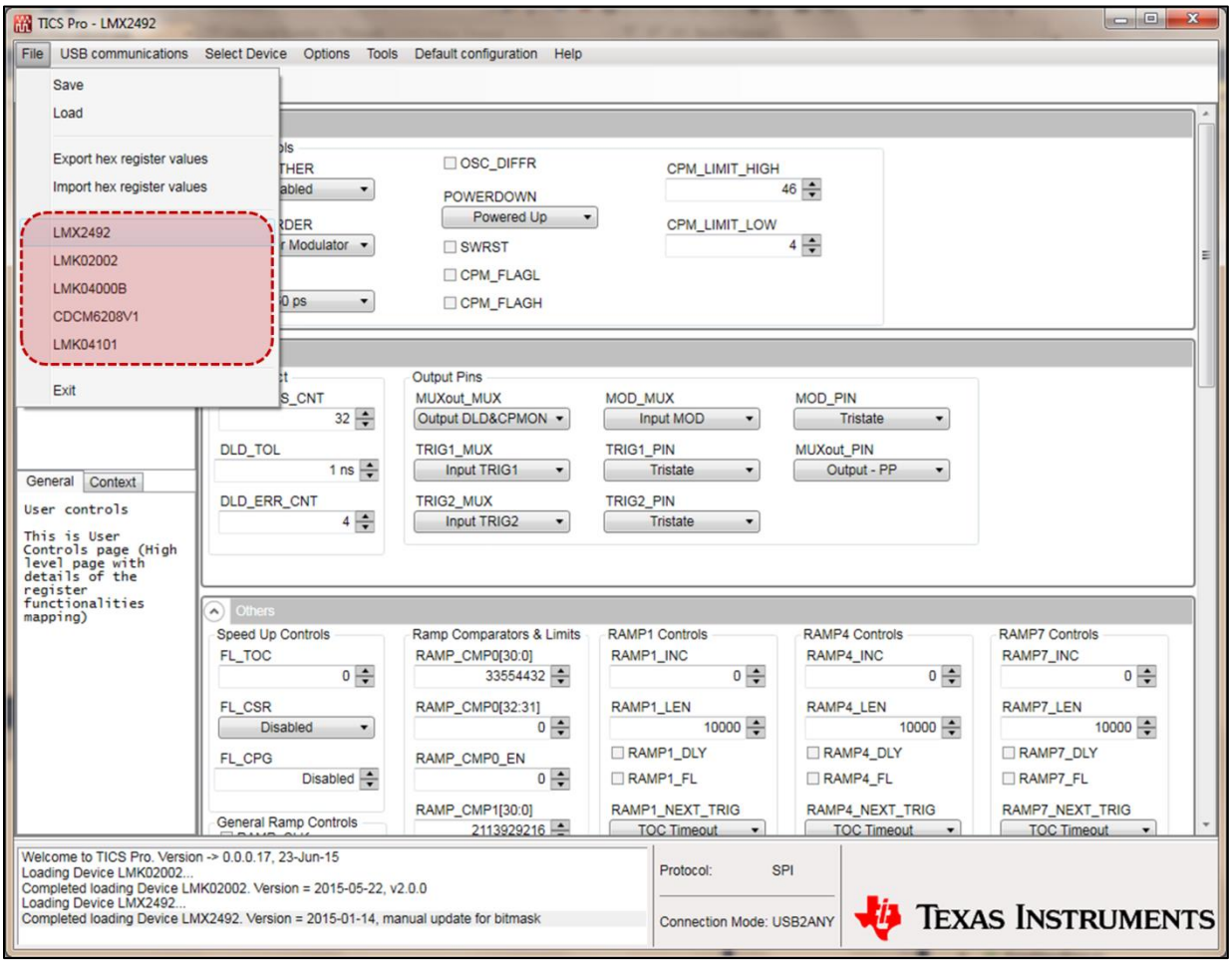
2.1.2. Load

This option allows you to load a configuration which was previously saved. It can be accessed in the following by traversing through the File menu and selecting the "Load" option

2.1.3. Recently Loaded Device Info

The five most recently loaded devices are listed under the file menu as shown below in the diagram.

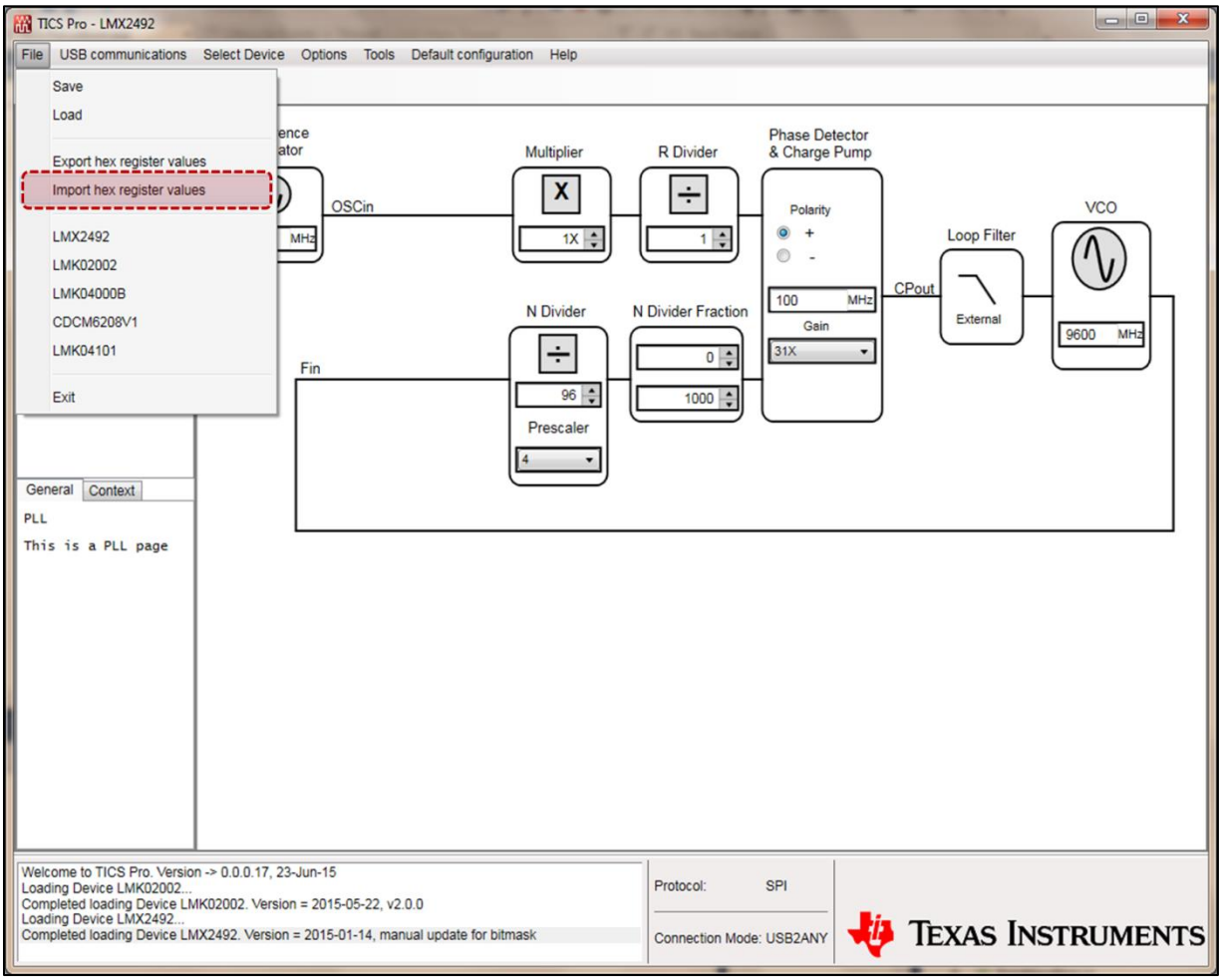
Figure 3 : File Menu - Recently Loaded Device Info



2.1.4. Importing Register Map Data

The "Import Hex Register Values" feature allows you to import the register information or data into the GUI in the following three formats

Figure 4 : File Menu - Import Hex Register Values



2.1.4.1. Format 1

CASE A (decimal reg#)
R0 0x80000100
R0 0x08030400
R1 0x00000101
...
R8 0x10000908
R9 0xa0022a09
R10 0x0082000b
R13 0x0286c00d

2.1.4.2. Format 2

CASE B (hex reg#)
R0x0 0x80000100
R0x0 0x08030400
R0x1 0x00000101
....
R0x8 0x10000908
R0x9 0xa0022a09
R0xb 0x0082000b
R0xd 0x0286c00d

2.4.1.3. Format 3

CASE C (no reg num)
0x80000100
0x08030400
0x00000101
...
0x10000908
0xa0022a09
0x0082000b
0x0286c00d

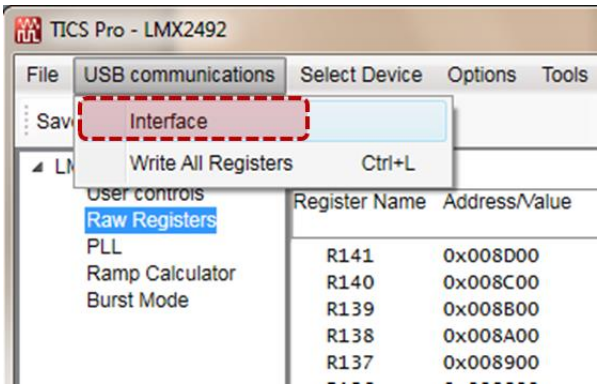
However when you export the register data using the "Export Hex Register Values" option, It is always exported in Format 1

2.2. USB Communications Menu

2.2.1. Interface

The Interface option allows you to select the protocol that is to be used for communication with the devices.

Figure 5 : USB Communications Menu - Interface



When you traverse through the USB Communications menu and click on the Interface option, the pop-up dialog shown below appears on the screen. The Communication setup window lists the Interfaces available for the device loaded. Select the required protocol and click on OK. The rectangular area at the middle of the window display image of the Pin Configuration of the respective interface selected.

Figure 6 : USB Communications - Interface Selection Dialog – In Simulation mode

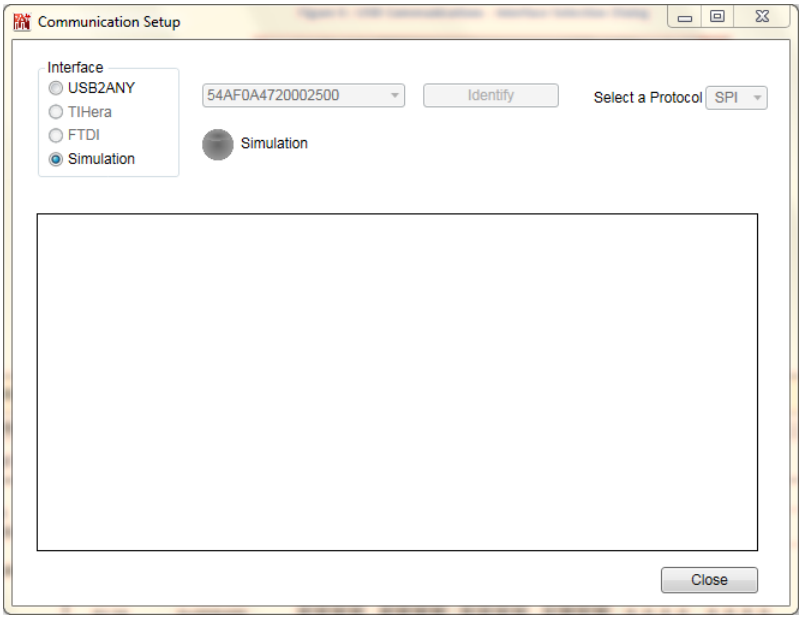


Figure 7 : USB Communications - Interface Selection Dialog – USB2ANY Device Disconnected

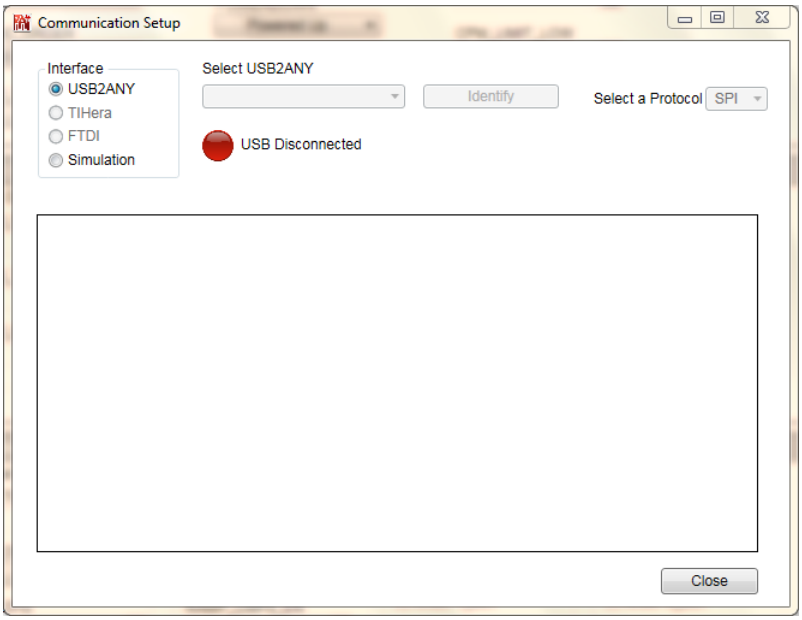
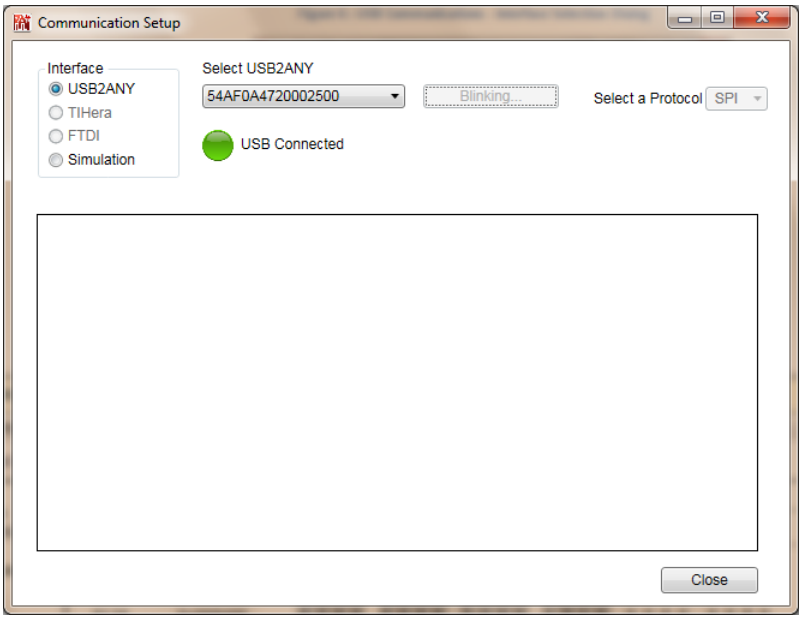


Figure 8 : USB Communications - Interface Selection Dialog – USB2ANY Device Connected



- **Connection Status**

  The connection status of a device can be identified from the LED where Green denotes one or more devices are connected. On the other hand Red LED denotes that the device is disconnected.

- **List of Devices**

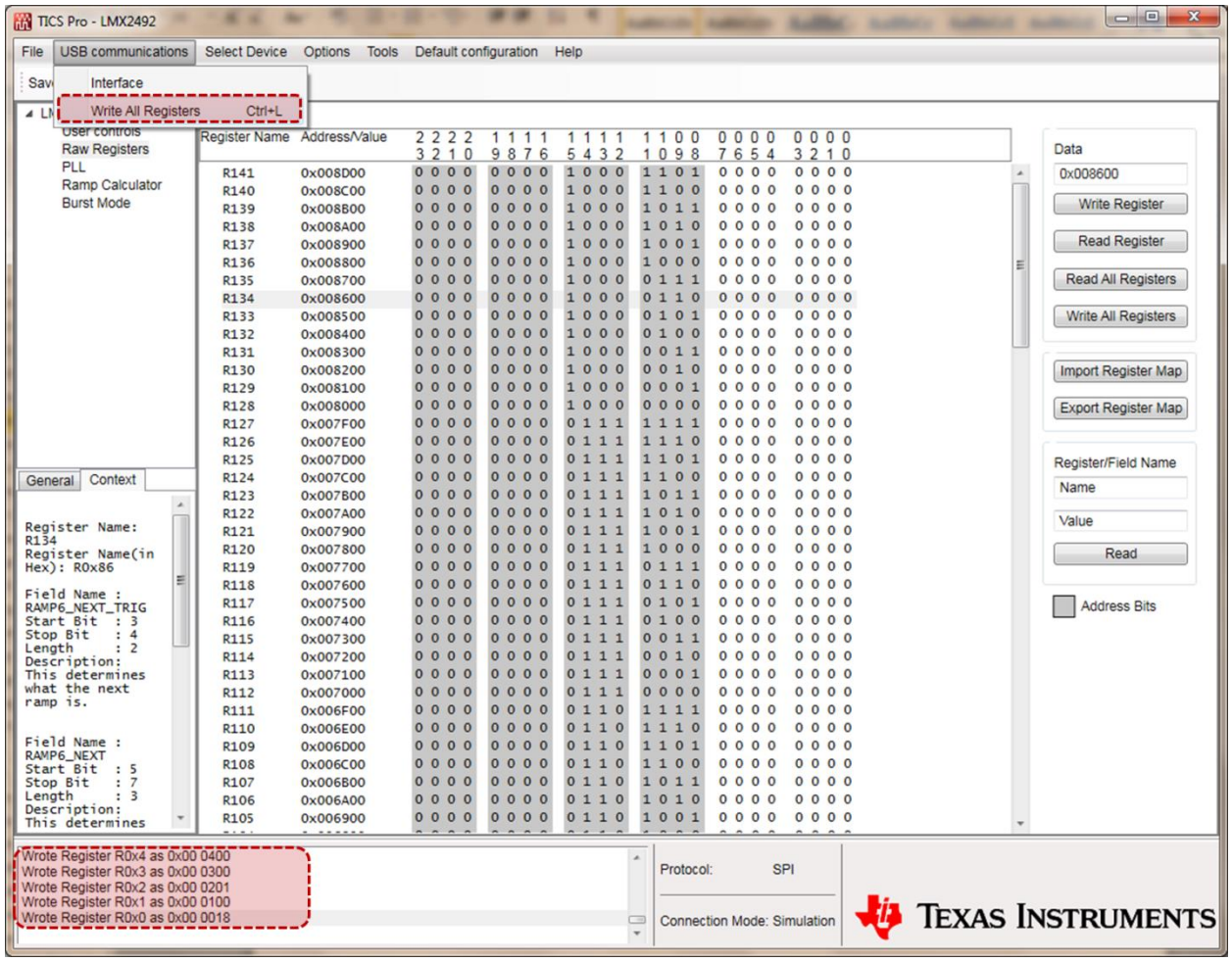
  The list of connected devices are populated in the listbox for each Interface respectively. Each device is represented as a unique string populated in the list box.

- **Identifying a device**
  The device connected could be identified by using the "Identify" option. Select one of the listed device and Click the identify option. The "Identify" option will change to "Blinking" state, that will toggle the LED of the selected device in the list.

2.2.2. Program All Register

The "Program All Register" option allows you to write all the register values into the device. The status bar at the bottom indicates this. It is similar to the "Write All" operation on the Raw Register Page

Figure 9 : USB Communications - Program All Register

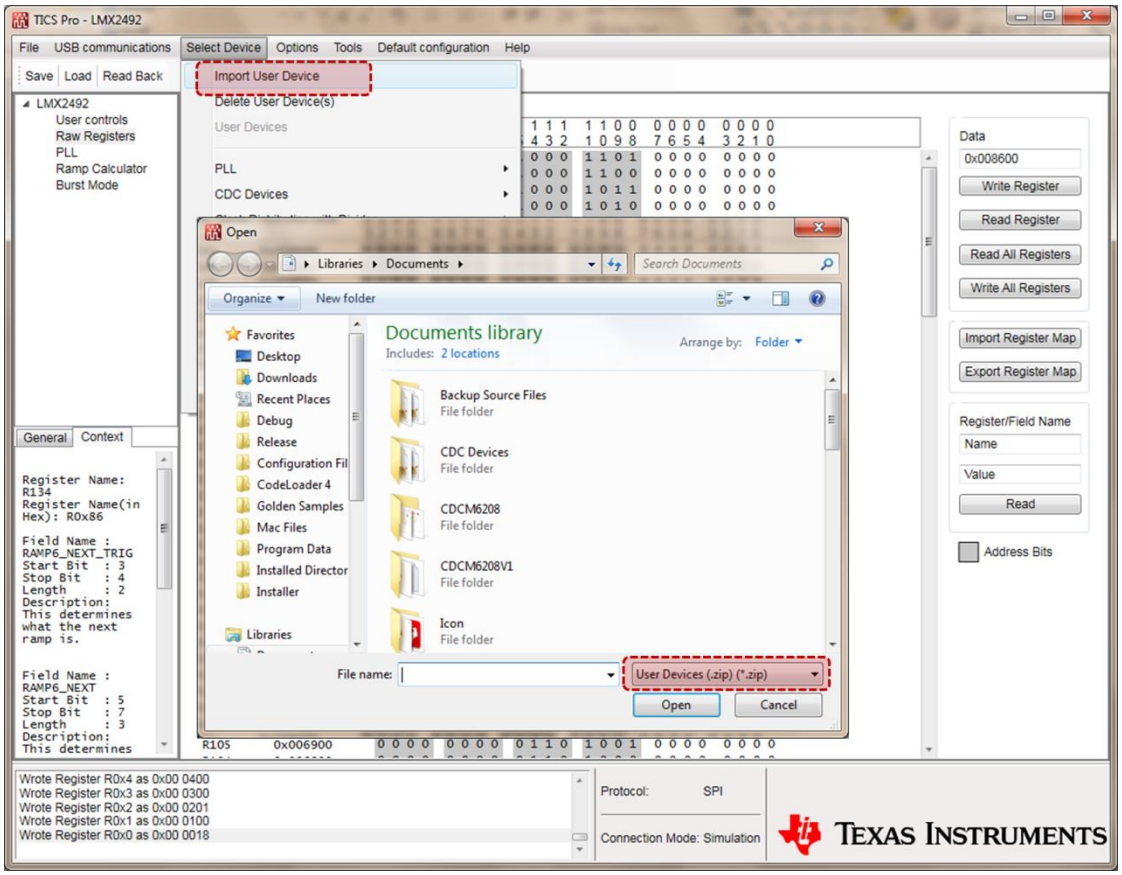


2.3. Select Device Menu

2.3.1. Import User Device

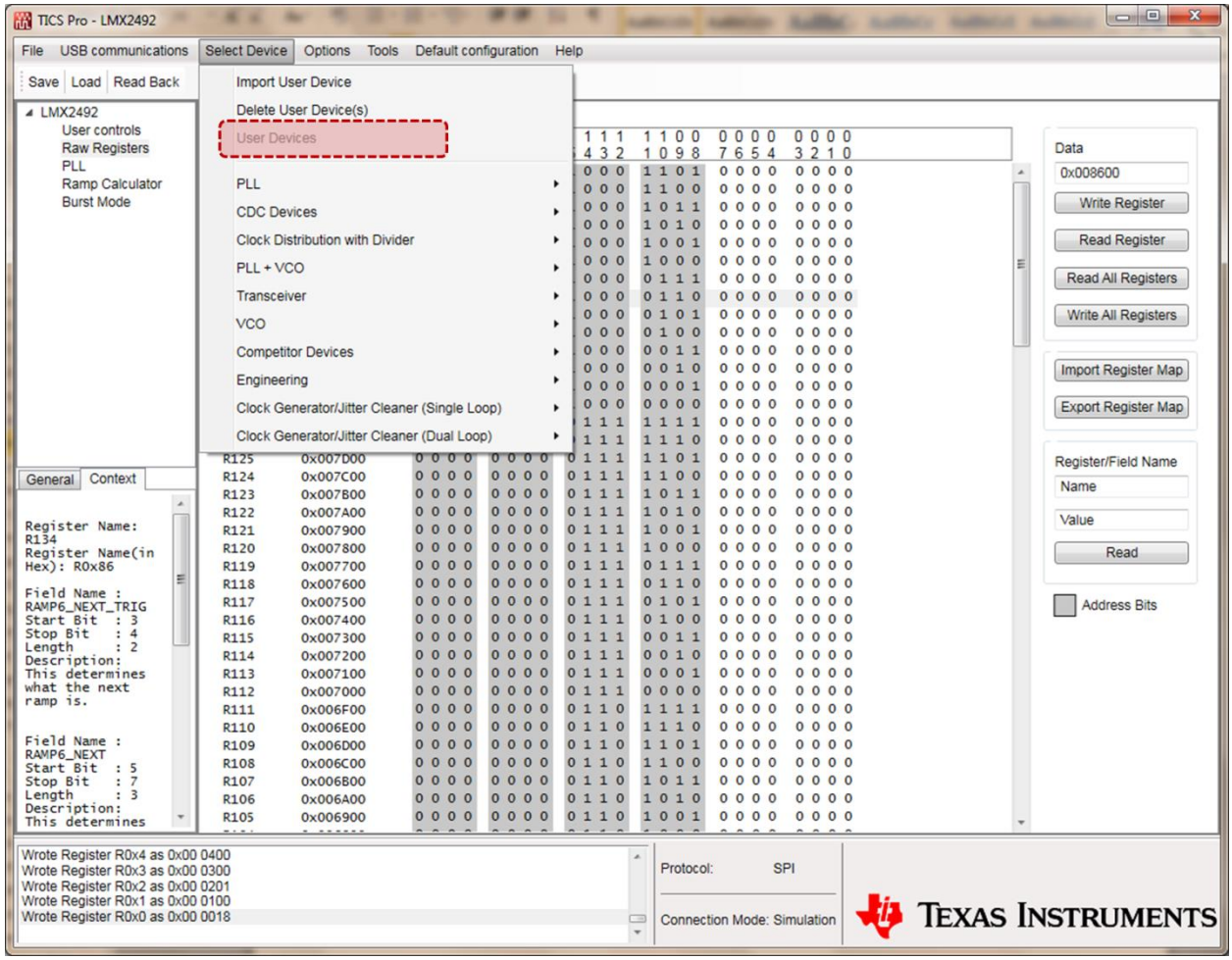
This option allows you to load new devices into the GUI. The User Device related files must be grouped together in a ".Zip" File. When the Import User Device option is selected, a dialog pops up to select the zip file. The GUI then unzips all the files and places them into the appropriate folders.

Figure 10 : Select Device - Import User Device



The new user devices would then appear in the Select Device menu as shown in figure below.

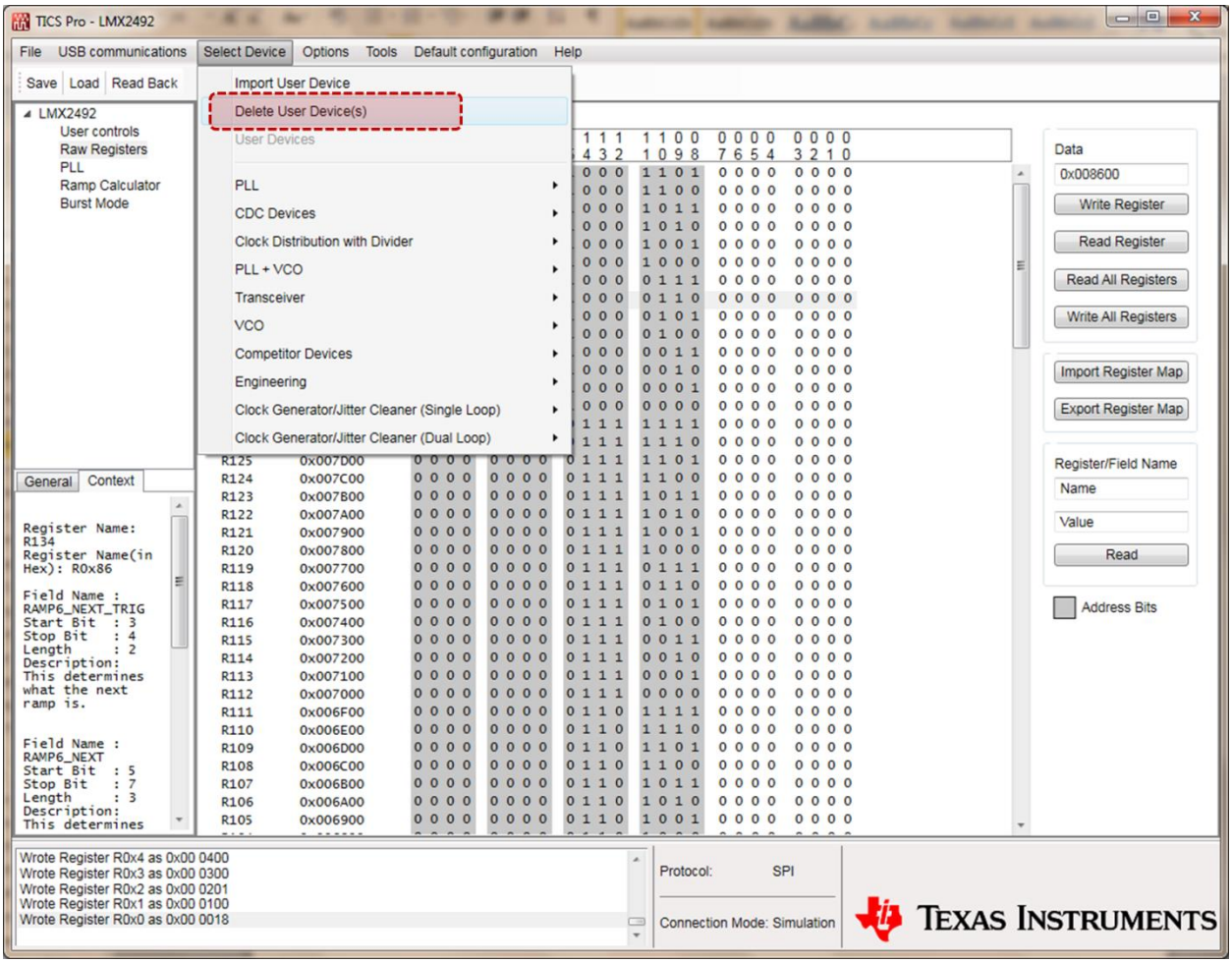
Figure 11 : Select Device - List of User Devices



2.3.2. Delete User Device

When you select the Delete User Device option, a pop-up displays on the screen. Select the required devices and press the Delete button

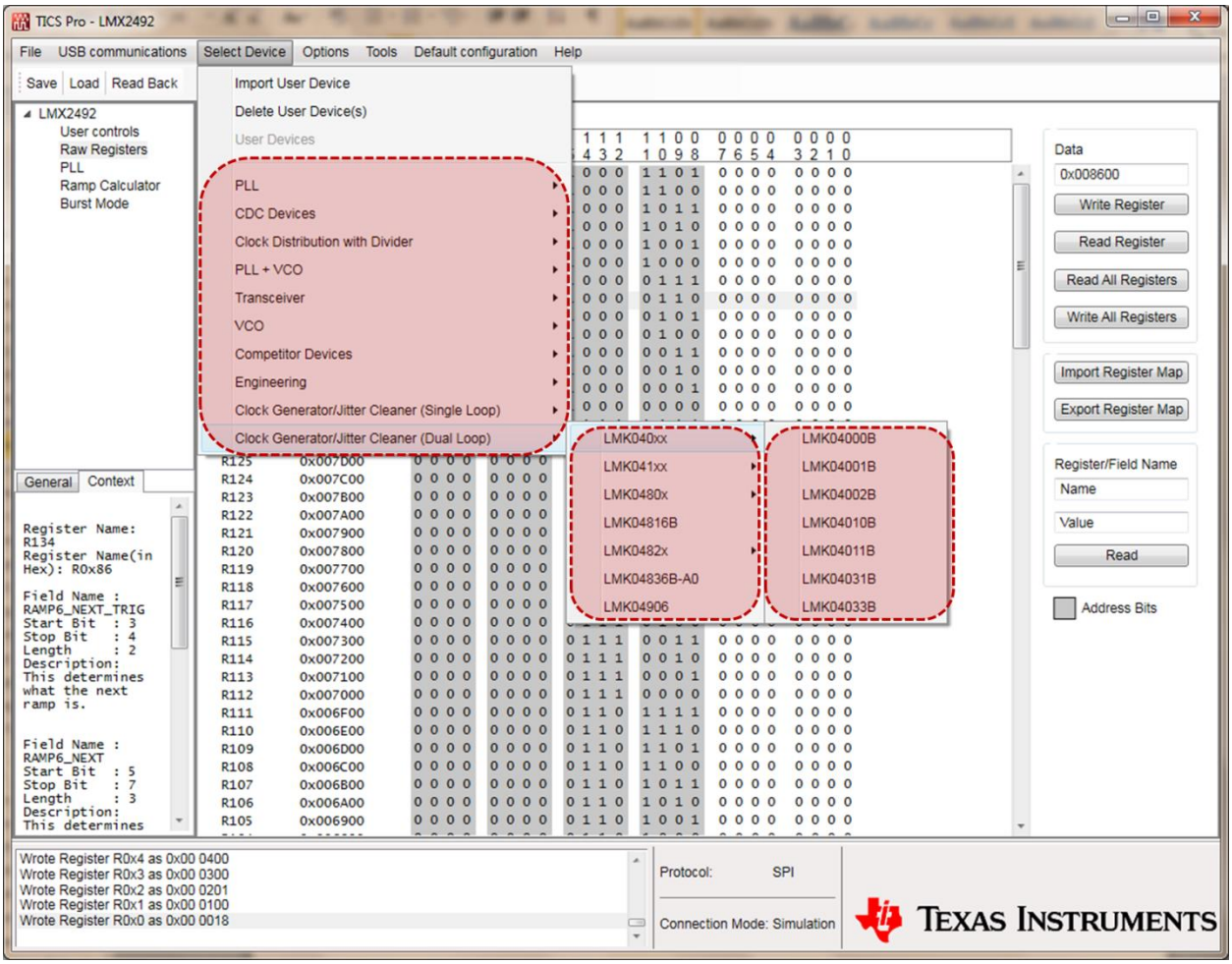
Figure 12 : Select Device - Delete User Device



2.3.3. List of Devices under groups

All the devices in the GUI are grouped into relevant categories based on the type of the device. There can be multiple levels of sub-menus based on the device types.

Figure 13 : Select Device - List of Devices under Groups

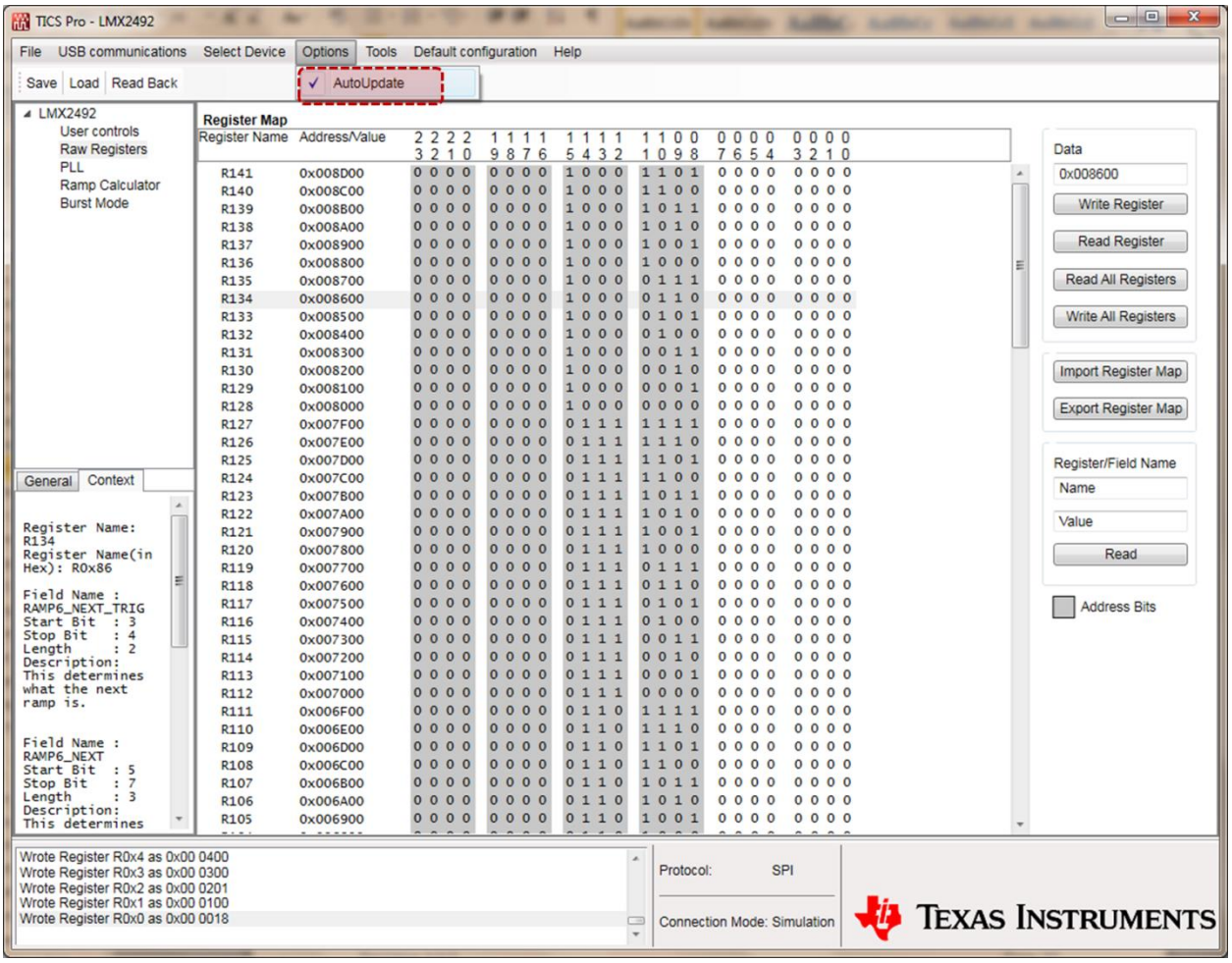


2.4. Options Menu

2.4.1. AutoUpdate

Whenever a change is made either to the register values or to any controls and the AutoUpdate option is true, the GUI automatically writes the value into the device. If the AutoUpdate option is not selected, any changes made to the UI – registers / controls, are not written to the device unless the Write Register / Write All option is explicitly selected

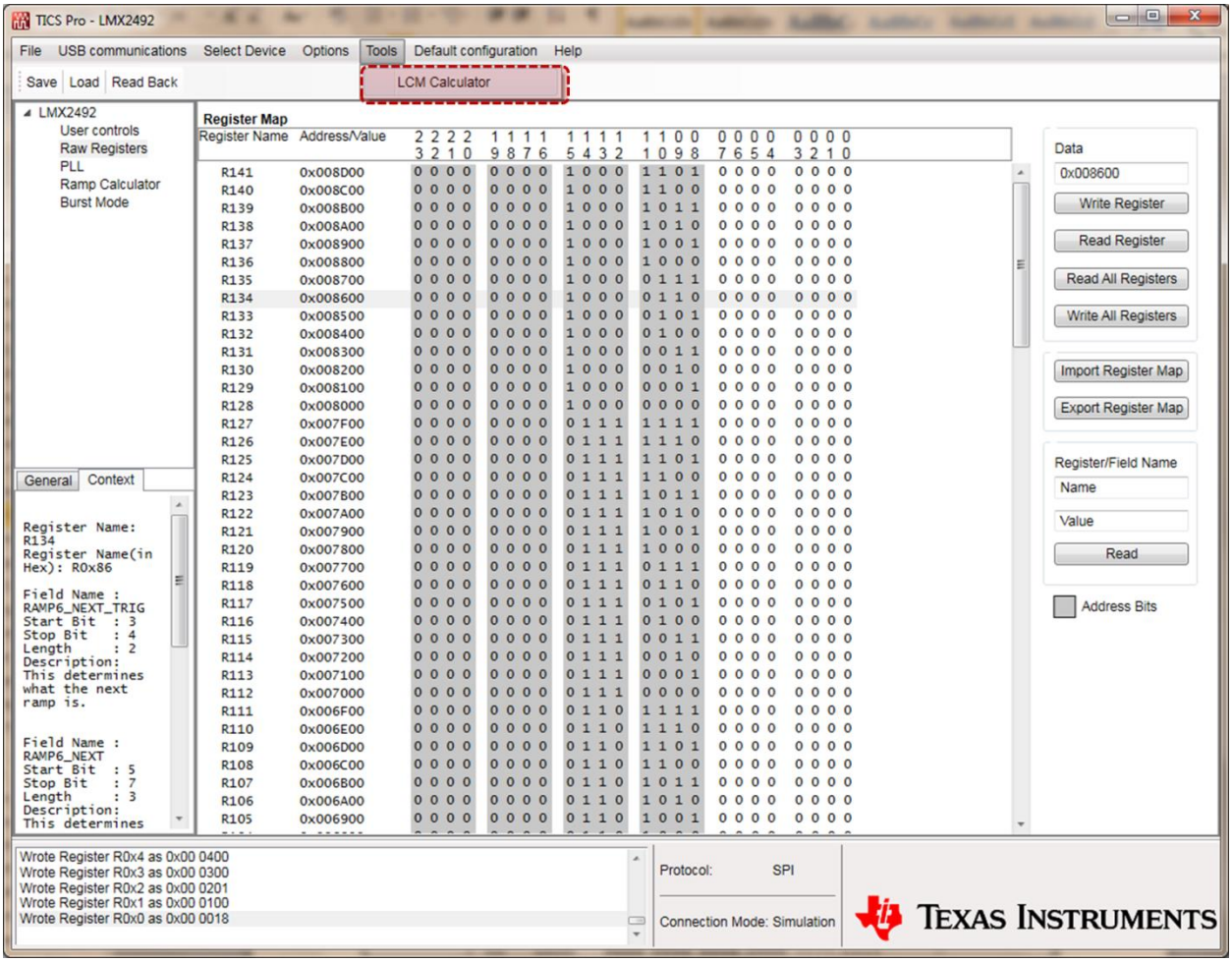
Figure 14 : Options - AutoUpdate



2.5. Tools Menu

This menu lists all the tools available for use in the TICS Pro GUI

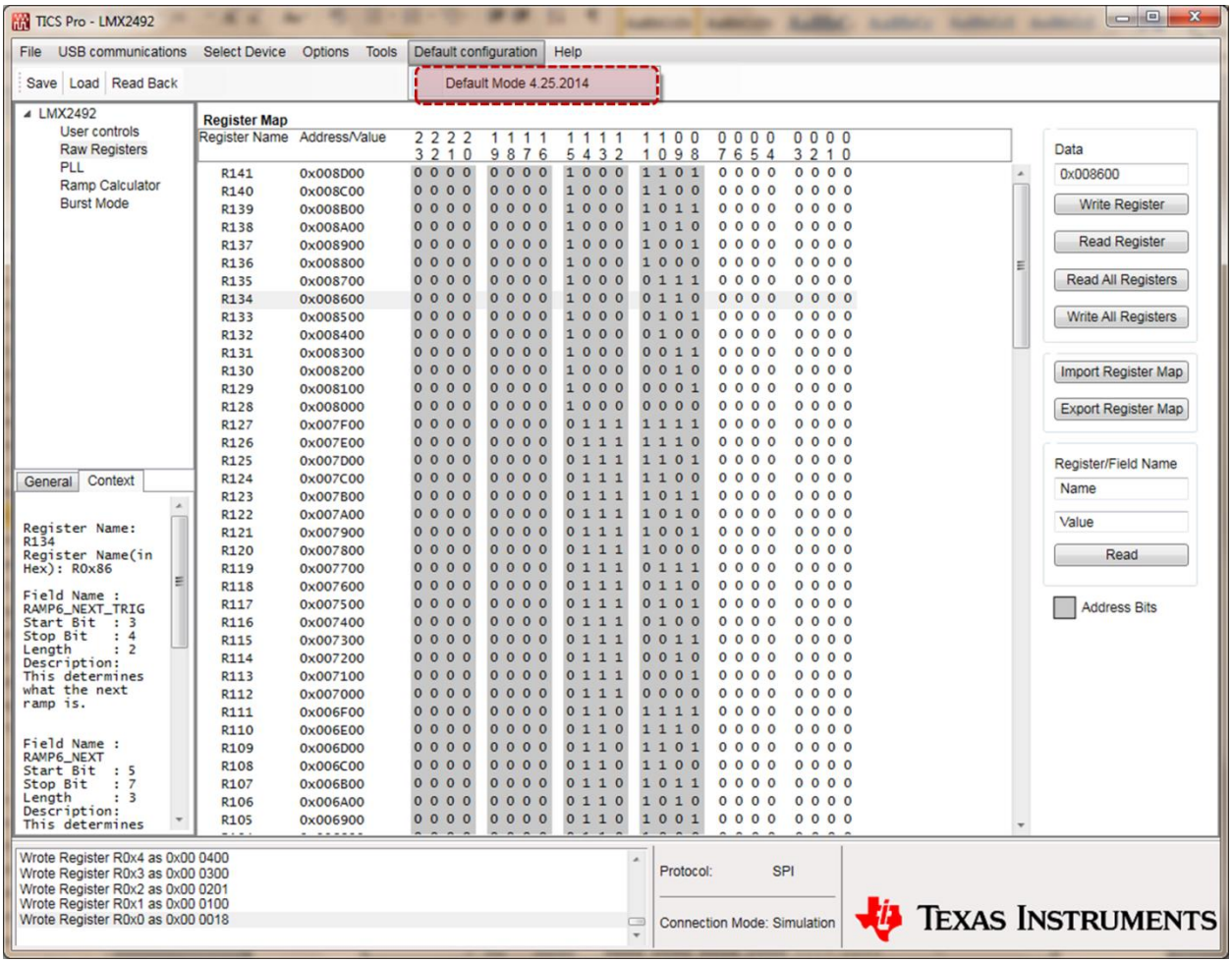
Figure 15 : Tools Menu



2.6. Default Configuration Menu

The GUI maintains a list of default modes. When you click on the Default Configuration menu all the available default modes are listed. You can select one of the modes and the device is automatically re-configured.

Figure 16 : Default Configuration Menu

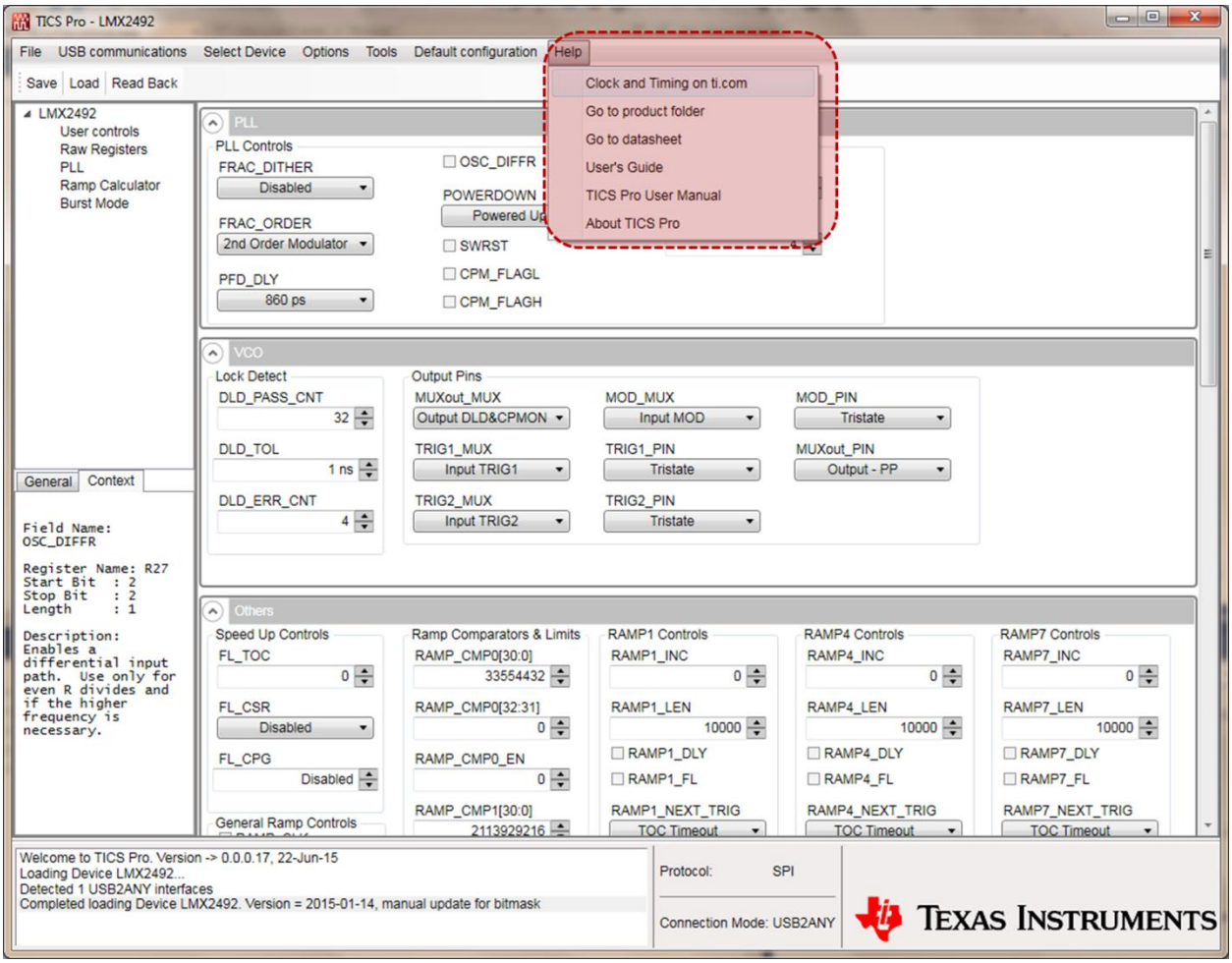


2.7. Help Menu

The Help Menu provides the following information

- Clock and timing on ti.com – Connects to the relevant page on the TI website that displays the clock and timing information
- Go to product folder – Opens the Installed directory of the software
- Go to datasheet – Opens the datasheet of the currently loaded device
- User's Guide – Opens the website link containing the User's Guide
- TICS Pro User Manual – Opens the User Manual
- About TICS Pro – Displays the about screen for the TICS Pro software

Figure 17 : Help Menu



3. TICS PRO Page Selection

3.1. TICS PRO Pages

The different pages available in the TICS PRO GUI are as listed below

- User Control Page
- Raw Register Page
- PLL Page
- Flex Page
- Burst mode Page

3.2. User Control Page

The User control Page displays the different controls to work with the device grouped by separate Master Groups

3.2.1. Master Group Display for a Device

The controls are grouped into separate Master Groups as shown in the figures below. User can expand one or multiple master groups to access the controls from the group.

Figure 18 : User Control Page

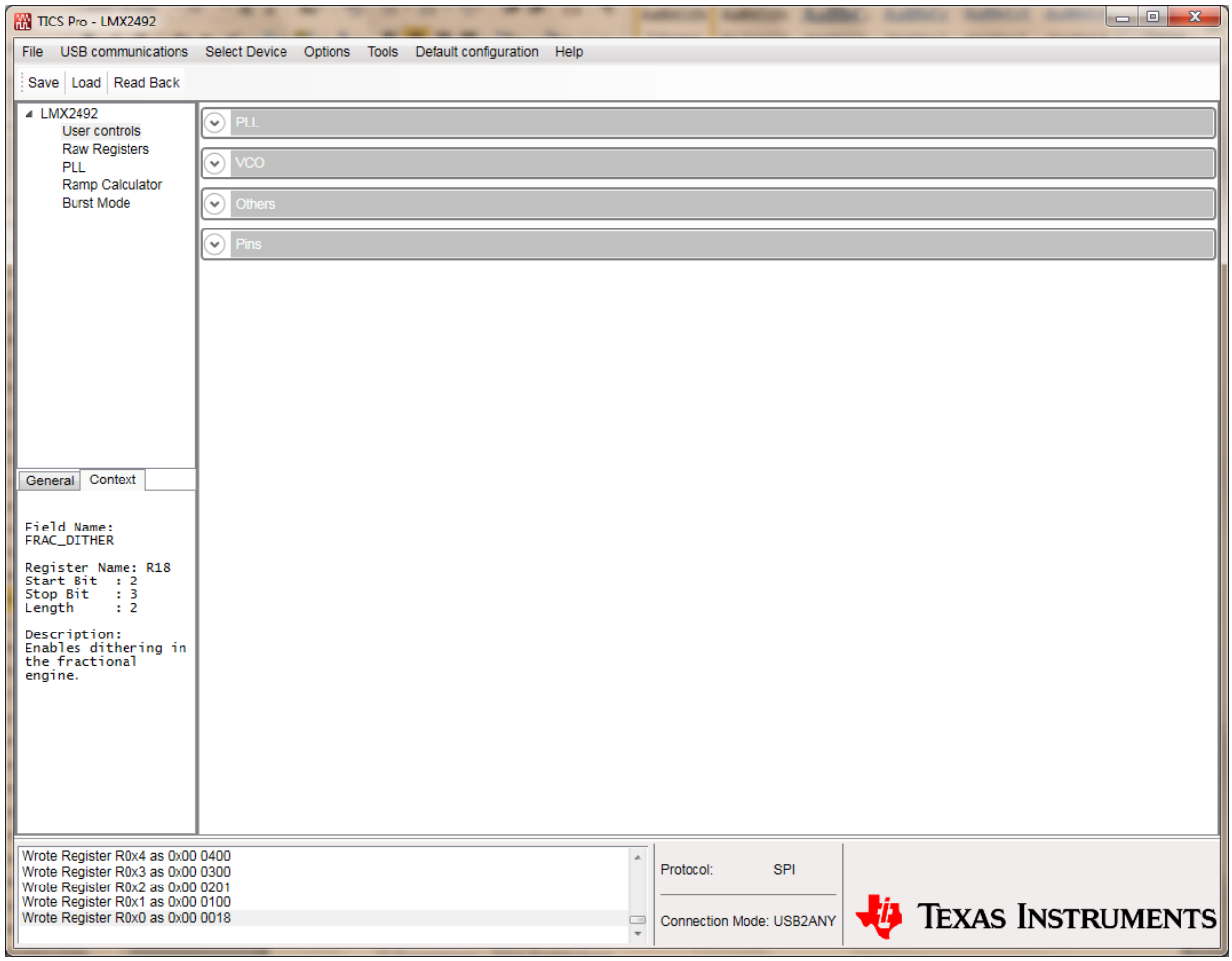
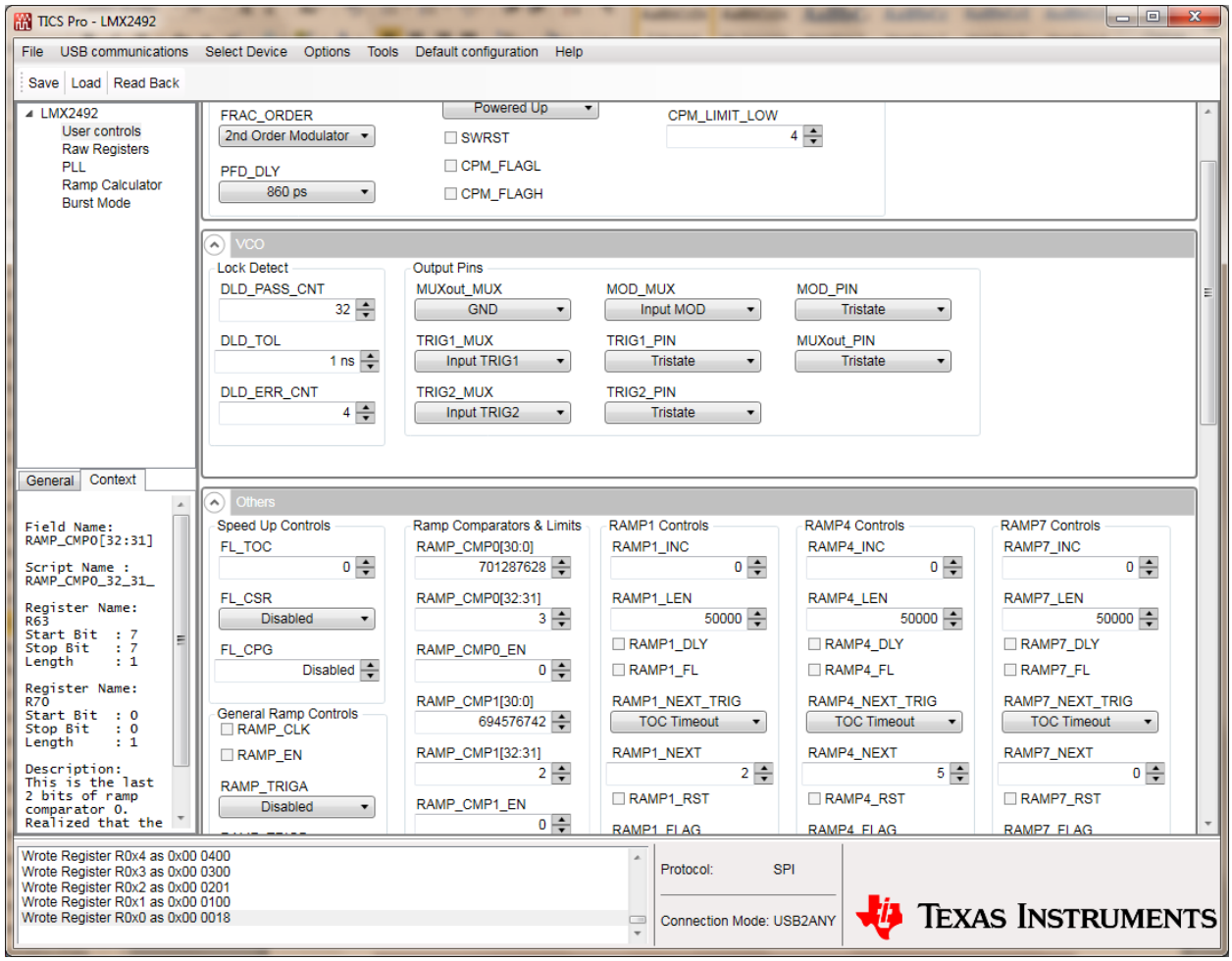


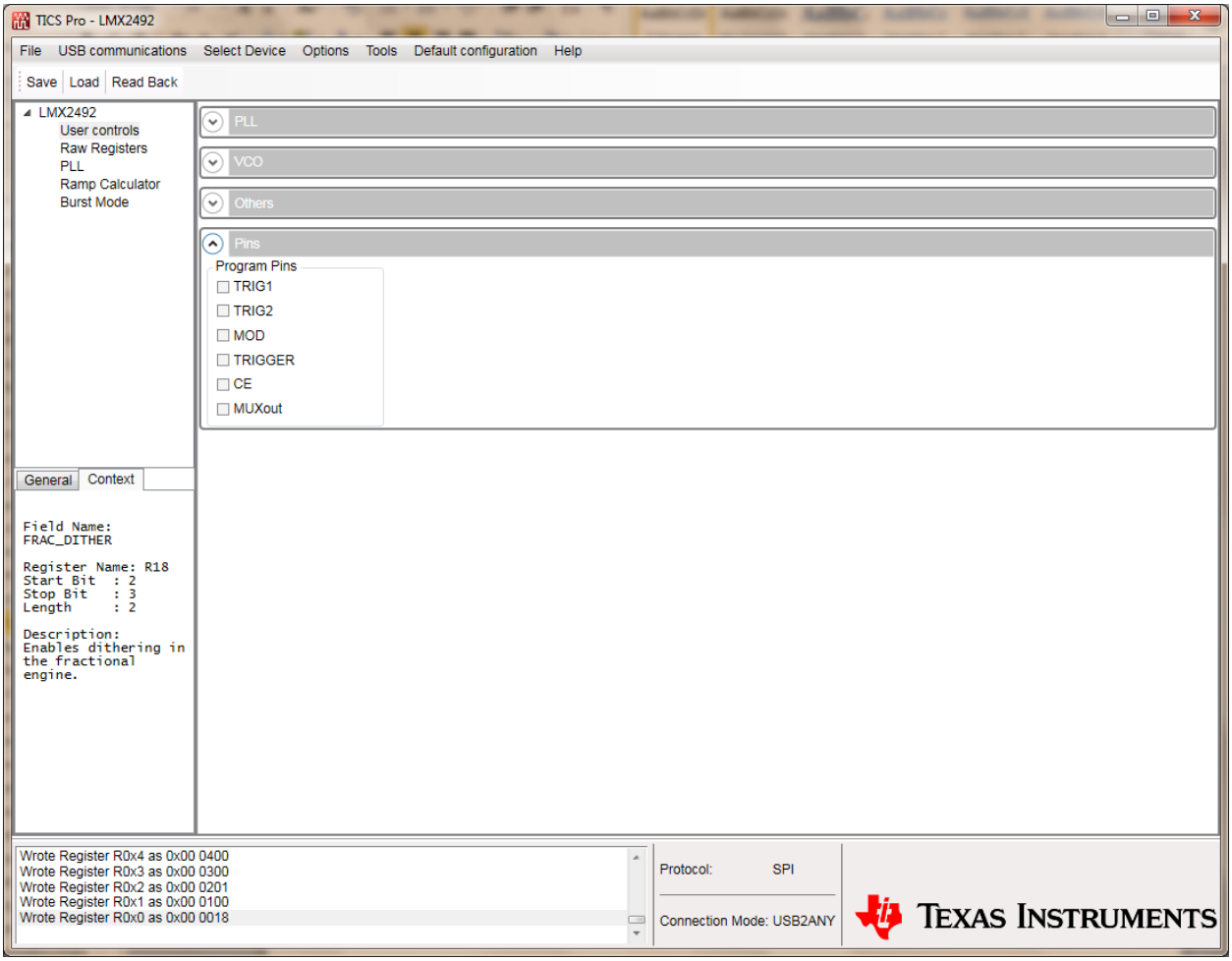
Figure 19 : User Control Page - Master Group



3.2.2. PINS Master group

The PINS Master Group is a common group that appears for all devices irrespective of the device type selected.

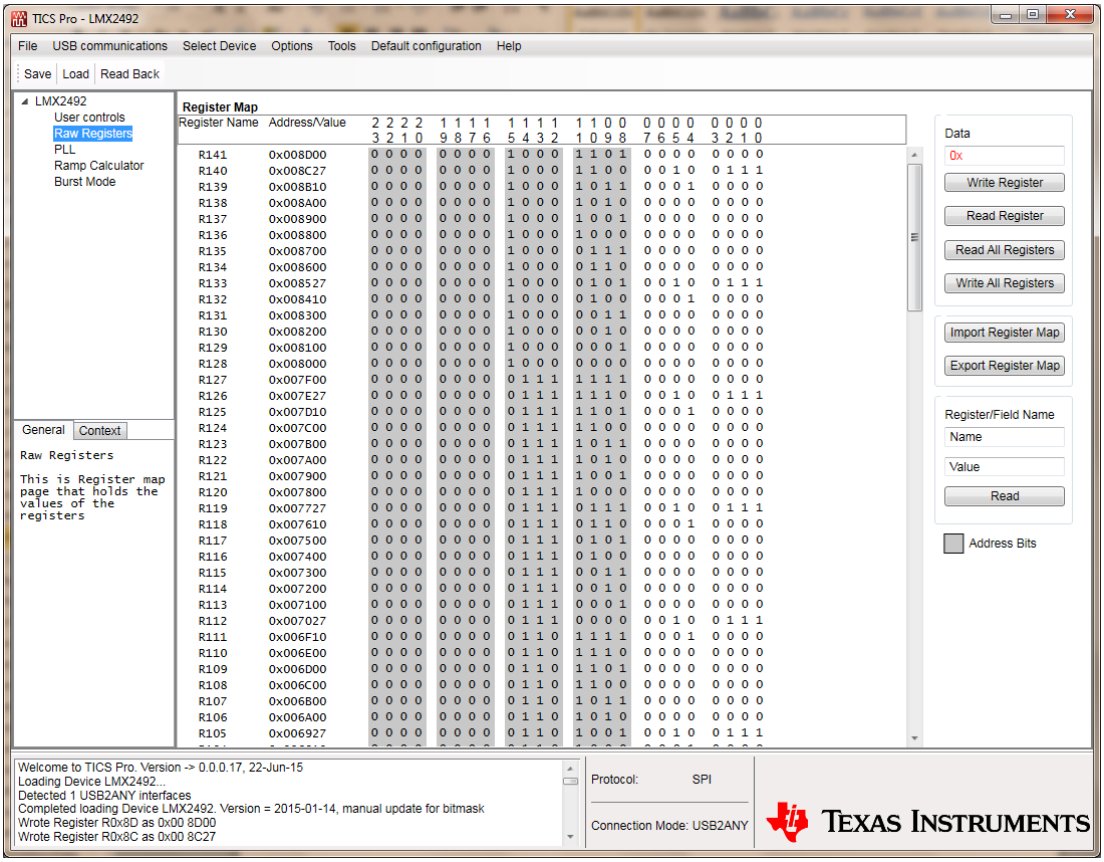
Figure 20 : User Control Page - PINS Master Group



3.3. Raw Register Page

The Raw Register Page displays the register map for the selected device.

Figure 21 : Raw Register Page – Register Map



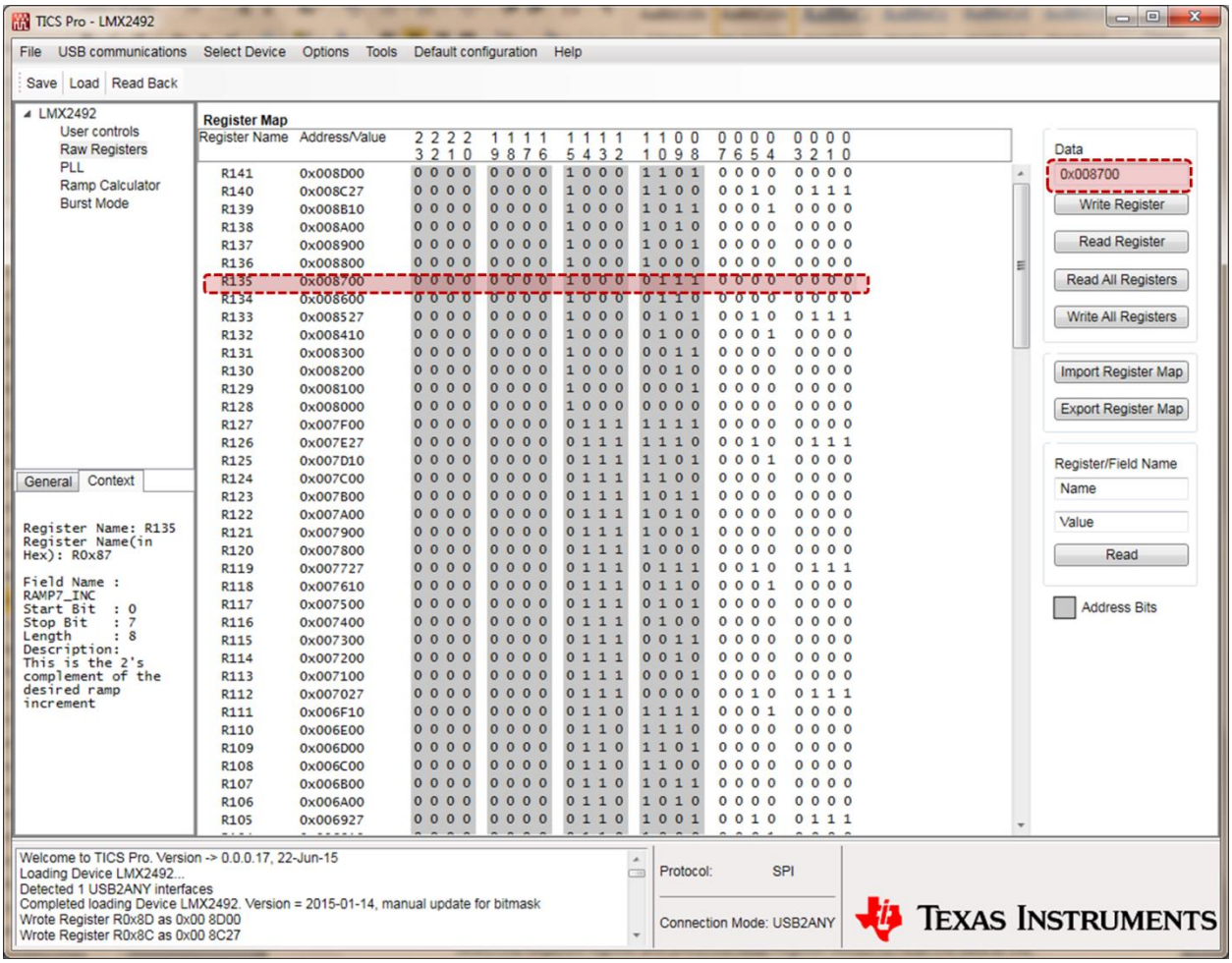
3.3.1. Operations

The following operations can be performed on the Raw Register Page.

a) Write Register

Select the required register, enter the data to be written and press the Write Register button to write the data into the register

Figure 22 : Raw Register Page - Operations



b) Read Register

Select the required register and press the Read Register button to read the data at the selected register in the device.

c) Read All

Pressing the Read All button, reads the data from all the registers in the device and updates the UI

d) Write All

Pressing the Write All button, writes all the data into the registers in the device

e) Import Register Map

This option allows you to load a register map into the GUI

f) Export Register Map

This option allows you to export or save the current register map into a file

g) Read

The Read option at the bottom of the page allows you to read the value of a particular register or a field in the device. This option does not cause any updates to the UI. The read data is only displayed in the box shown

Figure 23 : Raw Register Page – Read Register Value

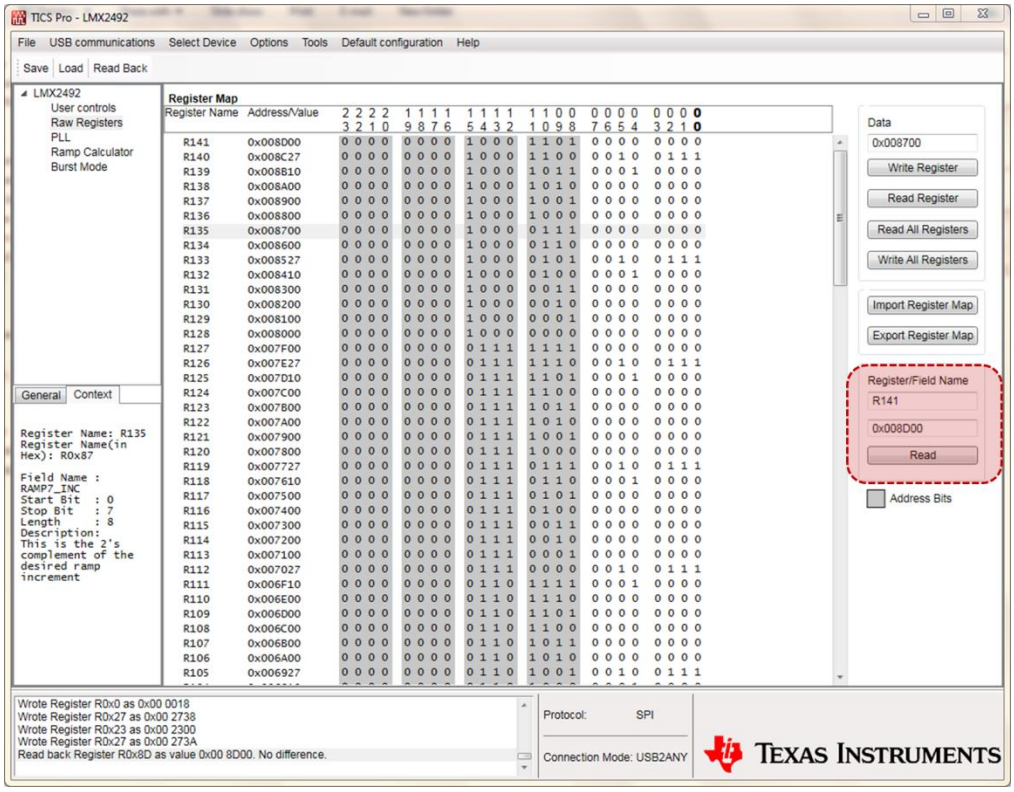
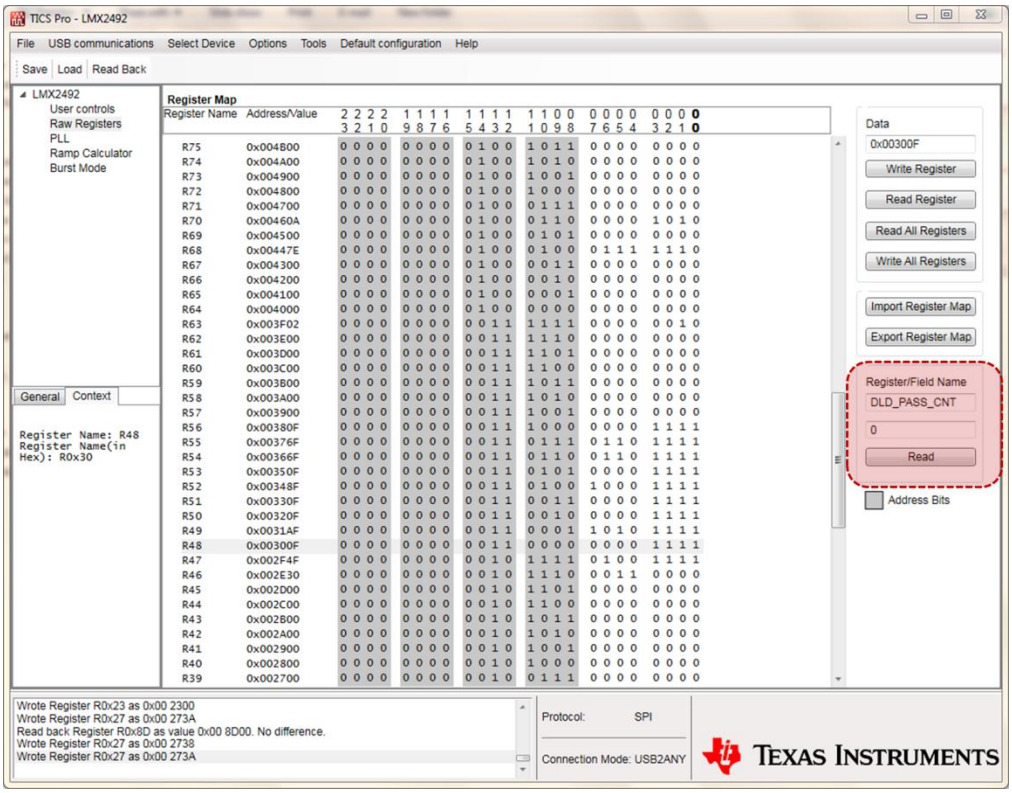


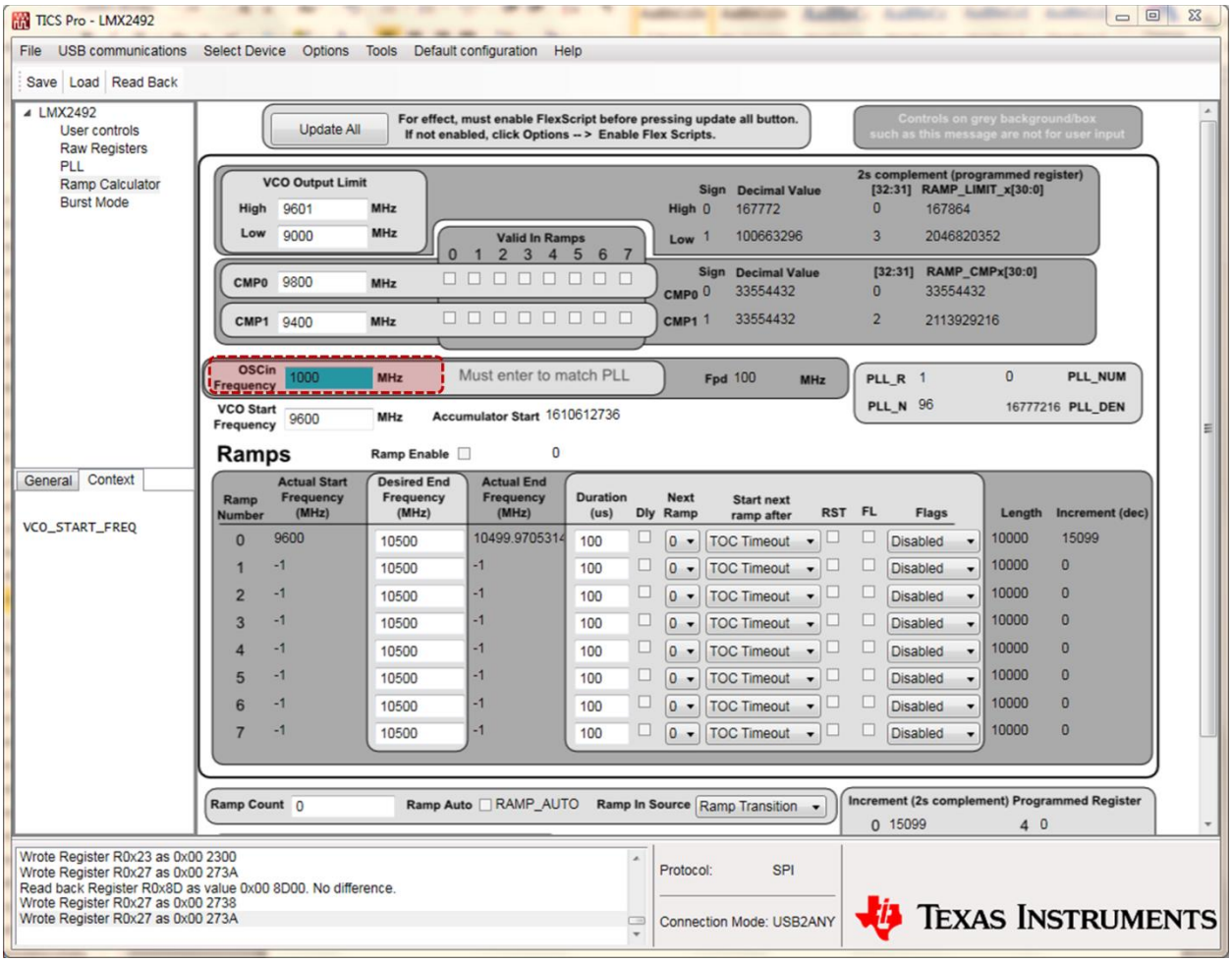
Figure 24 : Raw Register Page – Read Field Value



3.3.2. Using the AutoUpdate feature

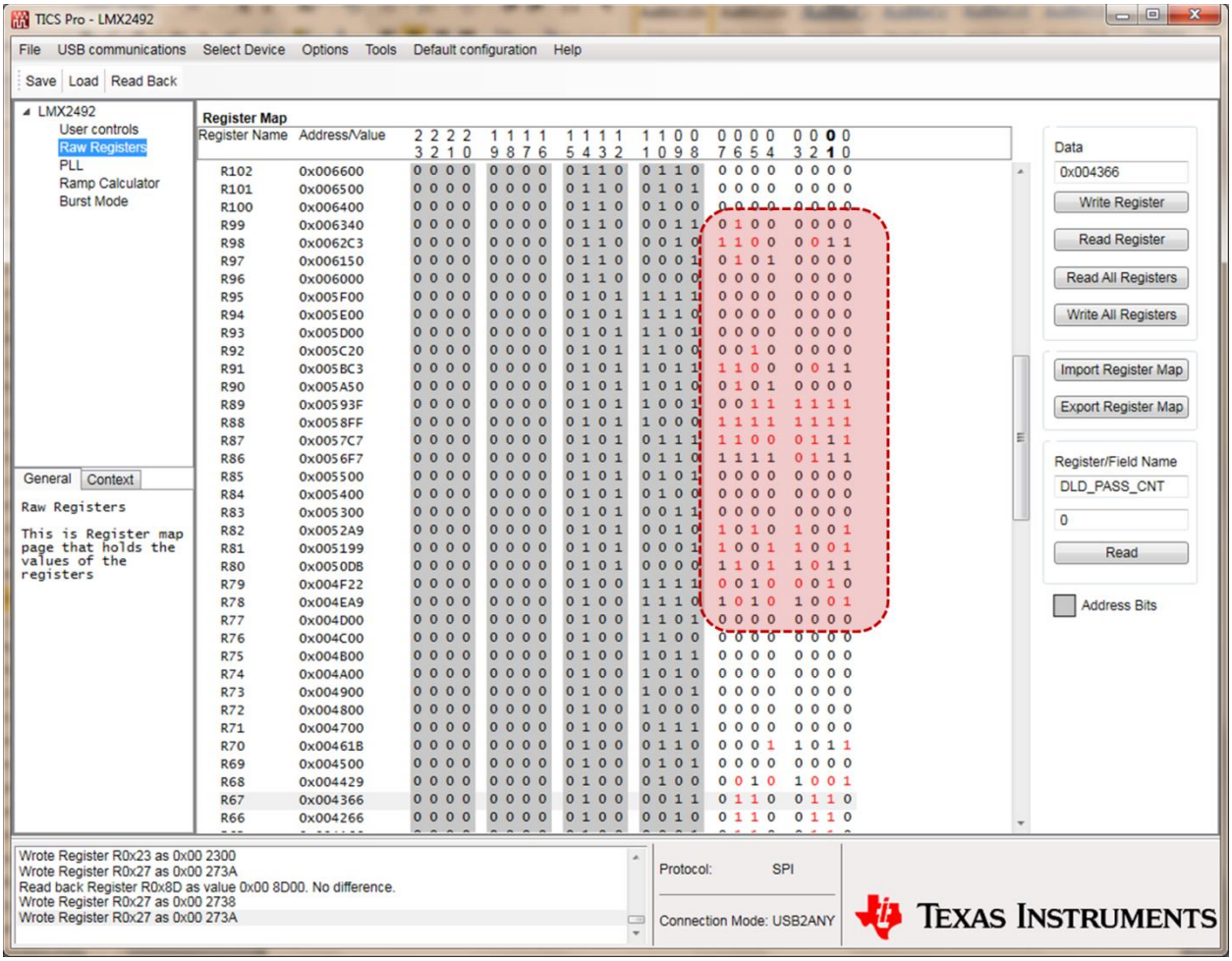
When the AutoUpdate Option is not selected and changes are made to any control in the GUI, the Raw Register Map displays the new data in red color indicating that these changes have not been written to the device.

Figure 25 : AutoUpdate OFF - Changes made to Control



For example, in the above page we have made changes to the "OSCin Frequency" as shown in the above figures and the AutoUpdate option is OFF. In the Raw Register Page, we will notice that for a couple of registers the data is updated in red color indicating that this has not been written to the device. An explicit write operation must be performed to write the data into the device.

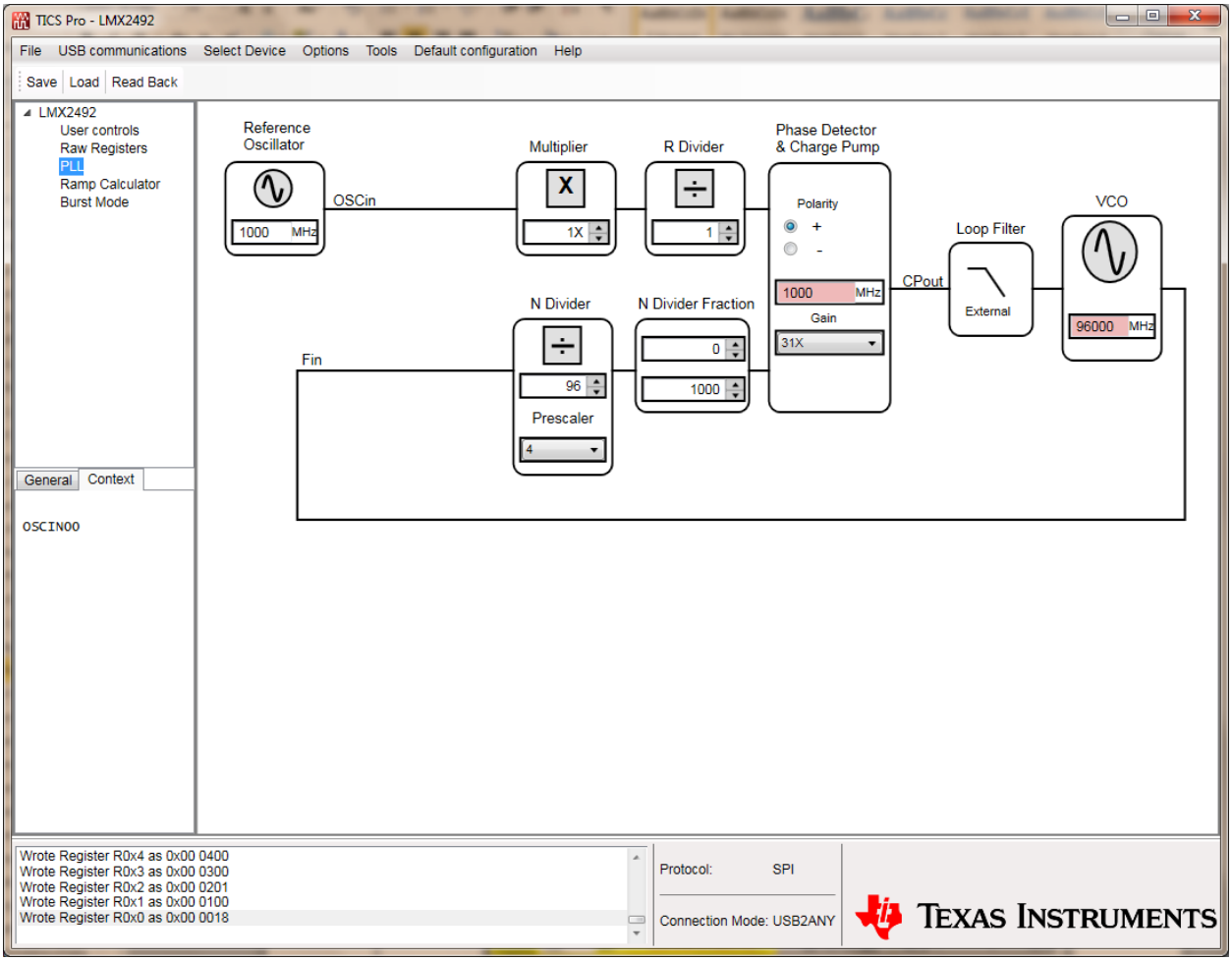
Figure 26 : AutoUpdate OFF - Change in Data in Raw Register Page



3.4. PLL Page

For a particular device, there can be one or more PLL pages and based on the device selected the required controls will be enabled on each of the PLL page.

Figure 27 : PLL Page

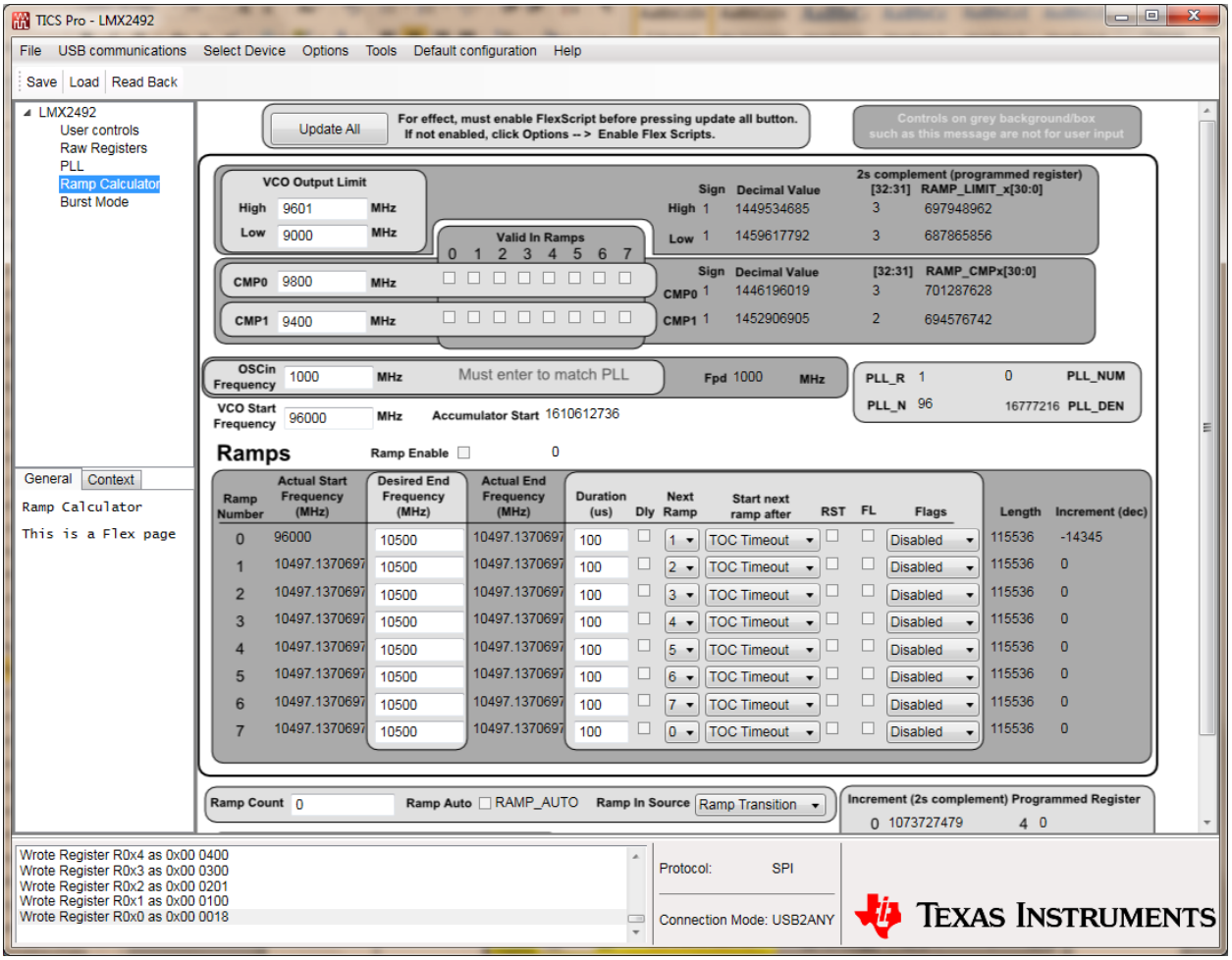


3.5. Flex Page

The Flex Pages in the TICS Pro GUI, provide the option to create dynamic pages and have complete control over the design and functionality of the page. A sample flex page is shown below.

The procedure to create a flex page is available as a separate document

Figure 28 : Flex Page



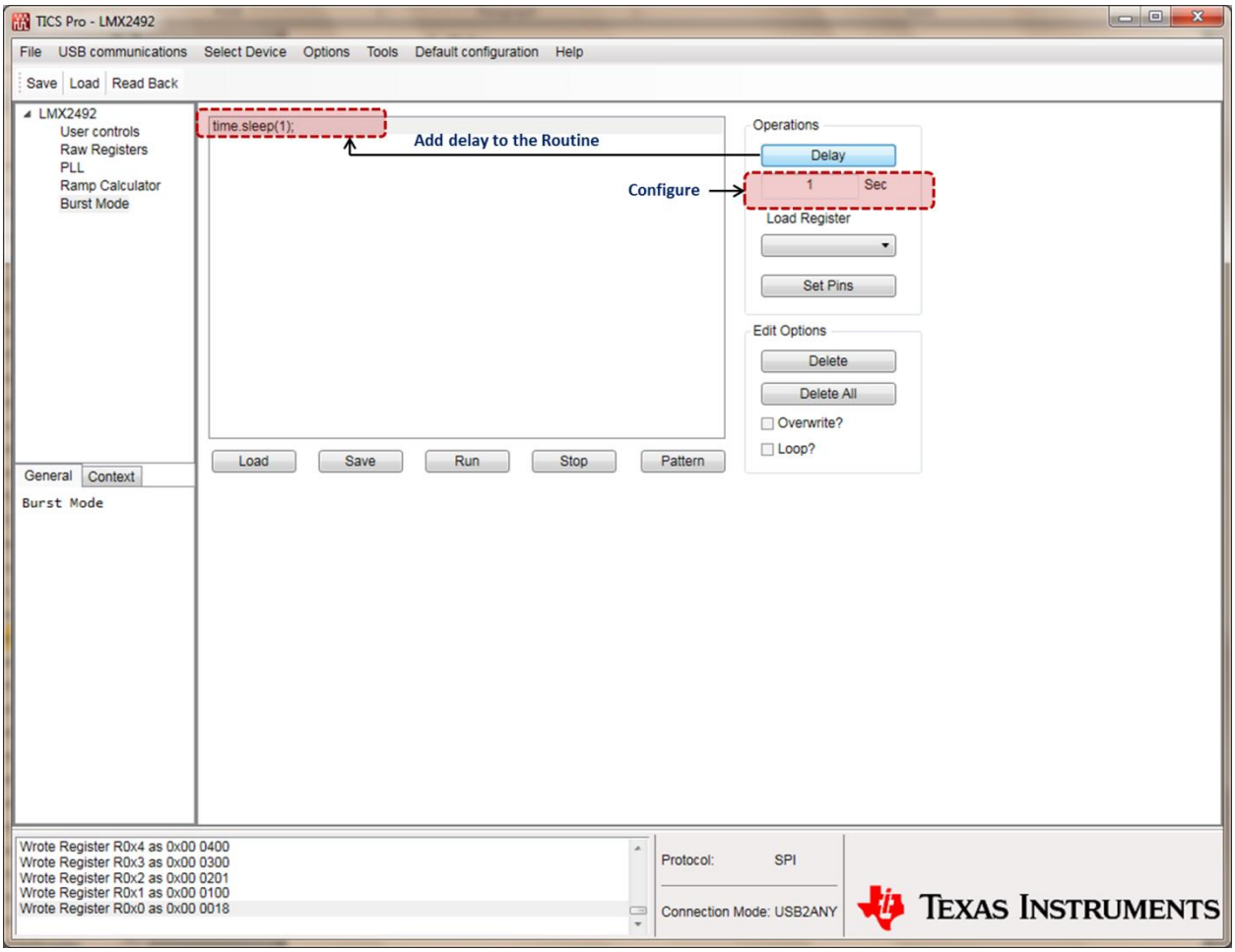
3.6. Burst Page

The Burst Page allows user to write simple routines and execute them or save them for later use

a) Delay

You can configure the delay in sec(s) using the text control as shown below. Pressing the Delay button will add a line corresponding to Delay in your routine

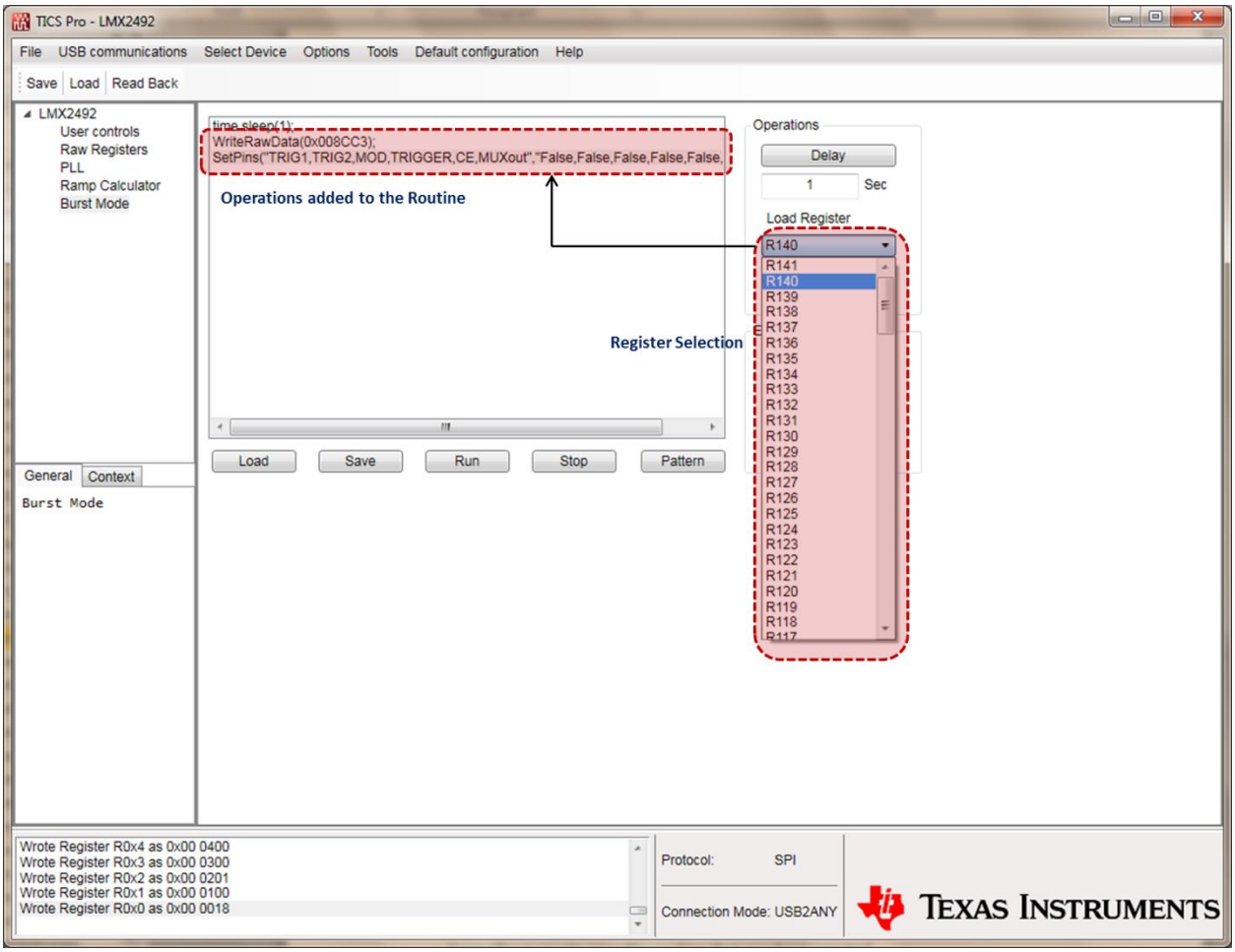
Figure 29 : Burst Page - Delay



b) Load Register

This option allows you to select the required register and add the lines corresponding to the "Write Register" and the "Set Pins" operation to the routine

Figure 30 : Burst Page - Load Register



c) Load

The Load options allows you to load a pre-saved routine and execute it

d) Save

Save allows you to save the routine that you have created

e) Run

The Run option executes the routine that you have created / loaded in the pane.

f) Stop

This option stops the execution of the routine which is currently executing

g) Overwrite ?

Selecting this option allows you to overwrite the steps in the routine

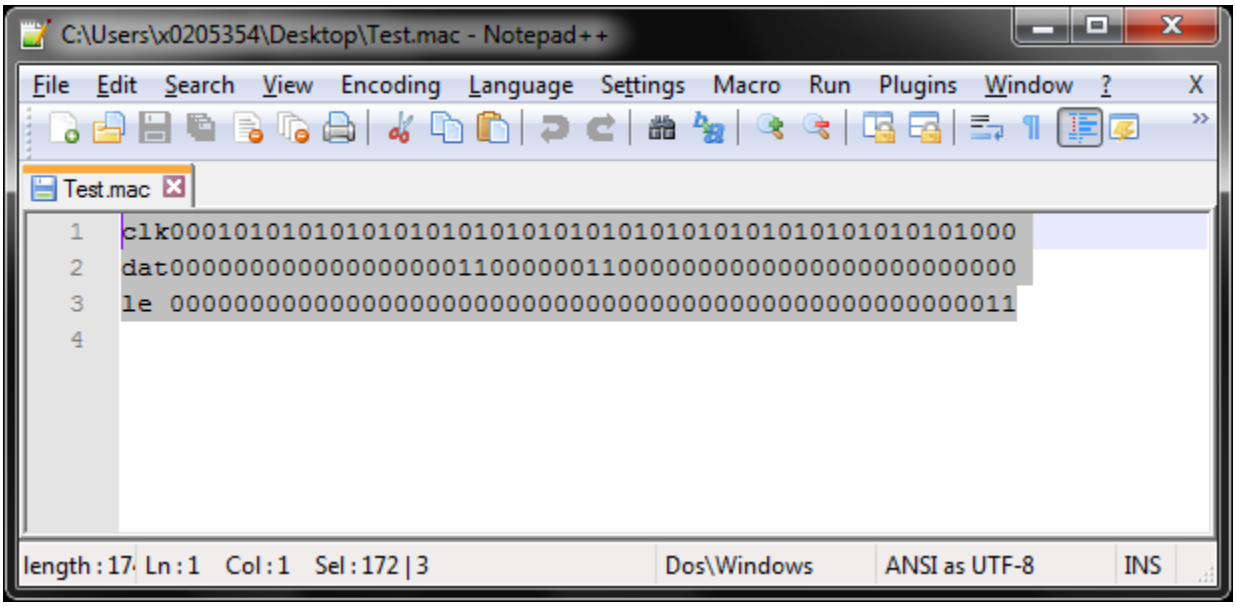
h) Loop ?

When this option is set to true, the routine is continuously executed until the stop button is pressed

i) Pattern

When you click on the pattern button, the data in the routine is saved into the individual line status file format for the active communication protocol as shown below

Figure 31 : Burst Mode - Pattern File



4. Automation Function

The automation functions help provide a means to access / modify the behavior of the control. Some of the automation functions that are available are listed below.

4.1. SetText

Boolean SetText(string controlName,string value)

- This function helps set the value, including setting a float or integer TextCtrl with a non-numeric String or setting a floating value for an integer TextCtrl.
- Note that a combo box is an 'integer' type for the index, but could have float values in the drop down which could match as a string.
- If function returns false it means no changes were made. True indicates that the operation was successful.

Explanation for Parameters

Parameter_Name	Description
controlName	– Name of the control as described in the INI file.
Value	– Use a text string to set the index of a control with items. (ComboBox, ListBox, SmartComboBox). – Or, set Text of a simple StaticText, TextCtrl, MultiTextCtrl, Button.

4.2. GetText

string GetText(string controlName)

- Returns the current text string of a control with the items.(ComboBox, ListBox, SmartComboBox)
- Or gets the Text of a simple StaticText, TextCtrl, MultiTextCtrl.

Explanation for Parameters

Parameter_Name	Description
controlName	– Name of the control as described in the INI file.

4.3. SetIndex

Boolean SetIndex(string controlName,int value)

- Sets the index directly of a control with items type control in Flex: ComboBox, ListBox, SmartComboBox, CheckBox, RadioButton

Explanation for Parameters

Parameter_Name	Description
controlName	– Name of the control as described in the INI file.
Value	– Index value of the item to be set in the control's list.

4.4. GetIndex

long GetIndex(string controlName)

- Gets the index of a control directly with items type control in Flex for the controls of type : ComboBox, ListBox, SmartComboBox, CheckBox, RadioButton.

Explanation for Parameters

Parameter_Name	Description
controlName	Name of the control as described in the INI file.

4.5. SetFieldValue

Boolean SetFieldValue(string controlName,long value)

- Sets the field value of the specified field in the register map for ComboBox, ListBox, SmartComboBox, CheckBox, TextCtrl, StaticText.Note, that a control which has linked bits will be set all together

Explanation for Parameters

Parameter_Name	Description
controlName	– Name of the control as described in the INI file.
Value	– The unsigned integral value of the field to be set.

4.6. GetFieldValue

ulong GetFieldValue (string controlName)

- Gets the field value of the specified field in the register map for Combo Box, ListBox, SmartComboBox, CheckBox, TextCtrl, StaticText. Note that a control which has linked bits will be set all together

Explanation for Parameters

Parameter_Name	Description
controlName	– Name of the control as described in the INI file.

4.7. PressButton

Boolean PressButton(string controlName)

- Triggers the press button event and returns "False" if not valid and "True" if valid.

Explanation for Parameters

Parameter_Name	Description
controlName	– Name of the control as described in the INI file.

4.8. PressSpinButton

Boolean PressSpinButton(string controlName,int direction)

- Triggers the press button event and returns "False" if not valid and "True" if valid.

Explanation for Parameters

Parameter_Name	Description
controlName	Name of the control as described in the INI file.
direction	Up if direction >0 by the value. Down if direction < 0 by the value.

4.9. WriteRawData

Boolean WriteRawData(long lValue)

- Parses the input value into address and data and writes the data of the register to corresponding address of the device

Explanation for Parameters

Parameter_Name	Description
lValue	The raw value of the register that is a combination of address and data. The length of this value is equal to the register length.

4.10. WriteAddressData

Boolean WriteAddressData(long lAddress,long lValue)

- Writes the input value to the device at the corresponding input address passed

Explanation for Parameters

Parameter_Name	Description
lAddress	The address of the register to be written to the device.
lValue	The data of the register to be written to the device.

4.11. WriteRegisterByIndex

Boolean WriteRegisterByIndex(int Index)

- Writes the current value of the register that is located at the index passed in the register map

Explanation for Parameters

Parameter_Name	Description
Index	The index of the register in the register map for which the value to be written to the device.

4.12. WriteRegister

Boolean WriteRegister(string registerName)

- Writes the value of the register having the input register Name to the device.

Explanation for Parameters

Parameter_Name	Description
registerName	The name of the register to be written to the device.

4.13. WriteAllRegisters

Boolean WriteAllRegisters()

- Writes the value of all registers to the device.

4.14. GetDevice

Boolean GetDevice(out string deviceName, out string deviceType)

- Retrieves the name and type of the current device loaded and assigns them to the variables deviceName and deviceType

Explanation for Parameters

Parameter_Name	Description
deviceName	The name of the device currently loaded.
deviceType	The type of the currently loaded that could be either of the following • `BuiltinDevice` • `UserDevice` • `Tools`

4.15. SelectDevice

Boolean SelectDevice(string deviceName)

- Loads the input built in device to the GUI.

Explanation for Parameters

Parameter_Name	Description
deviceName	The name of the built in device to be loaded.

4.16. SelectUserDevice

Boolean SelectUserDevice(string userDeviceName)

- Loads the input user device to the GUI.

Explanation for Parameters

Parameter_Name	Description
userDeviceName	The name of the user device to be loaded.

4.17. SaveSetup

Boolean SaveSetup(string filePath)

- Saves the current configuration (values of all the registers, non-field controls and pin values) into a .mac file into the file path passed.

Explanation for Parameters

Parameter_Name	Description
filePath	The complete file path to which the current configuration to be saved as .mac file.

4.18. RestoreSetup

Boolean RestoreSetup(string filePath)

- Restores the configuration from a .mac file passed that contains the values of all the registers, non-field controls and pin values to the currently loaded device

Explanation for Parameters

Parameter_Name	Description
filePath	The complete path of the file from which the field and non-field values to be restored.

4.19. RunScript

Int RunScript(string filePath,string[] scriptVariables)

- Sets the array of variables passed to the scope of the Python engine and Runs the Iron Python script located at the passed file path.
- Returns 0 if the script execution is completely successful, else returns -1.

Explanation for Parameters

Parameter_Name	Description
filePath	The path to the file that contains the python script to be run.
scriptVariables	The array of variables to be added to the Python engine scope, that might be used in the Iron Python script executed.

4.20. CheckModeText

Boolean CheckModeText(string modeName)

- Checks whether the mode name passed is available in the default configuration menu and returns true if one exists.

Explanation for Parameters

Parameter_Name	Description
modeName	The name of the mode which to check for existence in default configuration .

4.21. SetMode

Boolean SetMode(int Index)

- Loads the mode at the index passed in the default configuration menu.

Explanation for Parameters

Parameter_Name	Description
Index	The index of the mode to be loaded in the default configuration menu..

4.22. SetMode

Boolean SetModeText(string modeName)

- Loads the mode in default configuration menu that is matching with the mode Name passed.

Explanation for Parameters

Parameter_Name	Description
modeName	The name of the mode that is to be loaded from the default configuration menu.

4.23. SetPin

Boolean SetPin(string pinName,Boolean setFlag)

- Sets/ Clears the pin with passed pin name based on the setFlag.

Explanation for Parameters

Parameter_Name	Description
pinName	The name of the pin to be set / cleared.
setFlag	The flag that specifies whether the pin to be set or cleared. If it is true, the pin is set else cleared.

4.24. GetPin

int GetPin(string pinName)

- Returns 1 if the pin passed is set, else returns 0.

Explanation for Parameters

Parameter_Name	Description
pinName	The name of the pin which flag to be returned.