IPG-UI User's Guide

This document covers the usage of the Integrated Power Graphical User Interface (IPG-UI) software from Texas Instruments.

Contents

1 Introduction 3
2 Supported Features 3
3 Limitations 3
4 Revisions 4
5 Installation 4
6 Launching the GUI 6
7 Landing Page 7
8 Views Overview 11
9 Introduction View 13
10 Register Map View 18
11 Single Register 30
12 Register Controls 33
13 Device Controls 35
14 Macros 36
15 Adapter Controls 38
16 Status Bar 39
17 Quick Access Controls 43
18 File Menu 44
19 GUI Settings 45
20 Report 46
21 Errors and Notifications 47
22 Search 49
23 Frequently Asked Questions (FAQ) 51

List of Figures

1 Splash Screen 6
2 Landing Page 7
3 USB2ANY Selection 8
4 New Project Creation 9
5 Opening an Existing Project 10
6 Recent Devices and Projects 11
7 Introduction View 13
8 Device Description 14
9 Device Block Diagram 15
10 Download Data Sheet 16
11 Get Started Button 17
12 Register Map View 18
13 Register Selection 19
14 Read/Write ALL 20

15 Read/Write Group 21
16 Single Register R/W 22
17 Bit Descriptions 23
18 Favorite Registers 24
19 Order By Category 25
20 Order By Favorite 26
21 I^2C Address Control 27
22 Update Mode Control 28
23 AutoRead Control 29
24 Single Register View 30
25 Single Register Mini-Map 31
26 Generic I^2C Access 32
27 Register Controls View 33
28 Register Controls Auto Read 34
29 Device Controls 35
30 Macros View 36
31 New Macro Creation Screen 37
32 Status Bar 39
33 Transaction History 40
34 Hardware Connected Status 41
35 Changes Written Status 42
36 Quick Access Controls 43
37 File Menu 44
38 GUI Settings 45
39 Report 46
40 Error Notifications 47
41 Notifications 48
42 Search 49
43 Search Controls 50

1 Introduction

This software is created using web-based technologies such as Javascript, AngularJS, and Node-Webkit. This software requires using a USB2ANY adapter from Texas Instruments to communicate with the PMIC EVM. The USB2ANY adapter provides a USB interface to the host PC for receiving commands and then communicates with the PMIC EVM using the I^2C protocol.

The IPG-UI supports multiple devices with a single executable, which eliminates the need to install multiple GUIs when working with more than one device.

For details on how to configure the EVM for the device being used, see the EVM user's guide for that PMIC in the product folder of TI.com.

2 Supported Features

The IPG-UI supports the following features:

- Supports multiple PMIC devices with a single GUI
- Reading and Writing device registers
- Setting register values using individual bits, hexadecimal values, or human-readable controls
- Saving and Loading project files
- Tracking previous devices and projects for easy access
- Auto-reading registers at a variable polling interval
- Write on change or manual write options
- Searching registers and bits
- Organizing registers by address, category, or favorites
- USB2ANY communication with the EVM using I^2C
- Configuration of USB2ANY I^2C protocols
- Support for Microsoft® Windows® XP, Windows 7, Windows 8, Linux 32 and 64-bit (Ubuntu 14.04 LTS), and Mac OSX 10.10.2

3 Limitations

The IPG-UI has the following limitations:

- Windows 8 may require disabling the Enhanced Power Management for the USB2ANY device. If this is enabled, the USB2ANY adapter will blink when it is connected and the IPG-UI application is not running. For details on how to disable this feature, see the Microsoft® Windows USB Core Team Blog help topic. The VID and PID of the USB2ANY adapter are 0x2047 and 0x0301.
- I^2C is the only supported protocol to communicate with the device.
- Only the USB2ANY adapter is supported. Other adapters such as USB-to-GPIO or EV2300 are not supported. If you do not have a USB2ANY adapter, contact your local TI sales team to obtain one.
- Requires USB2ANY firmware version 2.7.0.0. For information on updating the firmware, see the *Frequently Asked Questions (FAQ)* section.
- If the USB2ANY adapter is in a locked state (that is, the LED on the adapter is blinking), the IPG-UI will freeze at the splash screen until the adapter has been unplugged. Reconnecting the adapter will be detected and the GUI will then be able to access the USB2ANY device. A common reason for the USB2ANY to get into a locked state is when the host PC is disconnected from the USB2ANY device while the USB2ANY device remains powered. For example if the USB2ANY is connected through a powered USB hub which provides power to the USB2ANY and the host is disconnected from the hub this issue can occur.

4 Revisions

This section details the features added with each release of the IPG-UI.

1.0.0.0 This is the initial release containing the features listed in the supported features section.

1.0.0.2

- Fixed issue with single register read to configure the adapter protocol for unknown register groups
- Added support for Linux and Mac OSX installers
- Fix I^2C address list for 10-bit addresses

2.0.0.0 Added support for the following features:

- Updated node-webkit version to 0.12.3
- Added support for Macros to enable replaying common command sequences
- Added support for Device Controls to configure device level settings
- Added advanced controls to allow the GUI to reflect the device status more accurately
- Additional device support files added to the base installation
- Preserve selections when switching between views
- Enhanced transaction history to show pre-commands and post-commands as well as which commands were sent as a single block
- Added a mini register map to the Register Controls view
- Added confirmation of write of unknown values
- Added autosave GUI option
- Updated styling of GUI elements

5 Installation

5.1 Windows

The IPG-UI is delivered as a Windows installer and supports the following features:

- Selection of installation path. TI recommends using the default installation path.
- Optional creation of a desktop shortcut to launch the GUI
- Creation of a Start Menu shortcut in the Texas Instruments directory
- Option to launch the GUI after installation
- Uninstaller which can be run through the Windows Programs and Features page in the Control Panel

After installation the IPG-UI will be installed by default into the C:\Program Files (x86)\Texas Instruments\IPGUI- <version>/IPG-UI directory. The executable name is nw.exe. In this directory you can also find the software manifest for the IPG-UI which details the software components used and their corresponding licenses.

5.2 Linux

The IPG-UI is delivered as a .run installer for Linux and supports the following features:

- Selection of installation path. TI recommends using the default installation path.
- Optional creation of a desktop shortcut to launch the GUI
- Prompt for sudo password to configure the host Linux system to access the USB2ANY adapter
- Option to launch the GUI after installation
- An uninstaller program in the installation directory for removing the GUI

In order to properly install the GUI the following pre-requisites should be met:

1. The user should have sudo permissions. In order to access the USB2ANY adapter and run the GUI udev rules and symlinks must be created along with a usb2any group. The user will be prompted during installation to provide their sudo password. If the password is incorrect or the user does not have sudo permissions they can re-run the IPG-UI-linux-setup.sh script installed in the installation directory to finish the setup at a later time.
2. The GUI installer should be marked as executable
3. For Ubuntu 14.04, the latest version of node-webkit and node-hid require GCC and libstdc 5.0 libraries. During the setup script execution, the following commands will be attempted to download and configure these software packages. This download and configuration will take additional time during installation.
   - sudo add-apt-repository ppa:ubuntu-toolchain-r/test
   - sudo apt-get update
   - sudo apt-get install g++-5

   If you are installing this GUI on a system behind a firewall you must export the http_proxy and https_proxy environment variables before installation in order for these commands to work. If this is not done then the libraries will not be updated and the GUI will not find the USB2ANY adapter. The user can then either run these commands manually or export the proxy configuration and re-run the IPG-UI-linux-setup.sh script.

As mentioned in the pre-requisites during installation the user will be prompted to provide the sudo password in order to finish configuration on their system. The same will also occur during uninstallation of the GUI.

After installation the IPG-UI will be installed by default into the user's home directory in Texas_Instruments/IPGUI-<version>/IPG-UI as IPG-UI.

NOTE: It is required that the user logout and login again for the changes to the udev rules and groups to take effect. Without this the GUI will be unable to find the USB2ANY adapter.

NOTE: You may need to unplug and reconnect the USB2ANY adapter after installation for the new udev rule to take affect. The detection process can take a few seconds before the USB2ANY adapter is detected.

5.3 OSX

The IPG-UI is delivered as an .app.zip installer for OSX and supports the following features:

- Selection of installation path. TI recommends using the default installation path.
- Option to launch the GUI after installation
- An uninstaller program in the installation directory for removing the GUI

In order to install on OSX the user should double-click on the .app.zip file to extract the application installer. Once the application installer is extracted the user can double-click it and the installation will begin. The IPG-UI will automatically be added to the list of applications in the application launcher under a Texas_Instruments group.

After installation the IPG-UI will be installed by default into the /Applications/Texas_Instruments/IPGUI-<version>/IPG-UI directory as IPG-UI. This is an application directory and can be run by double-clicking it in the finder.

6 Launching the GUI

The easiest method to run the IPG-UI is to use the desktop shortcut if it was created. If no desktop shortcut was created then you can browse to the installation directory referred to above for your OS and double-click the executable. For Windows systems there is also a shortcut created in the Start Menu in the Texas Instruments directory that can be used.

When the GUI begins execution, the GUI extracts its components to a temporary directory on the host PC and launches the Node-Webkit browser using those components to render the GUI web page.

It should be noted that any changes made to files inside of the temporary directory will be lost when the GUI exits. The default project save and open paths point to the following directories based on the host OS:

Windows: user's <My Documents>/Texas Instruments/IPG-UI directory

Linux: <user's home>/Documents/.Texas_Instruments which is a hidden directory in the user's home directory

OSX: <user's home>/Documents/.Texas_Instruments which is a hidden directory in the user's home directory

When the GUI first launches, you will see the splash screen which shows the status of the GUI initialization on the bottom left.

Figure 1. Splash Screen

After the GUI is initialized, the splash screen will clear and you will be taken to the GUI landing page

7 Landing Page

After the splash screen has cleared the GUI will show the device landing page. This page is used to allow the user to select which device to work with and the USB2ANY adapter to use. The following sections detail the options available on the landing page. After a device or project has been selected on the landing page, the GUI title bar will update with the device name.

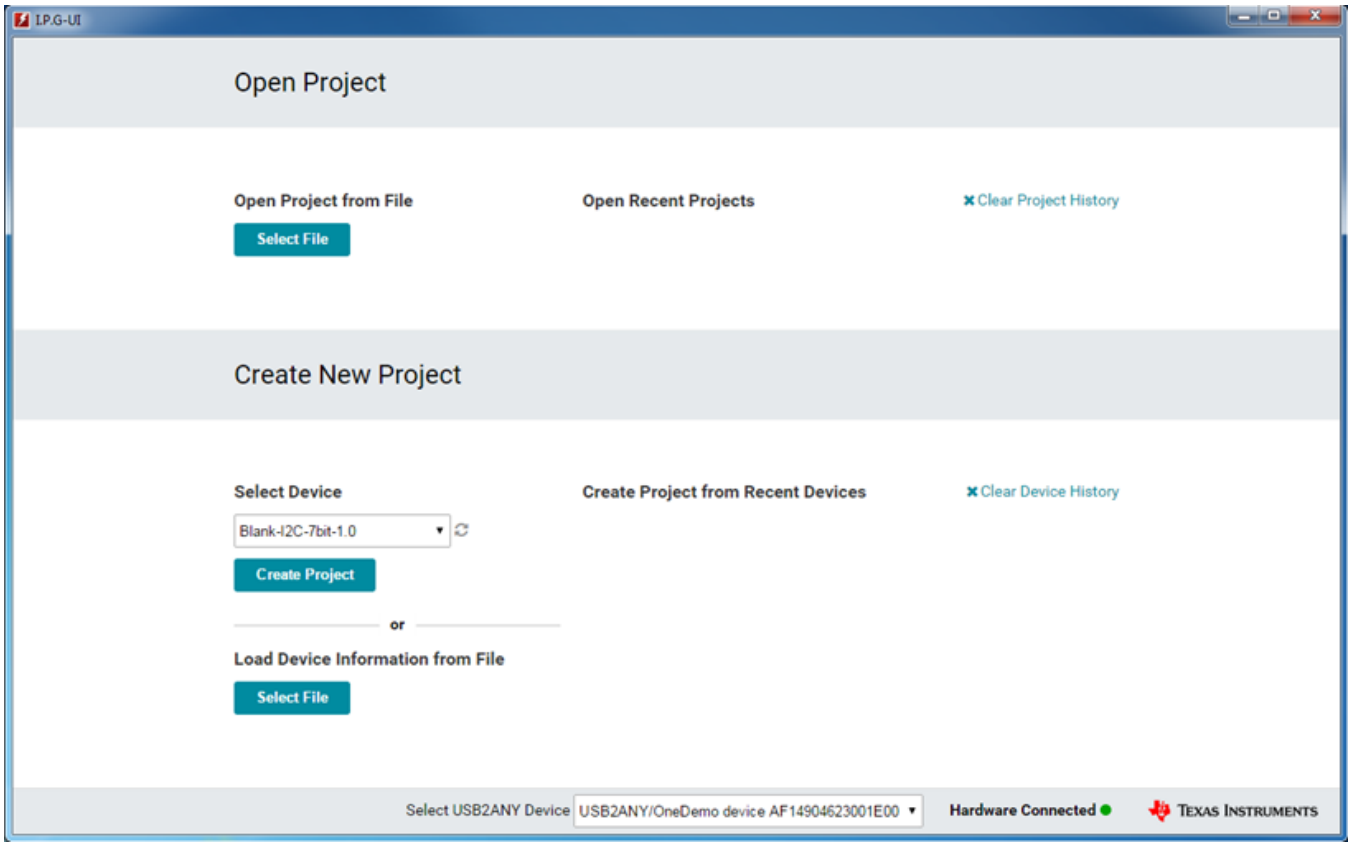

Figure 2. Landing Page

7.1 Select USB2ANY Adapter

In the case of multiple USB2ANY adapters being connected the user can select which one to use from the dropdown box at the bottom of the landing page. See the screen capture in Figure 3.

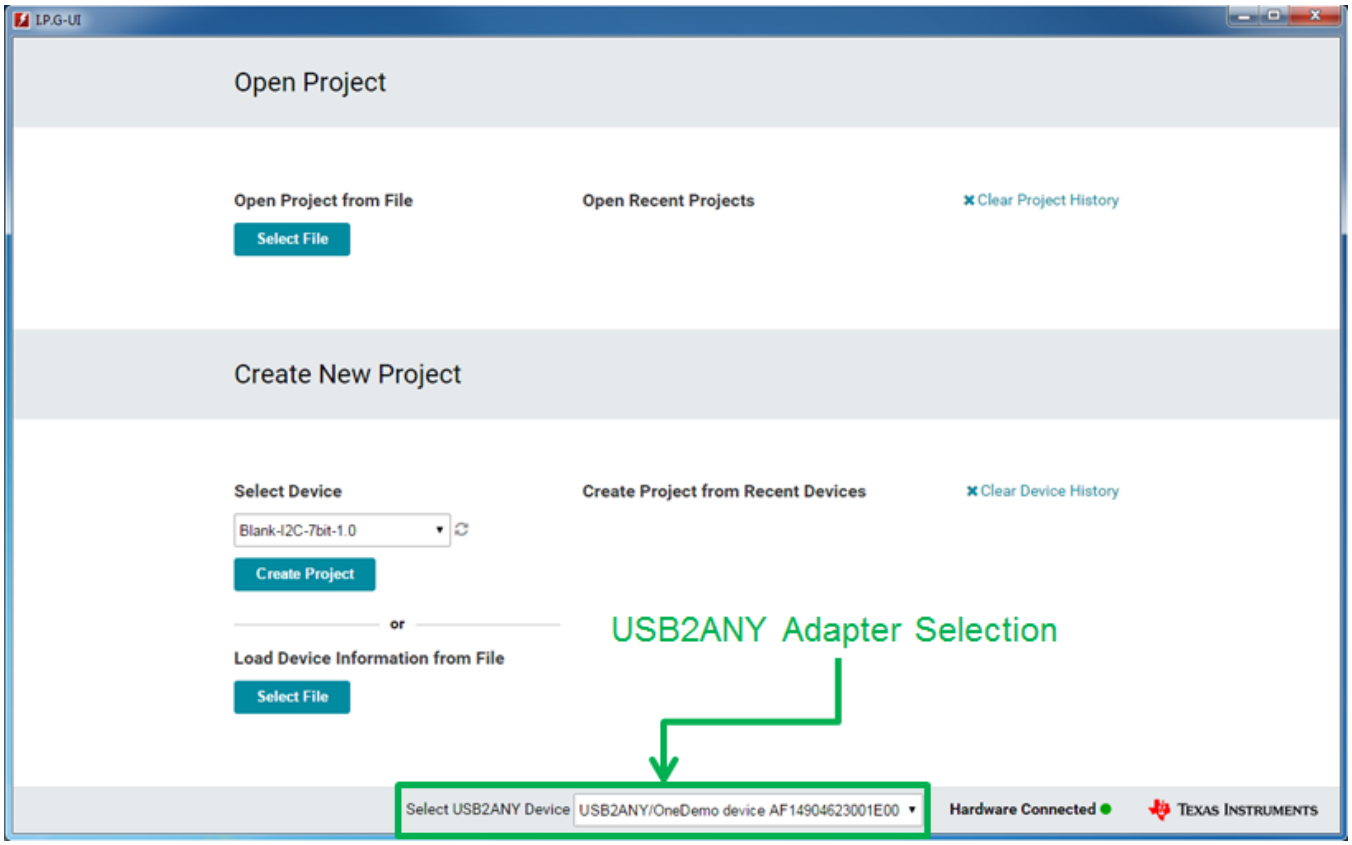

Figure 3. USB2ANY Selection

7.2 Create a New Project

In order to create a new project for a device the user can select from the following options:

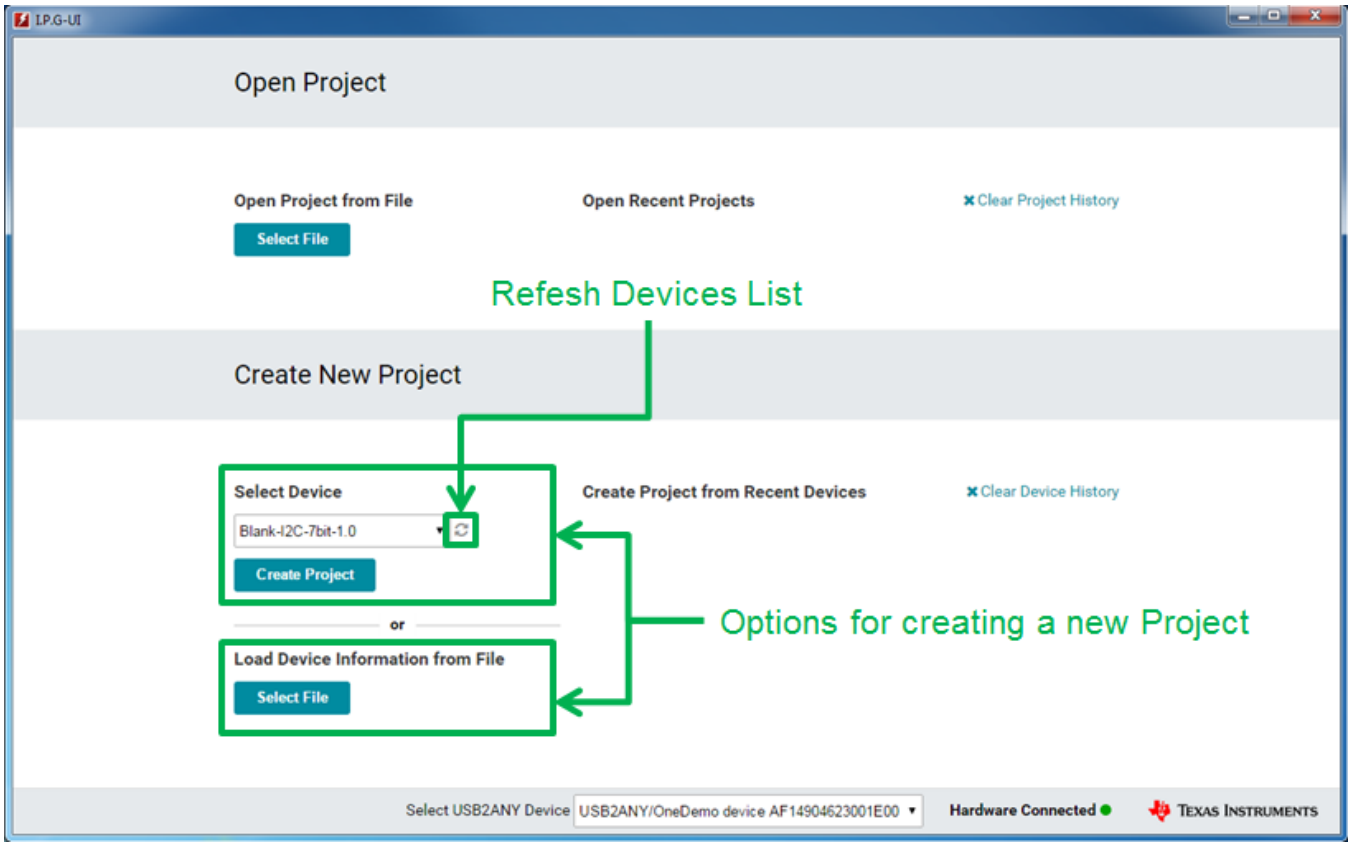

Figure 4. New Project Creation

1. Select Device – This option will give the user a list of pre-packaged devices supported by the IPG-UI as a dropdown. Once the device has been selected the Create Project button should be clicked to create the project and open the device introduction page for the selected device.
2. Load Device Information from File – This option allows the user to load a stand-alone device configuration file that is not part of the pre-packaged list and create a project. Clicking the Select File button will open a file explorer dialog where the user can browse to the file they wish to create a project from.

It should be noted that creating a project works with device description files and not with existing project files. Attempting to open an existing project file will give trigger an error notification. To open and existing project see the next section.

NOTE: The refresh button can be used to scan for additional device support files that have been installed while the GUI is still running and add them to the list of available devices.

7.3 Loading an Existing Project

To open an existing project the user should use the **Select File** button in the **Open Project** section of the landing page as shown below. Clicking this button will open a file explorer dialog where the user can browse to the project file they wish to load.

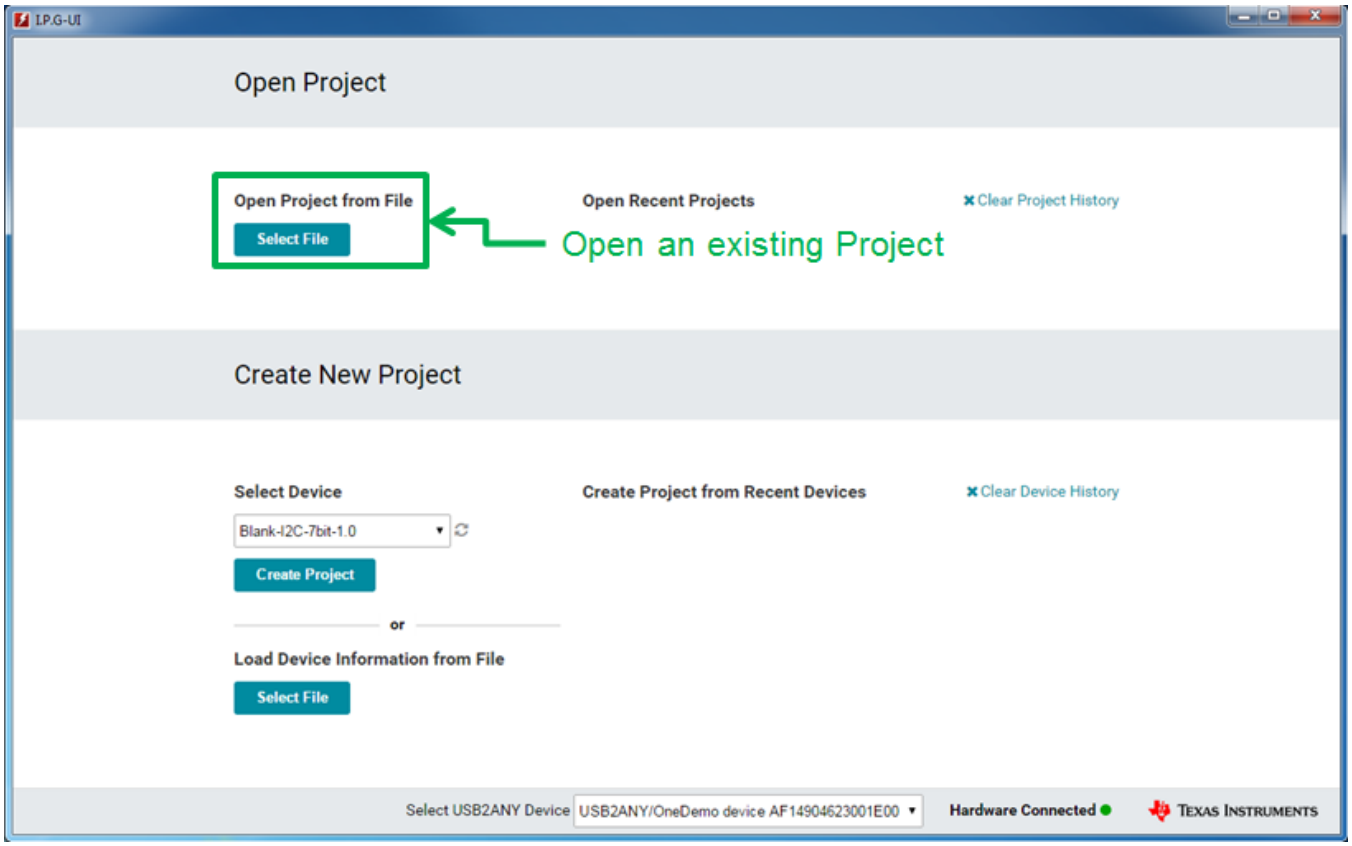

Figure 5. Opening an Existing Project

Attempting to open a device description file using the **Open Project** feature will display an error on the screen.

7.4 Recent Projects and Devices

For both **Create New Project** and **Open Project** the most recent three files opened are listed to the right of the selection area. Clicking one of these entries will re-open that file and provides a shortcut method for opening previous devices and files without requiring browsing the user file system. Each list of recent devices can be cleared using the Clear history link for that section.

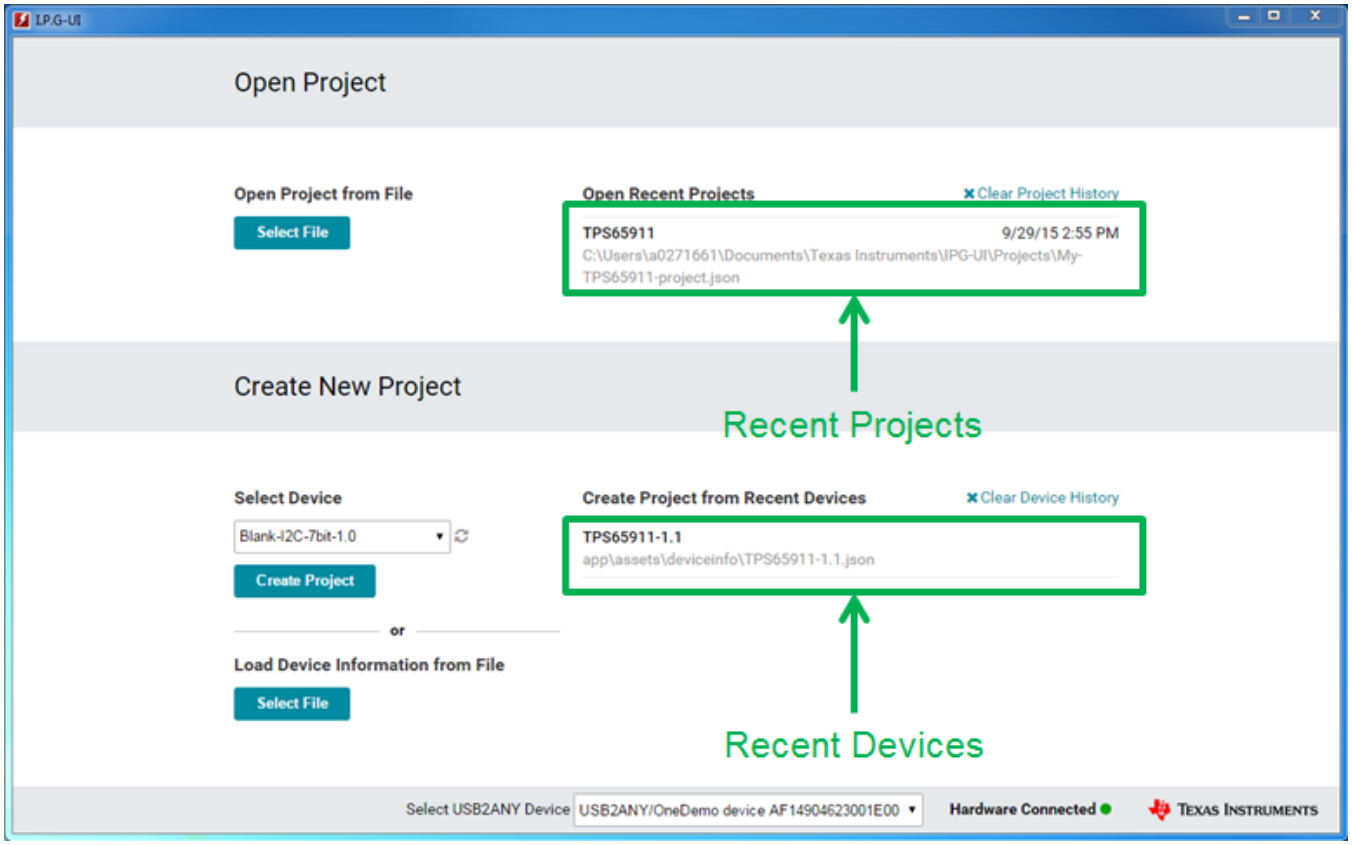

Figure 6. Recent Devices and Projects

Once you have either created a new project or opened an existing project you will be taken to the introduction view for that device. This will be covered in later sections.

8 Views Overview

This section provides an overview of the views available through the navigation pane on the left side of the GUI after a device or project file has been selected. For details of operations available in each view please see the following sections in this guide.

8.1 Introduction

The **Introduction** view provides the user with a description of the device, a block diagram of the device, and a link to the online device data sheet. Clicking the **Get Started** button will take the user to the **Register Map** view where they can begin to interact with the device.

8.2 Register Map

The **Register Map** view provides the user with a representation of the device register map where values can be read or written. If a device is not connected with a USB2ANY adapter the user is still able to make modifications to the register map and save them as a project to enable writing them to the device at a later time when the hardware is connected.

8.3 Register Controls

The **Register Controls** view shows all of the human readable controls defined for the device organized by categories. Changes to the controls will automatically set the corresponding bits in the register map. This is provided to make it easier to configure the device without requiring looking up bit settings in the data sheet.

8.4 Single Register

The **Single Register** view allows the user to work with only a single register at a time. This is particularly useful when using a device such as the blank I^2C device which will allow the user to read and write to any register on any I^2C bus. If a register is a known register for the device selected then the information for that register will be displayed as well.

8.5 Adapter Controls

The **Adapter Controls** view is used to configure the USB2ANY adapter settings such as pull-ups, I^2C address length, and I^2C address protocol.

8.6 Device Controls

The Device Controls view shows controls that are not tied to a specific register but rather are intended to be used to control the EVM or device. These controls can be items such as whether to enable auto password setting, or to run particular command sequences to enable some functionality of the Device. This view will only be available if the selected device has Device Controls declared.

8.7 Macros

The Macros view is always available and will show the user any existing macros that are already declared for the device selected. Macros are sequences of commands that can be re-run with a single button click to help make complex or frequently-performed operations easier. Even when no pre-defined macros exist, the Macros view can be used to allow the user to create custom macros from the transaction history of commands they have sent to the device.

9 Introduction View

This section will cover the introduction view for a device and the features available.

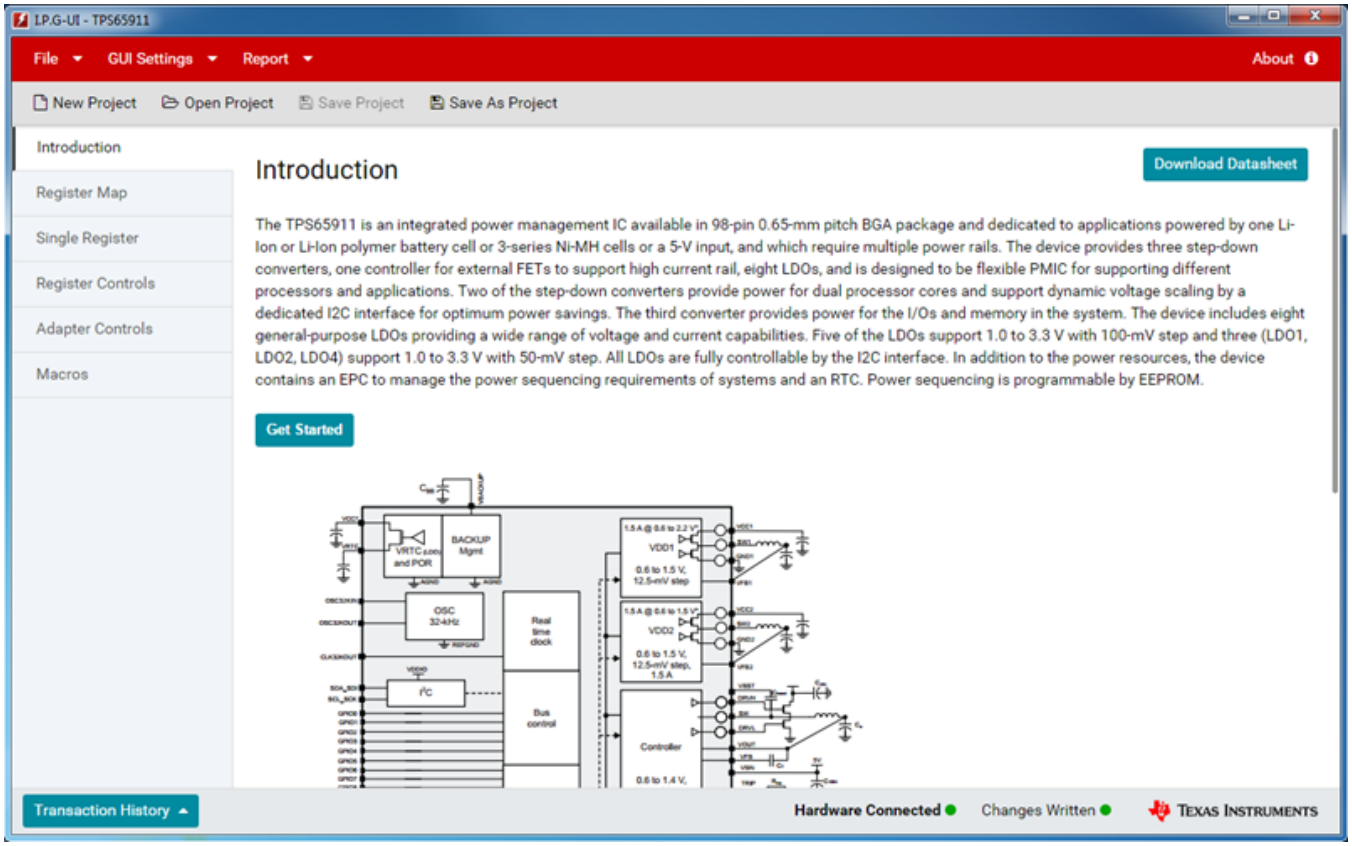

Figure 7. Introduction View

9.1 Device Description

The device description gives an overview of the device features and uses. It is presented as a paragraph of text at the top of the introduction view.

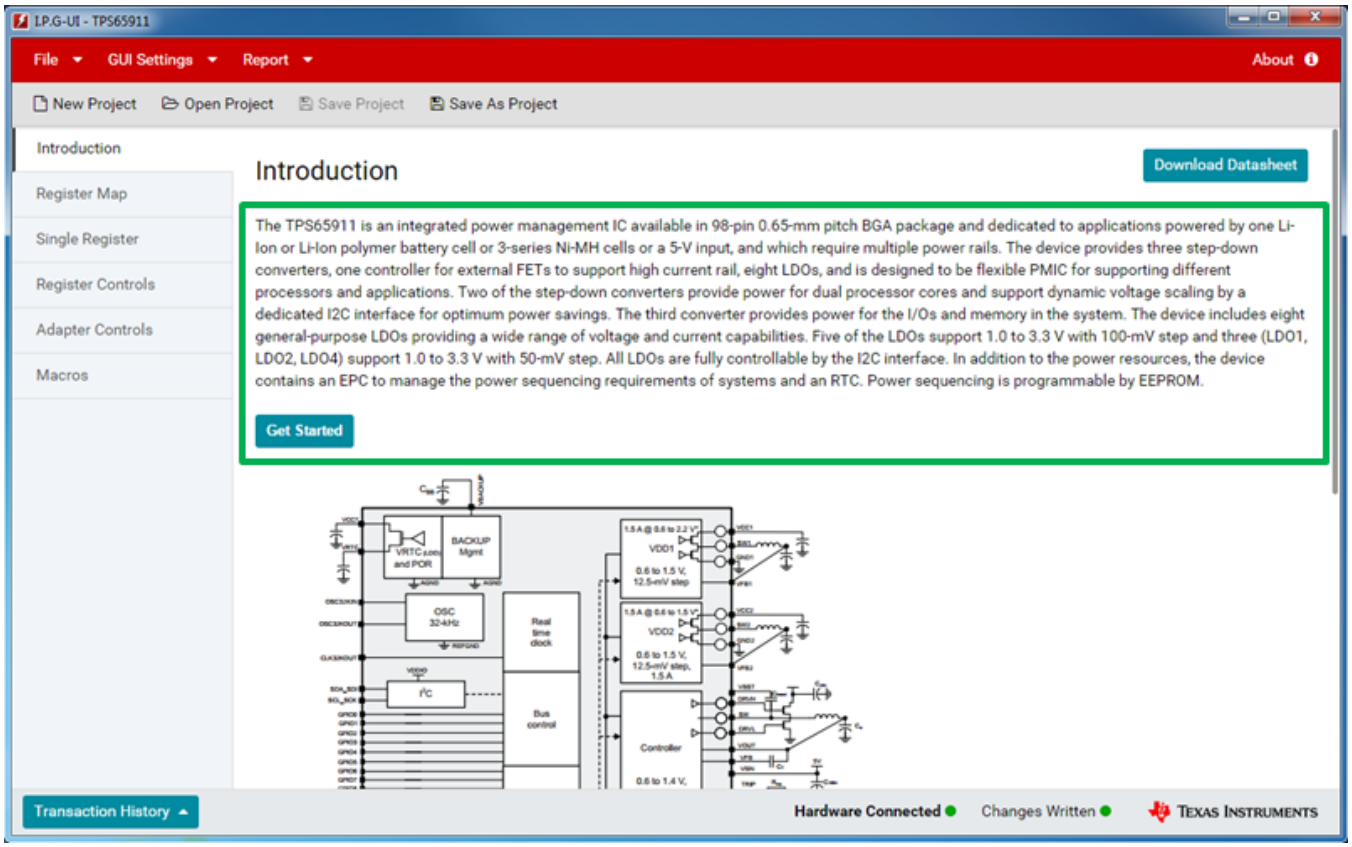

Figure 8. Device Description

9.2 Device Block Diagram

The device block diagram will be shown at the bottom of the introduction view. If the block diagram scrolls off the screen the user can scroll or resize the GUI to see the rest of the image.

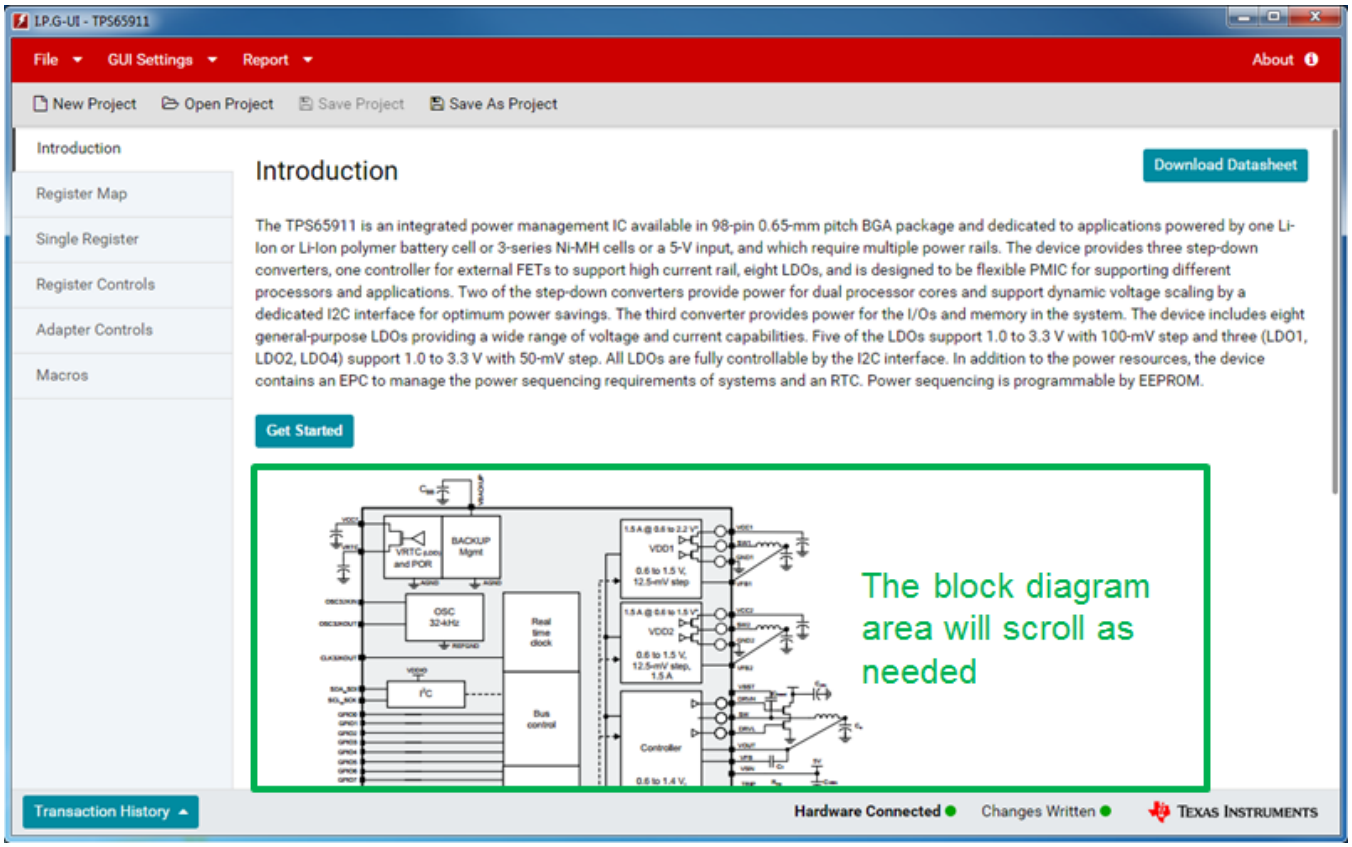

Figure 9. Device Block Diagram

9.3 Download Data Sheet

The download data sheet button in the upper right corner will open the user's browser to the data sheet for the selected device on TI.com.

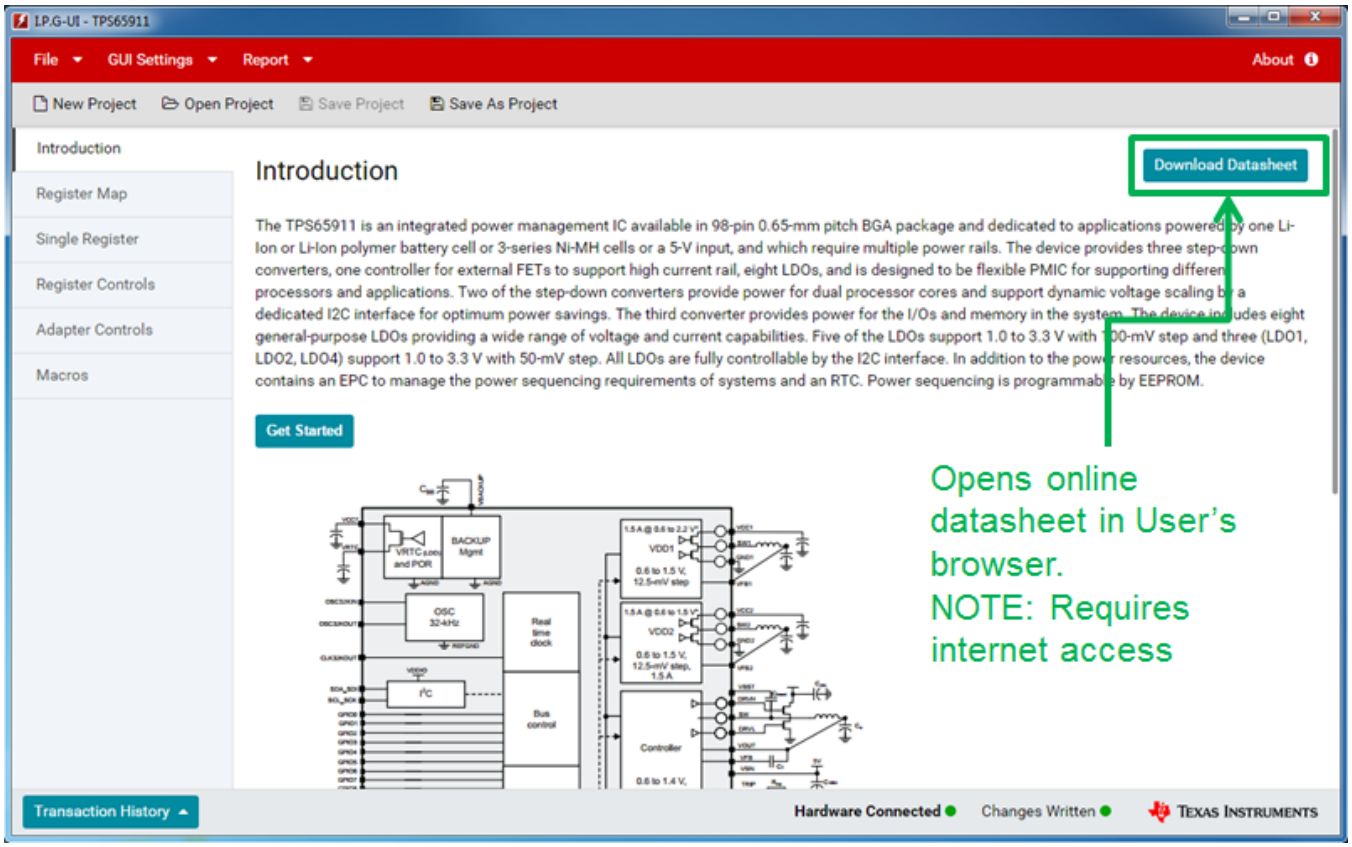

Figure 10. Download Data Sheet

9.4 Get Started

The **Get Started** button located between the device description and device block diagram will take the user to the register map for that device.

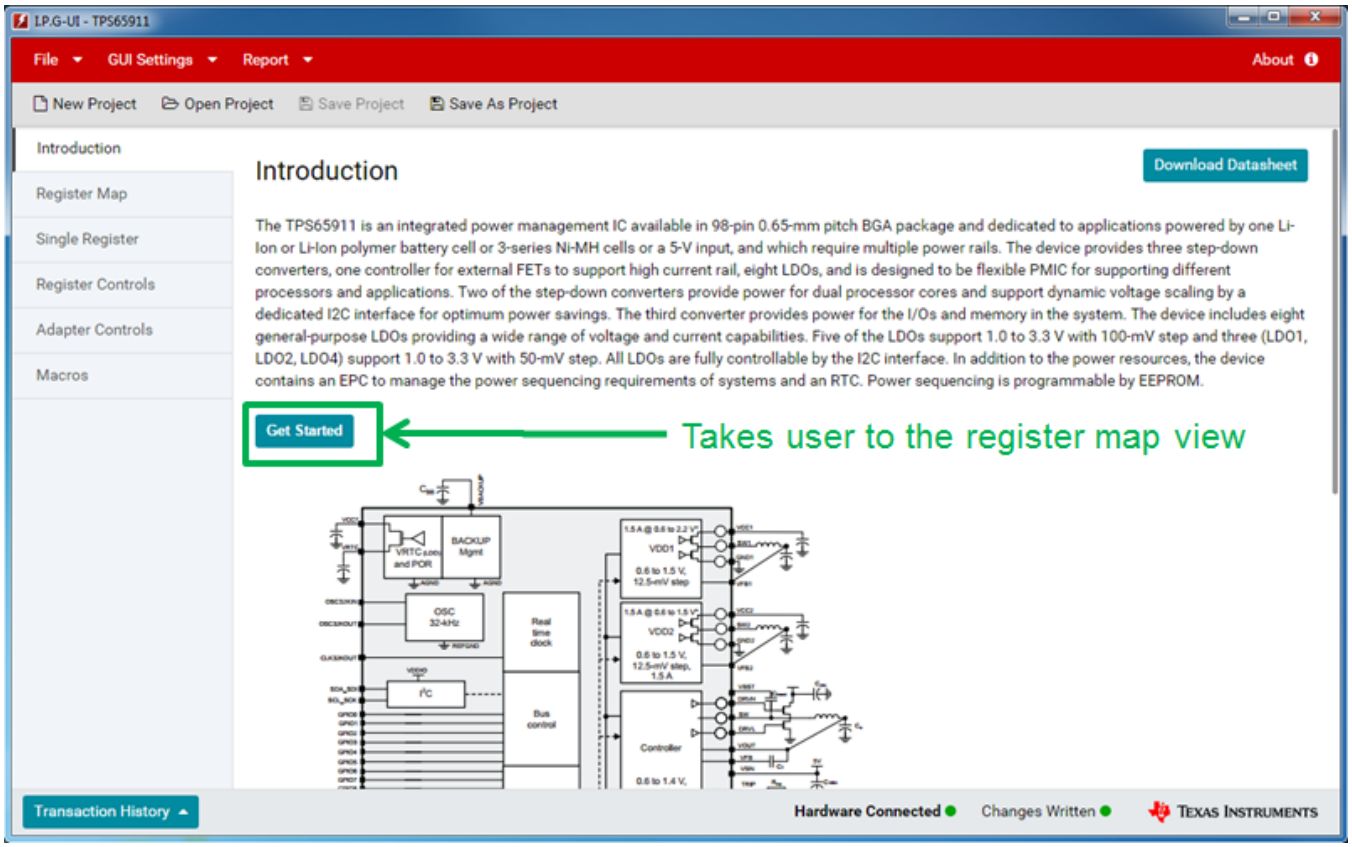

Figure 11. Get Started Button

10 Register Map View

This section will cover the register map view and the available operations and features. This is the default view after the **Introduction** view.

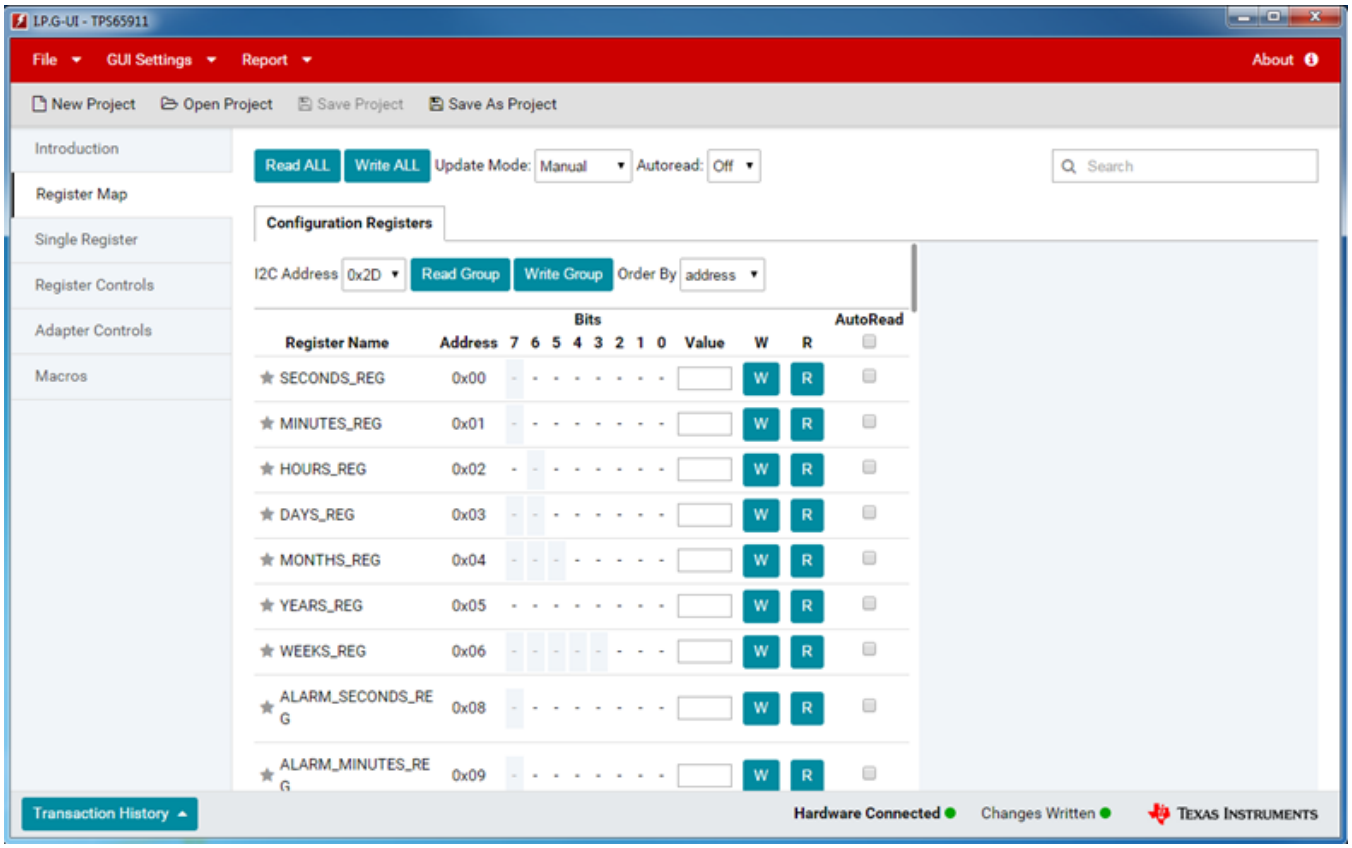

Figure 12. Register Map View

The register map can have multiple tabs with each tab representing a bank of registers. Within each tab is a scrollable table of registers that the user can interact with.

10.1 Selecting a Register

When the user clicks on a register the register will be marked as active by having the background for the register row turn to gray. This has the additional effect of populating the register description section to the right side of the GUI.

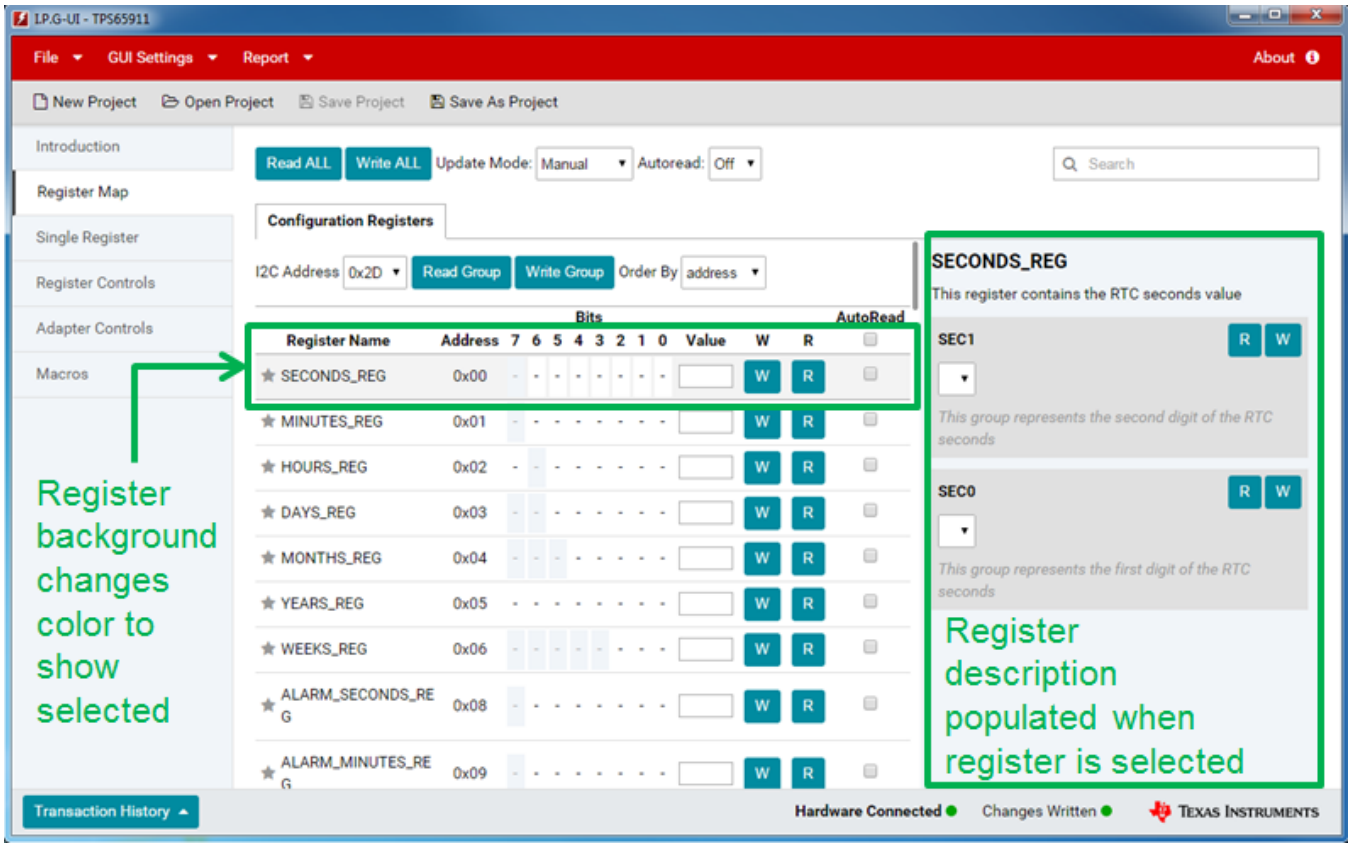

Figure 13. Register Selection

The register description area provides information about the selected register as well as the defined register controls. These controls can be used to modify the bits in the register as described in later sections.

When hardware is connected and any of the following actions are performed the GUI will automatically read the register value:

1. Any bit in the register bits is clicked
2. Any control in the register group is changed
3. The user selects the Value field and then moves focus elsewhere in the GUI (blur operation)

Additional register read controls are detailed in the following section.

10.2 Read/Write Registers

The following controls are available for reading and writing registers:

1. **Read/Write ALL** – The **Read ALL** and **Write ALL** buttons located at the top of the register map view will read and write all registers in all banks. Many devices only define one register bank but for devices with more than one register bank this allows working with all registers at once.

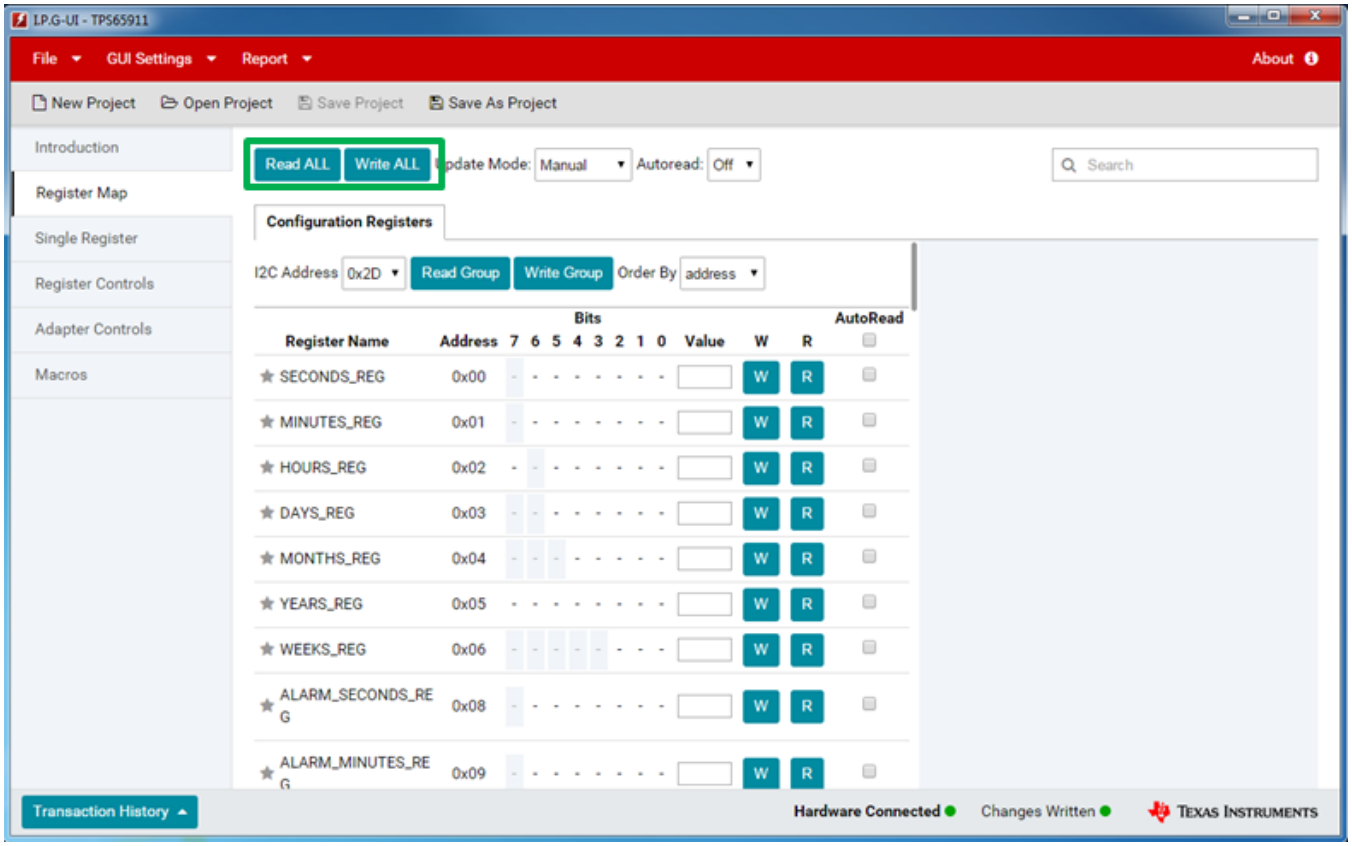

Figure 14. Read/Write ALL

2. **Read/Write Group** – The **Read Group** and **Write Group** buttons located inside of each bank of registers enable the user to read or write only that bank of registers. All registers in that bank will be read or written when this button is clicked.

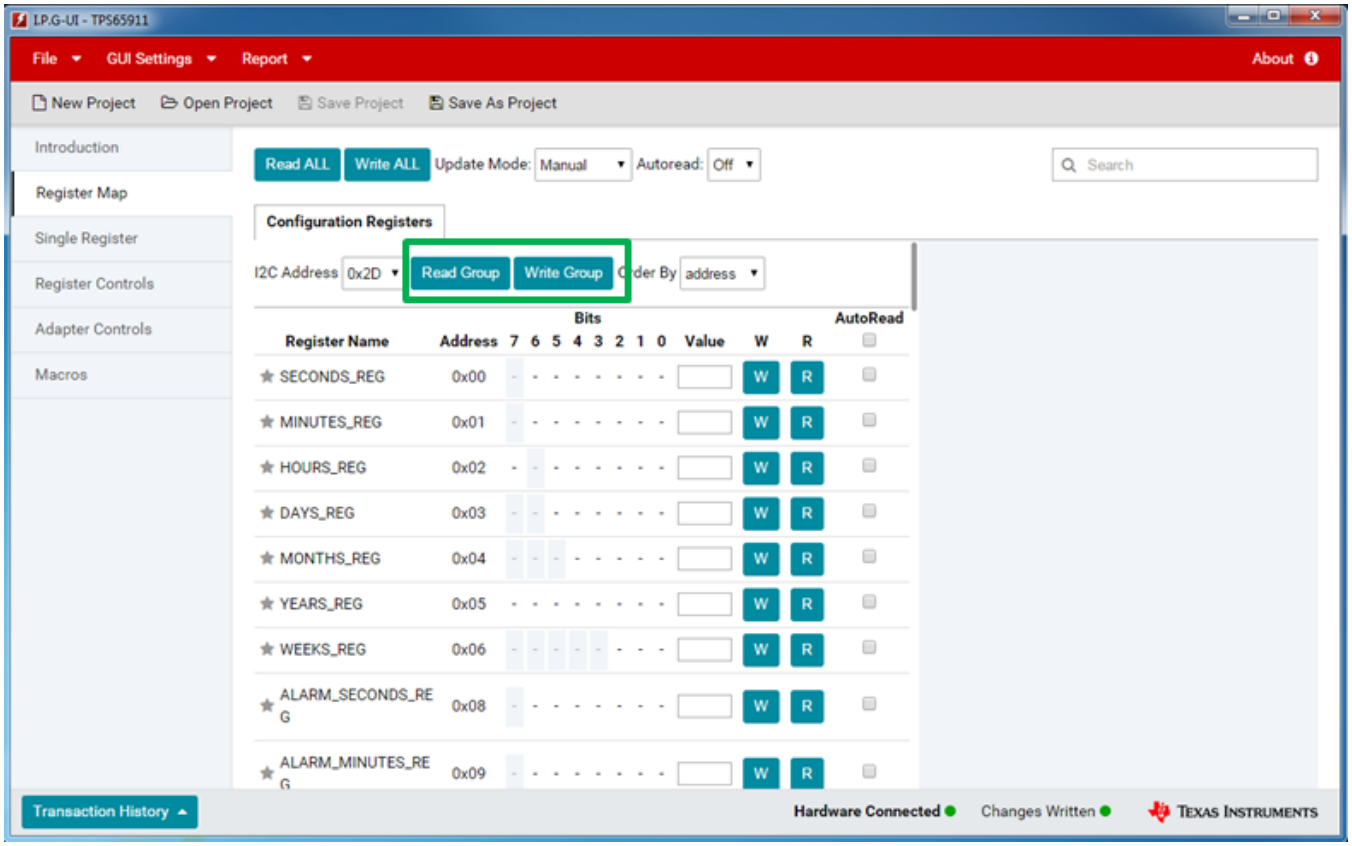

Figure 15. Read/Write Group

3. **R/W Buttons** – Each register has a read (**R**) and write (**W**) button in the register row in the register map. These buttons will read or write only that single register. For registers that are read-only the write button is disabled and only the read button is available. Additionally each control group in the register description field will have a read and optionally a write button which can be used to read or write the register.

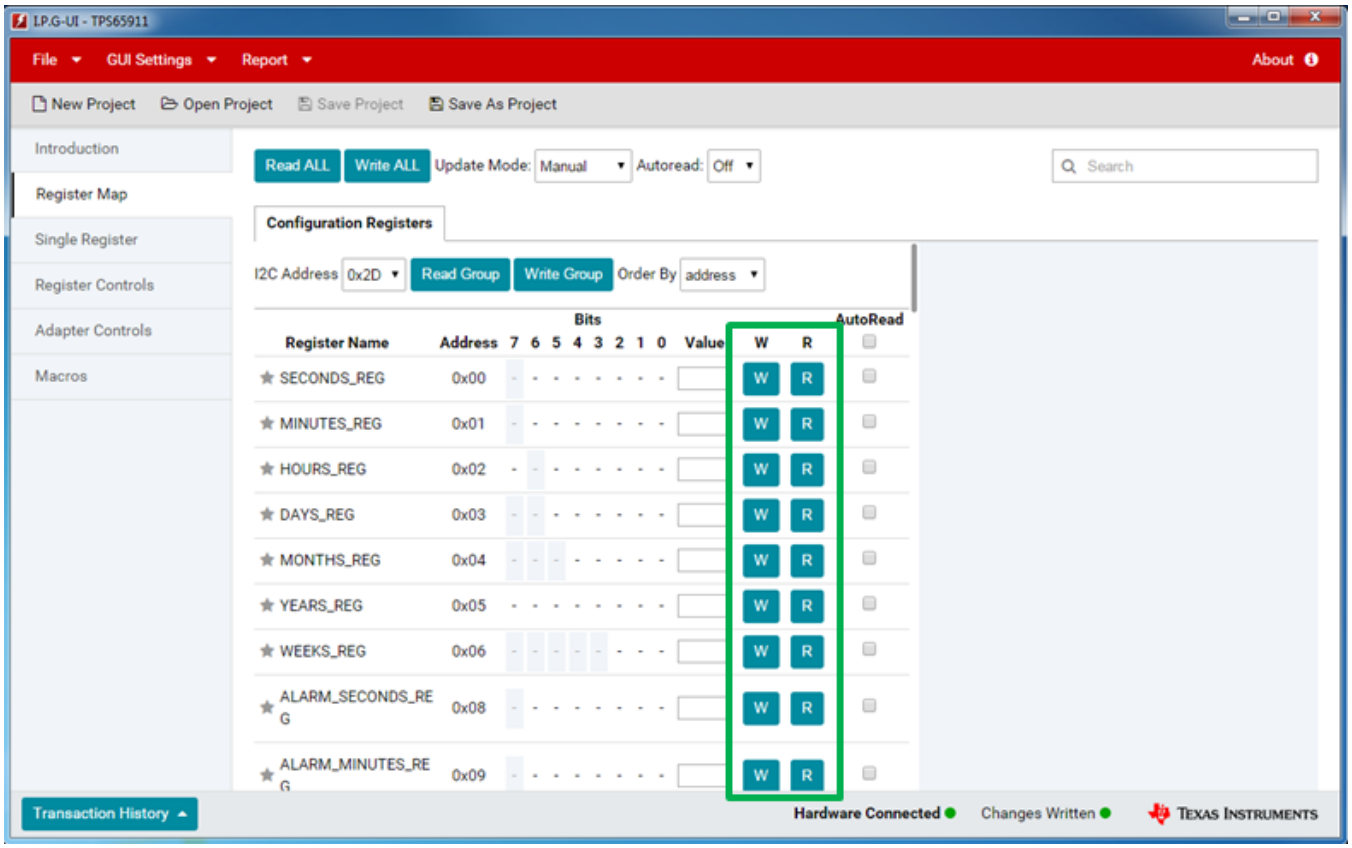

Figure 16. Single Register R/W

10.3 Changing Register Values

Register values can be changed using any of the following methods:

1. **Clicking a Bit** – By clicking on any bit in the register map the following action will occur. If the register has not been read then the register will be read from the device if the hardware is connected. If no hardware is connected then the register bits will be initialized to zero. Clicking on a bit in a register that has already been read and initialized with change the bit value.
2. **Using the Value Field** – The value field can be used to enter the hex value for the register. This is useful when setting all the bits in the register to a specific value rather than having to modify each bit individually. If the hex value entered attempts to overwrite reserved or read-only bits it will be modified accordingly to preserve the settings in those bits and only change the writeable bits. For read-only registers the value field will not allow for user input.

NOTE: Hex values should be entered without the leading *0x*. For example, *0xff* would be entered as *ff*.

3. **Using the Control Groups** – The control groups in the register description field can be used to modify the register values. Each control group will modify only the bits that are part of that group and those changes will be reflected in the register map table. For read-only bits the control group will only display the information of the meaning of the current value and will not allow for modifying the value.

10.4 Coloring/Highlighting

The following colors are used in the register map and register description fields to indicate status:

- **Gray Register Row** – This is used to indicate which register is selected. The selected register will have its description populated in the register description field.
- **Yellow** – This is used to show highlighted bits. The use cases for highlighting bits are:
  - Moving the mouse over a bit will highlight all the bits in the group that bit belongs to, if any. Further if the register is active then the corresponding control group in the register description will be highlighted as well.
  - Moving the mouse over a control group in the register description will highlight all bits that are part of that group in the register map table.
- **Orange** – This is used to show the current search result
- **Blue –** When a bit is changed from the last value read the background for the bit and the bit text will turn blue to indicate a change. Either reading the register again or writing the register will clear this color change. When a register is first read the default comparison value is zero. This means that any values of 1 read from the device will be marked as changed so that the user can see the shift in bit value.
- **Light Blue Bit** – A light blue bit is a read-only bit. Clicking on this bit will cause a register read for an uninitialized register, but any further clicks will not perform any action. The mouse icon will change when it is over a read-only bit for the selected register to indicate that the bit cannot be clicked.

10.5 Hovering

Hovering the mouse over any bit in the register map will display information about the register and that bit in the register to the user. This enables the user to find out more information about a particular bit.

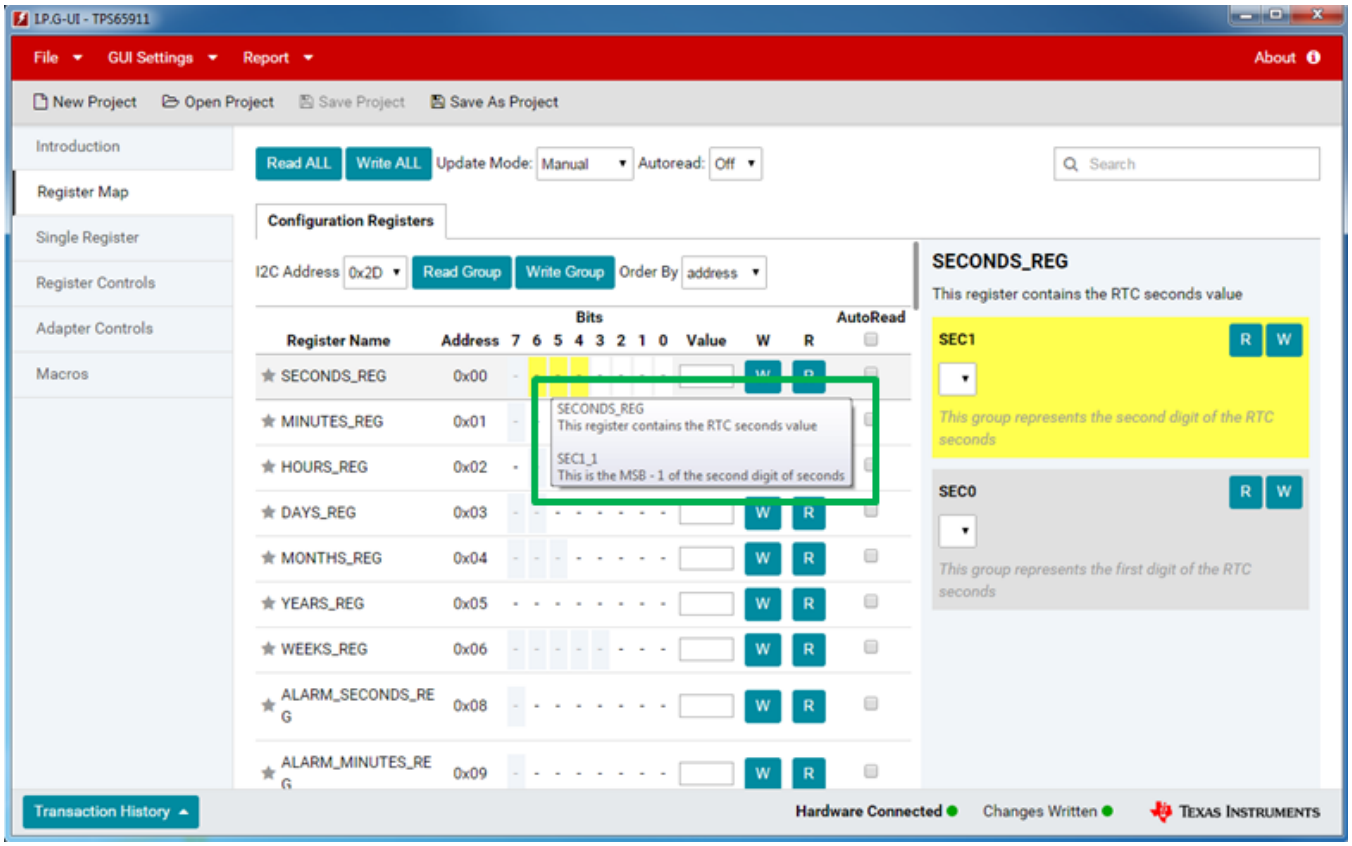

Figure 17. Bit Descriptions

10.6 Marking a Register as Favorite

The star icon at the left of each register row enables the user to mark the register as a favorite register. Marking a register as favorite will change the star to yellow while non-favorite registers will have a gray star. It is possible to sort the register map to bring favorite registers to the top of the map as detailed in the **Order By** section.

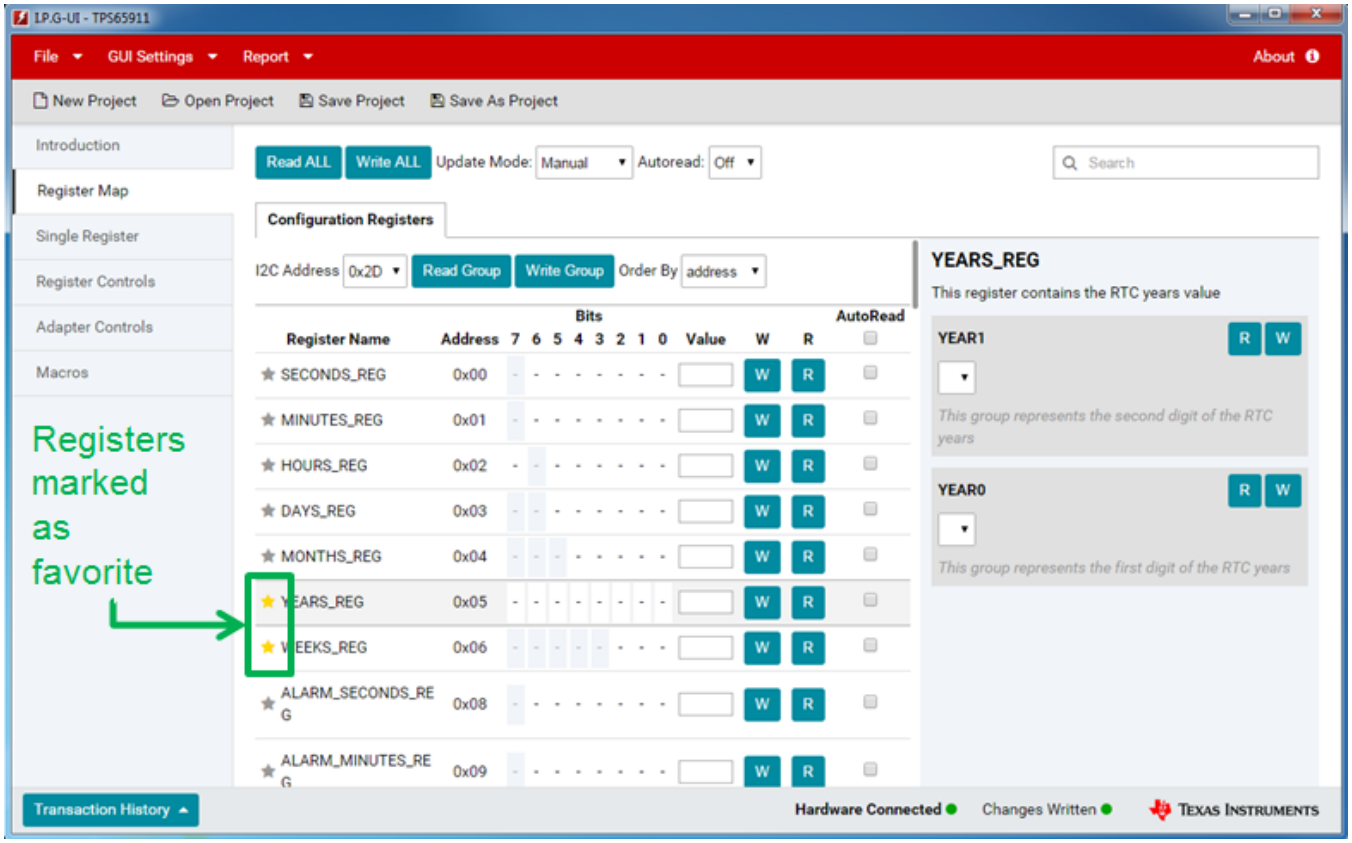

Figure 18. Favorite Registers

10.7 Order By

The **Order By** control determines how the register map is sorted. By default the registers are sorted by address from lowest to highest. However the use can also select to order the registers by the following:

- **Category** – This will group registers with the same category together. This is useful to help bring all registers related to a similar topic together.

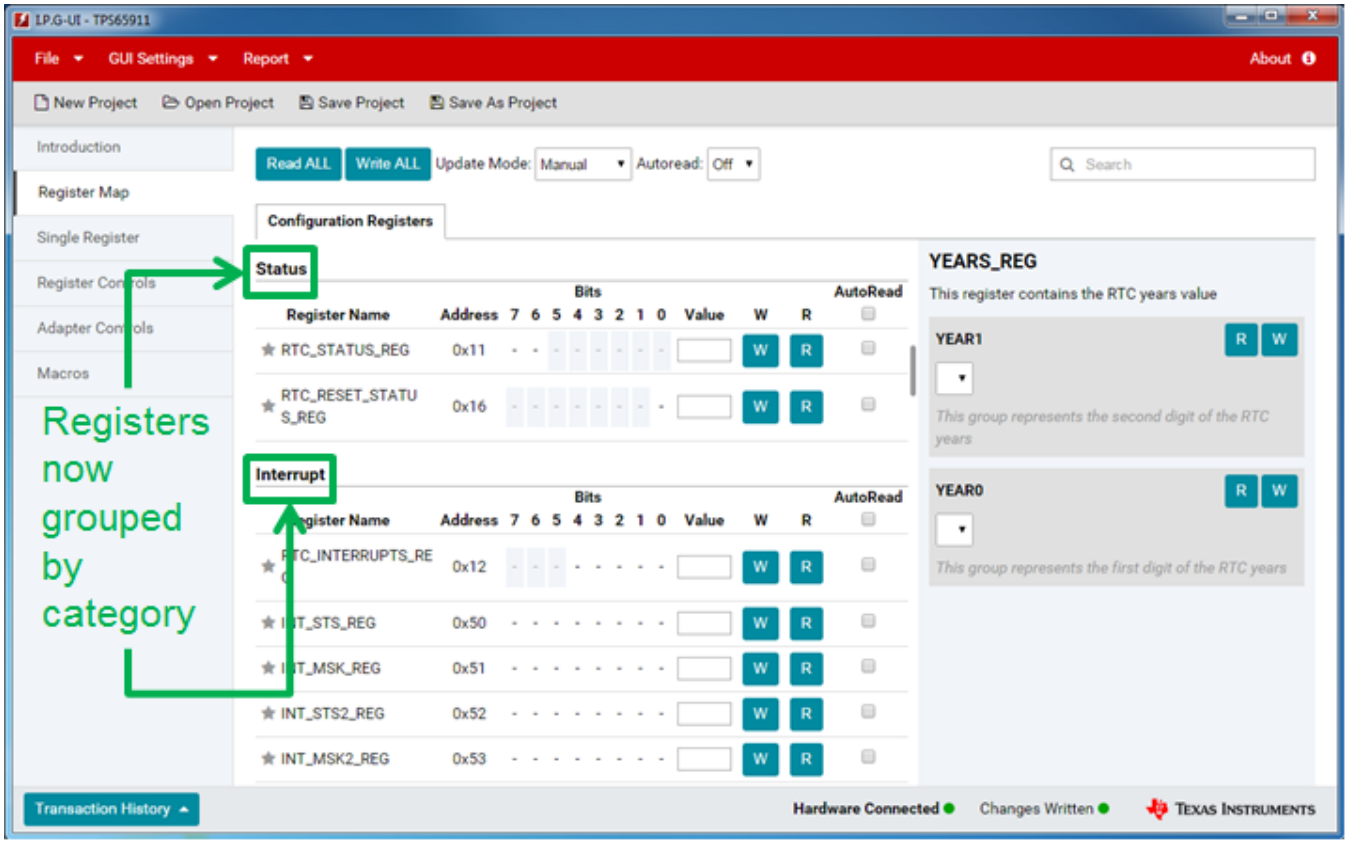

Figure 19. Order By Category

- **Favorite** – This will move all registers that have been marked as favorite to the top of the register map. All other registers will be listed below the favorite registers. This is useful when the user wishes to interact with a set of registers repeatedly that are scattered throughout the register map. By moving these registers to the top of the list the user can avoid having to scroll the register map between registers.

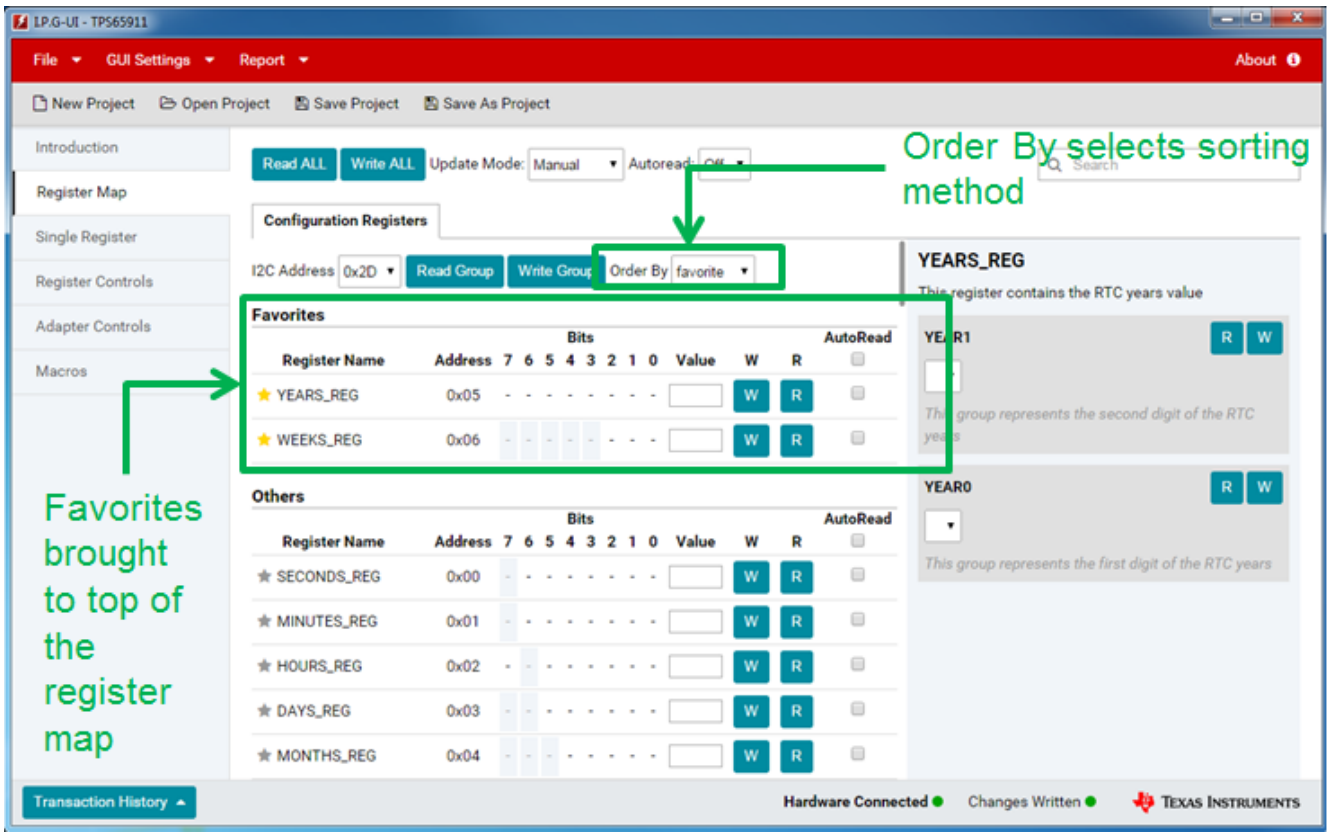

Figure 20. Order By Favorite

10.8 I²C Address

The **I²C Address** control which is found as part of each register group allows the user to change the I²C slave address for that bank of registers. The default value is set to work with the TI EVM but for devices with a configurable I²C slave address this enables the user to select the modified address if they change the I²C slave address from the default.

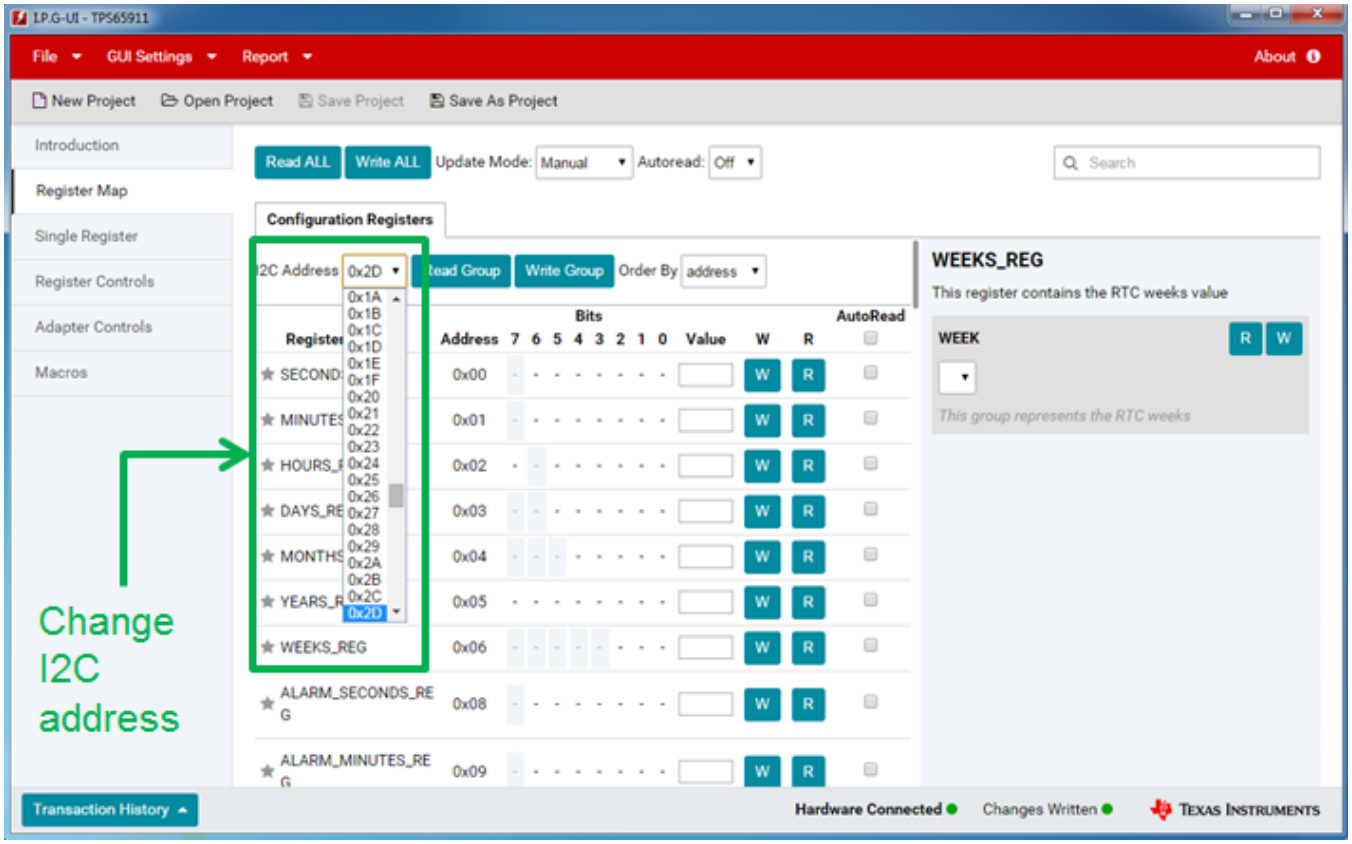

Figure 21. I²C Address Control

10.9 Update Mode

The **Update Mode** control found at the top of the register map view can be used to configure when write operations are performed. There are two settings available:

- **Manual** – This is the **default** mode and means that a write will only occur when the user clicks one of the write buttons as detailed in the previous sections.
- **Immediate** – In this mode a write will be performed any time a register value is changed using any of the methods detailed in the previous sections. This is not the default mode as it should be used with caution. Because the write occurs any time the register value is changed it is possible to put the register in to an undesired state when clicking on bits to change their value. It is recommended in this mode that the user use the control groups or the value field to set the entire value of the register or register bit groups if interim values are not desired.

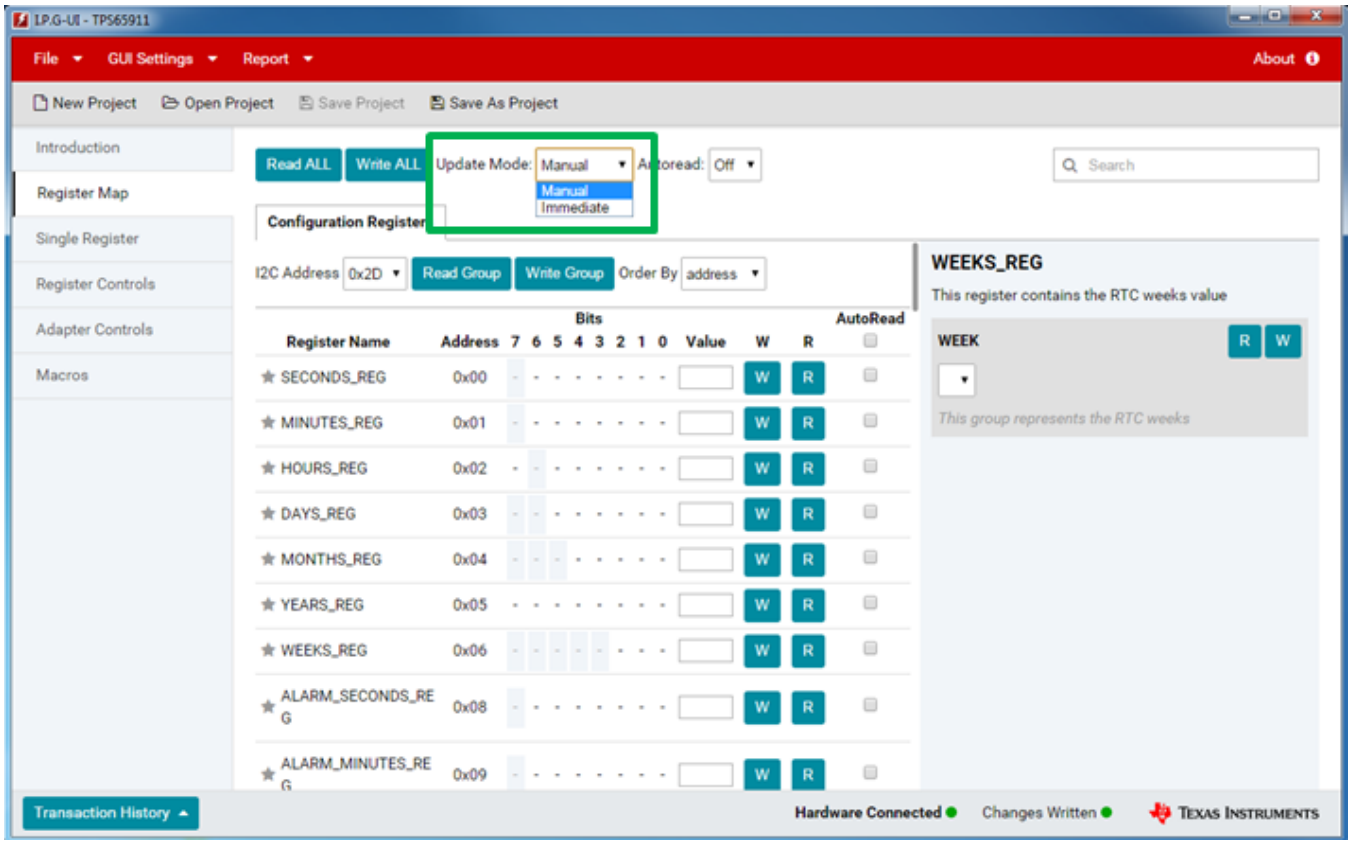

Figure 22. Update Mode Control

10.10 Autoread

It is possible to set registers to be automatically read at a given interval. This is useful for operations such as polling status registers of the device. There are two parts to this control in the register map.

1. To the right of each register in the register map there is a checkbox in the AutoRead column. Checking this box will mark the register to be read automatically when the polling interval is set as described in the next item. It is also possible to mark all registers to be automatically read by selecting the checkbox in the table header.
2. In the upper area of the register map view there is a drop-down with the label Autoread. This drop-down controls the polling interval for the autoread functionality. When the interval is set to any other than Off all registers that have the autoread checkbox marked will be read after each polling interval.

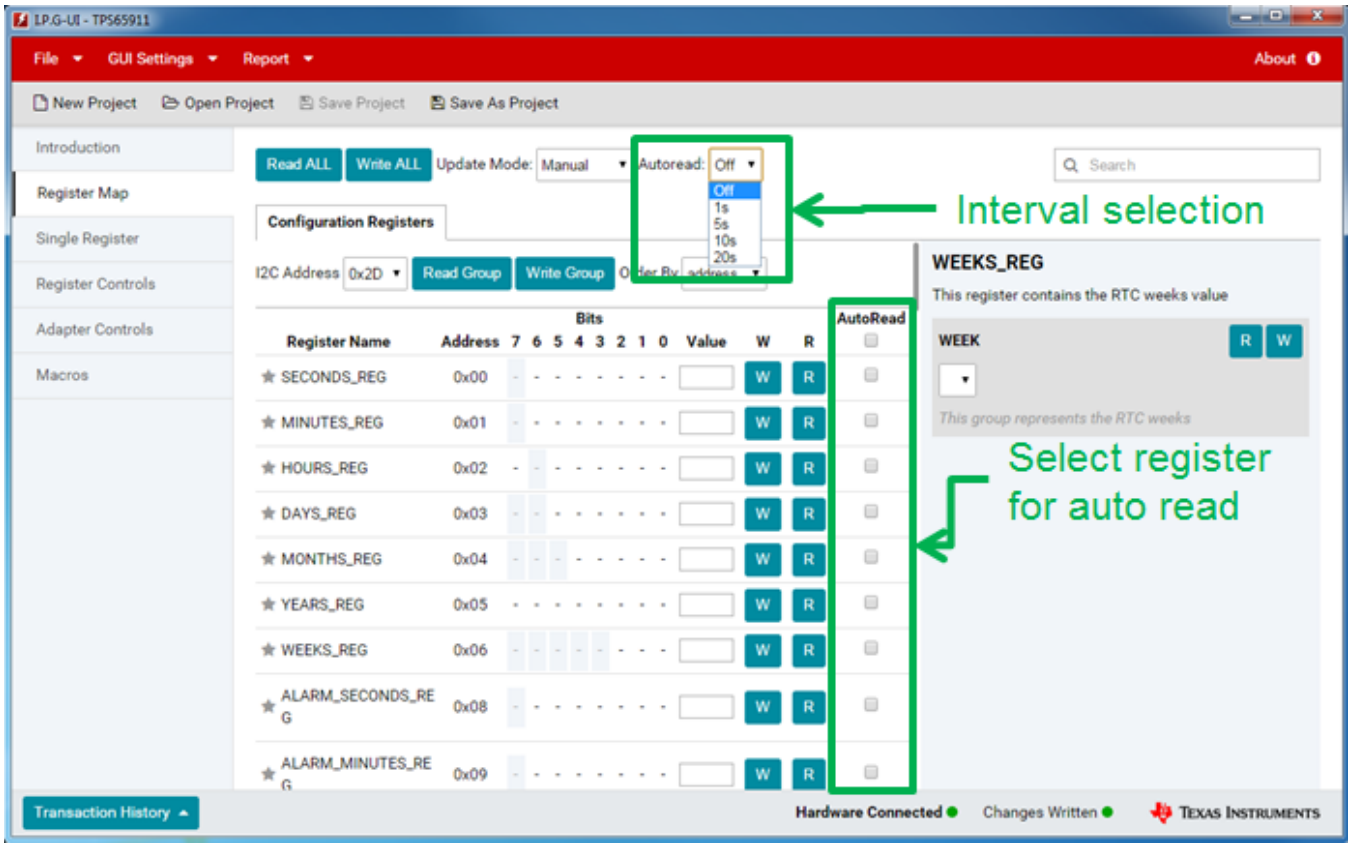

Figure 23. AutoRead Control

To stop automatically reading registers either return the polling interval to Off or uncheck the register's autoread checkbox.

11 Single Register

The single register view allows to user to focus on an individual register. It also enables some additional register controls that will be detailed in this section.

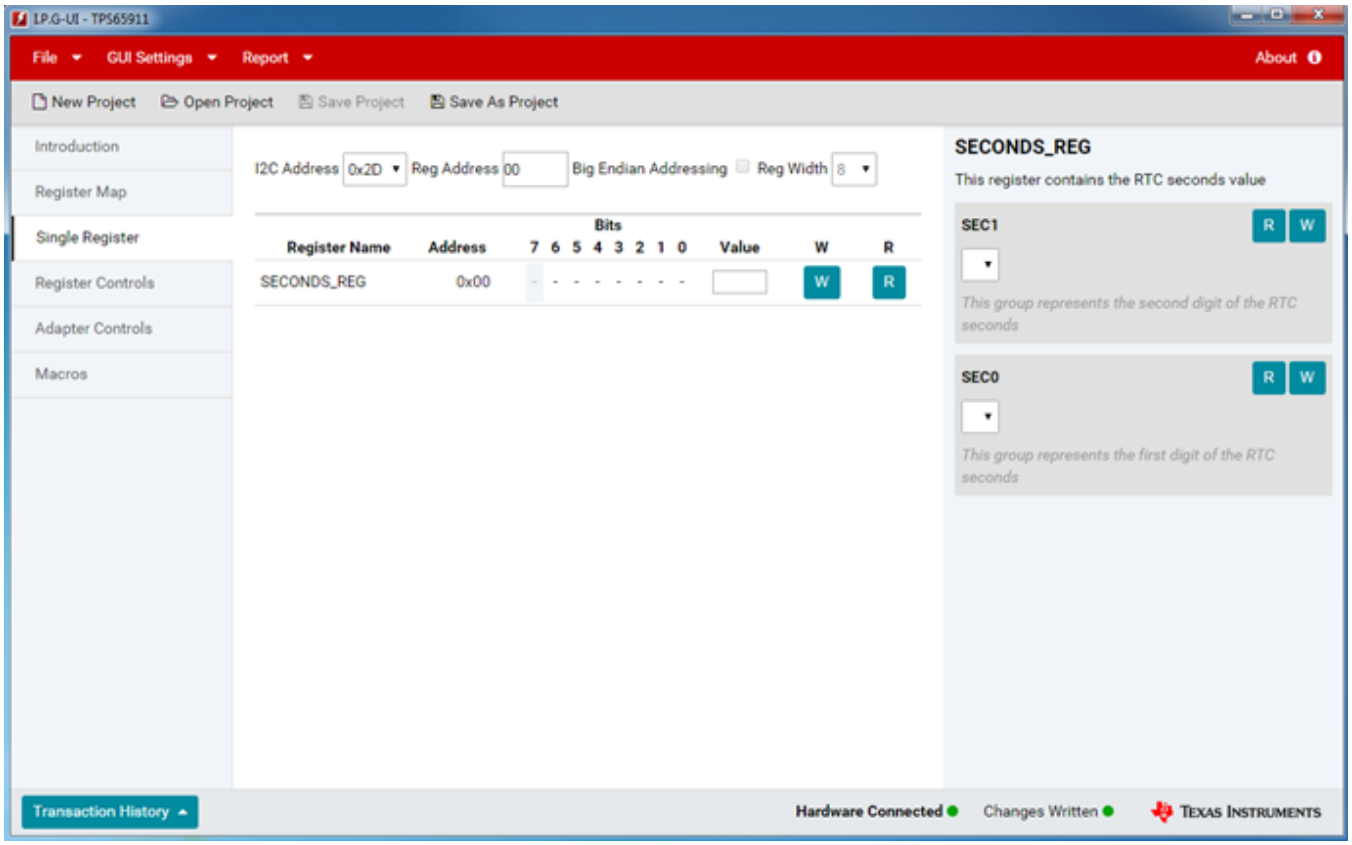

Figure 24. Single Register View

11.1 Register Mini-Map

The register mini-map shows a consistent look and feel to match with the register map view for each register. The controls are the same for this mini-map except that autoread and favorite selection are disabled. The register description is also present on the right side to be consistent with the register map view.

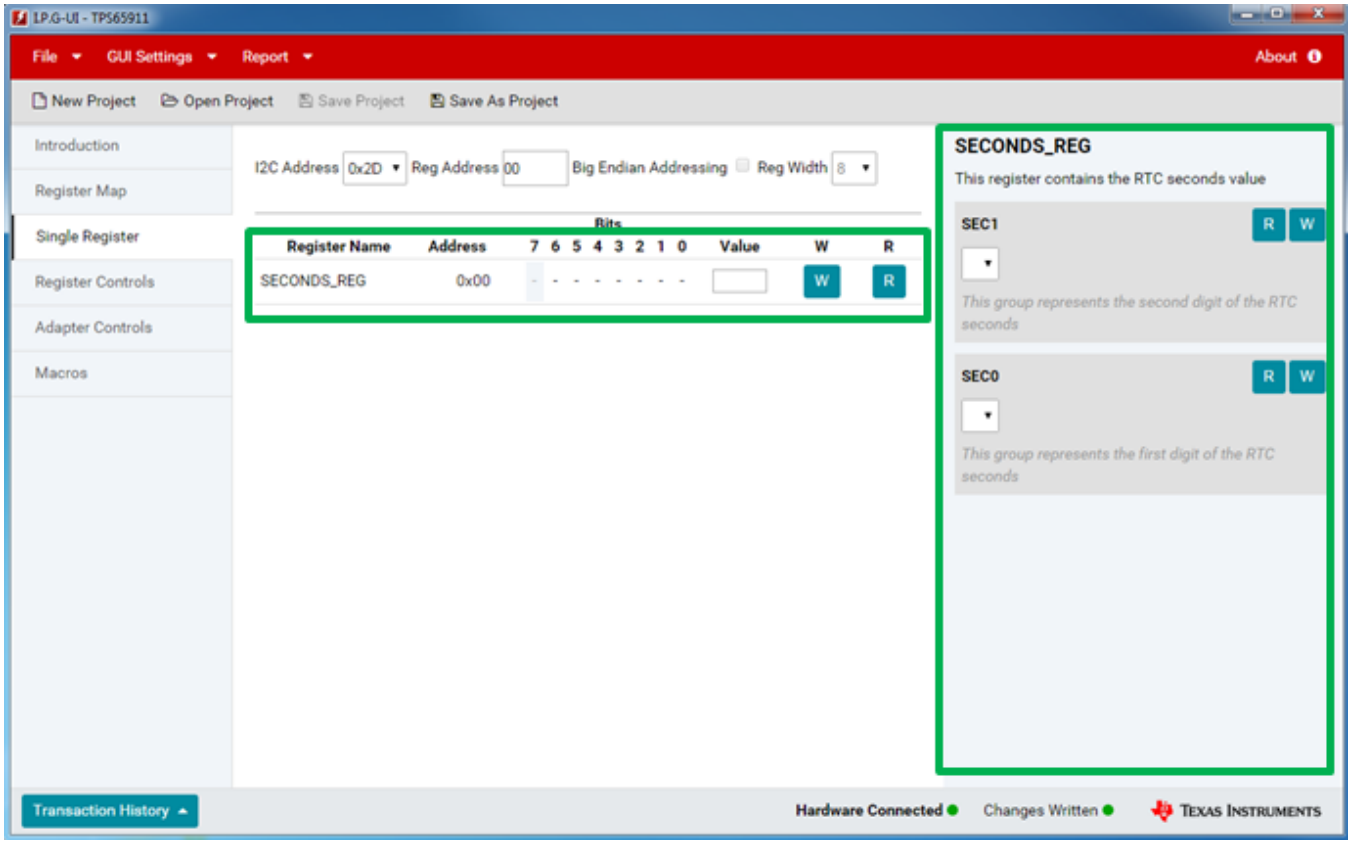

Figure 25. Single Register Mini-Map

11.2 Setting the Register Address

When the **Reg Address** and/or **I²C Address** field is updated the single register view will be updated to show one of the following options:

- **Known Register** – If the register address is a known register for the selected I²C Slave address then the register mini-map will show that register and register description, including any changes that were made in the register map view. For known registers the **Big Endian Addressing** and **Reg Width** value cannot be changed.
- **Unknown Register** – If the register address is an unknown register for the selected I²C Slave address then a generic register mini-map will be created. This allows working with registers that may not have been defined yet. Additionally it is possible for unknown registers to change the endianness of the register address which is used for registers with addresses greater than 8 bits as well as to adjust the register width.

11.3 Big Endian Addressing

Big Endian Addressing is used for registers with an address greater than 8 bits (for example 16-bit addressing). This allows the user to set the endianness of the address so that they can enter the register address as normal, that is, 0x0001, and the software will take care of sending this over the I²C bus as 0x0100.

11.4 Reg Width

The **Reg Width** control can be used to change the number of bits in any unknown register. This allows programming registers that are wider than 8 bits.

11.5 Generic I²C Access

The **Single Register** view can be used for generic I²C access in the following manner. When creating a new project from a device file on the landing page the user can select the Blank-I2C-7bit device to create the project. Because this device has no known registers all registers on any I²C slave address can be configured for endianness and register width. The user can then use the read and write buttons to read and write the register.

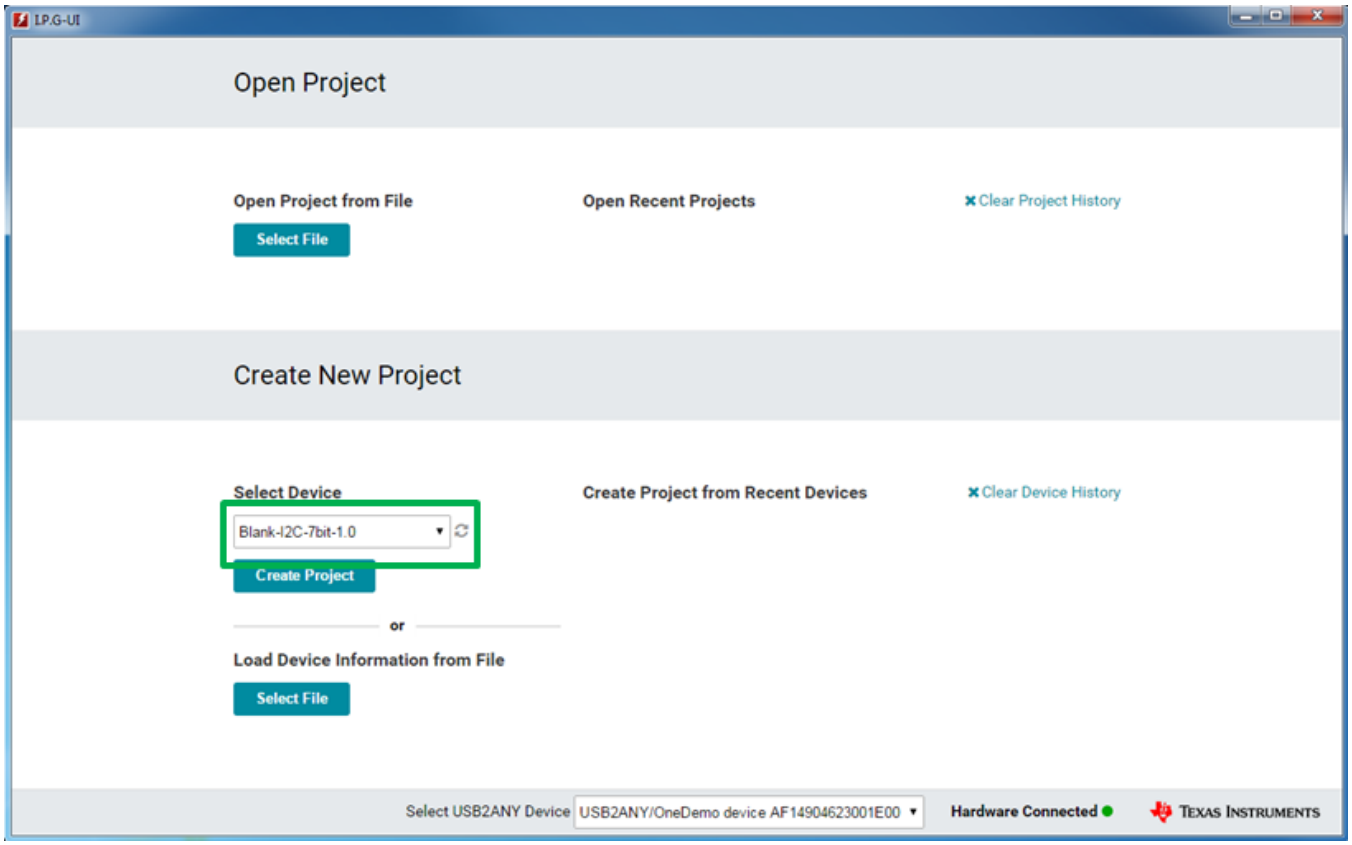

Figure 26. Generic I²C Access

12 Register Controls

The **Register Controls** view enables the user to view all of the control groups sorted by category. This gives the user the ability to access the human-readable controls without having to select between registers on the register map or search the data sheet to find out what registers contain a particular category of controls.

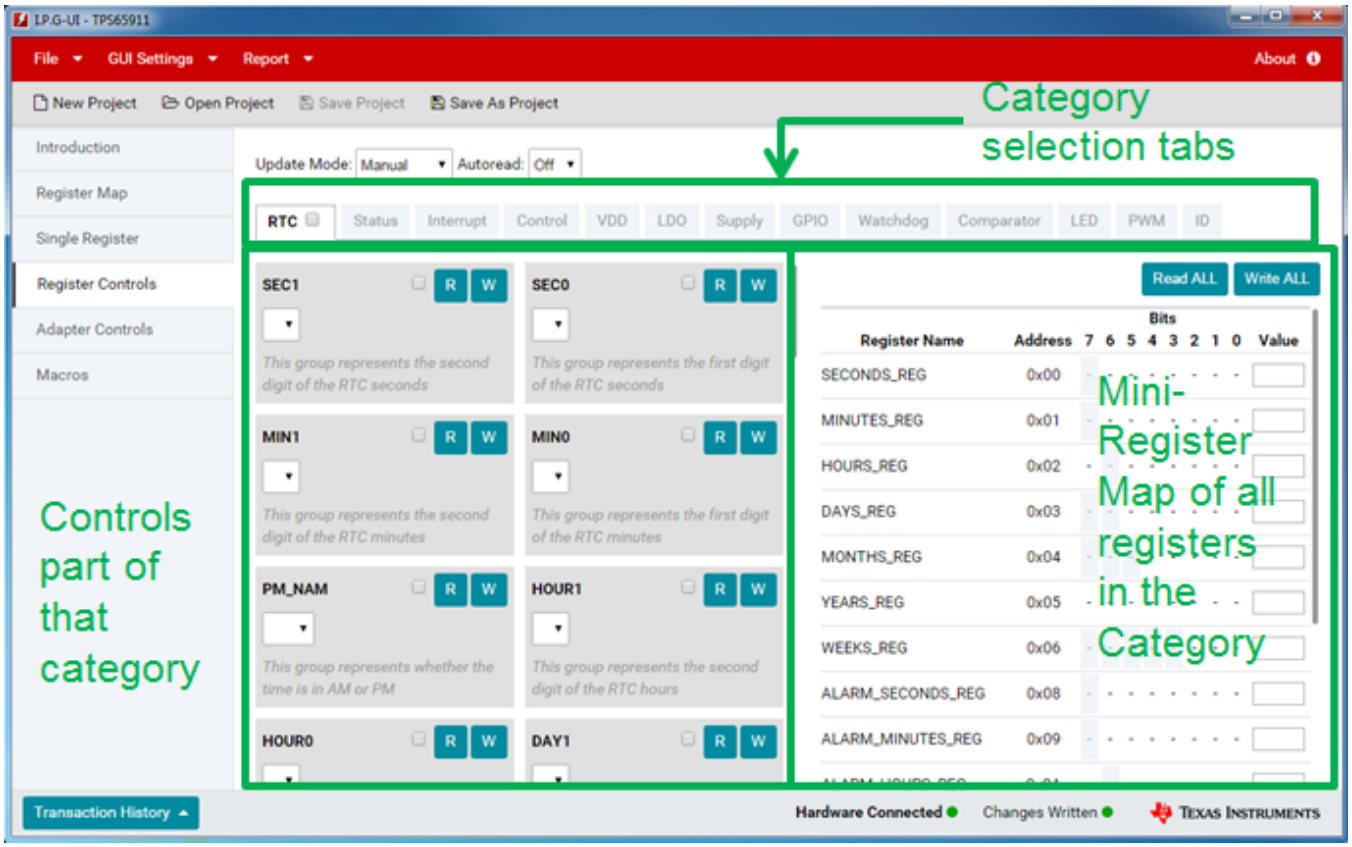

Figure 27. Register Controls View

Selecting a category tab at the top of the view will bring up all of the control groups that are part of that category. Interacting with the controls is the same as using the control groups in the register description as covered previously. All changes made in the register controls view will be visible in the register map view.

12.1 Register Controls Auto Read

It is possible to control the auto read settings from the Register Controls view as well. The checkboxes found inside of the register controls tabs and groups will enable the auto read for the various groups and all the groups within a category.

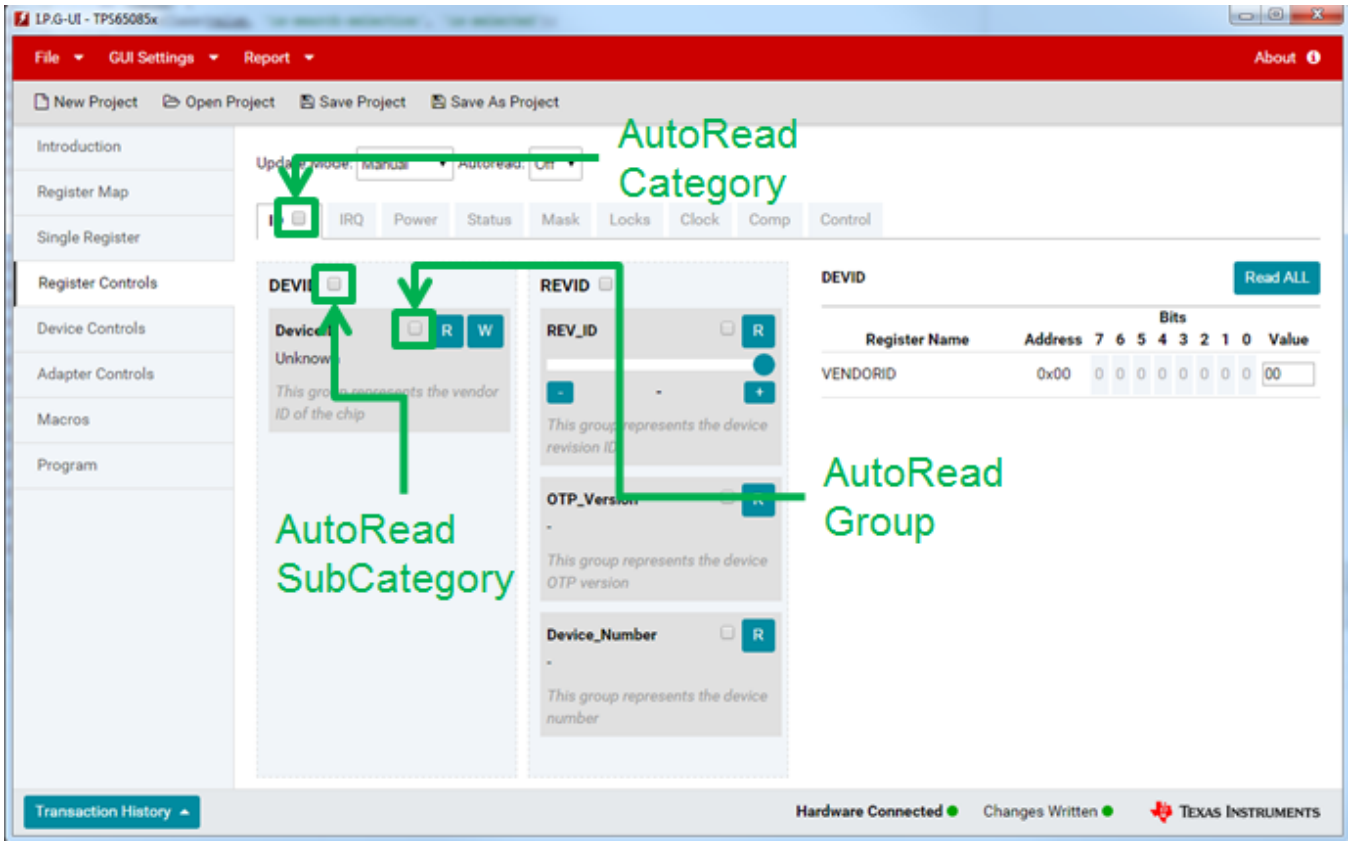

Figure 28. Register Controls Auto Read

NOTE: The read operation is done at a register level so selecting auto read for a group will also select auto read for all other groups contained in that register.

13 Device Controls

The device controls view allows the user to perform actions that impact the device/EVM and which are not tied directly to a single register. Example of this may be to run a sequence of commands to toggle a bit value or to set whether automatic password generation should be done. This view will only be shown if Device Controls are defined for the selected device.

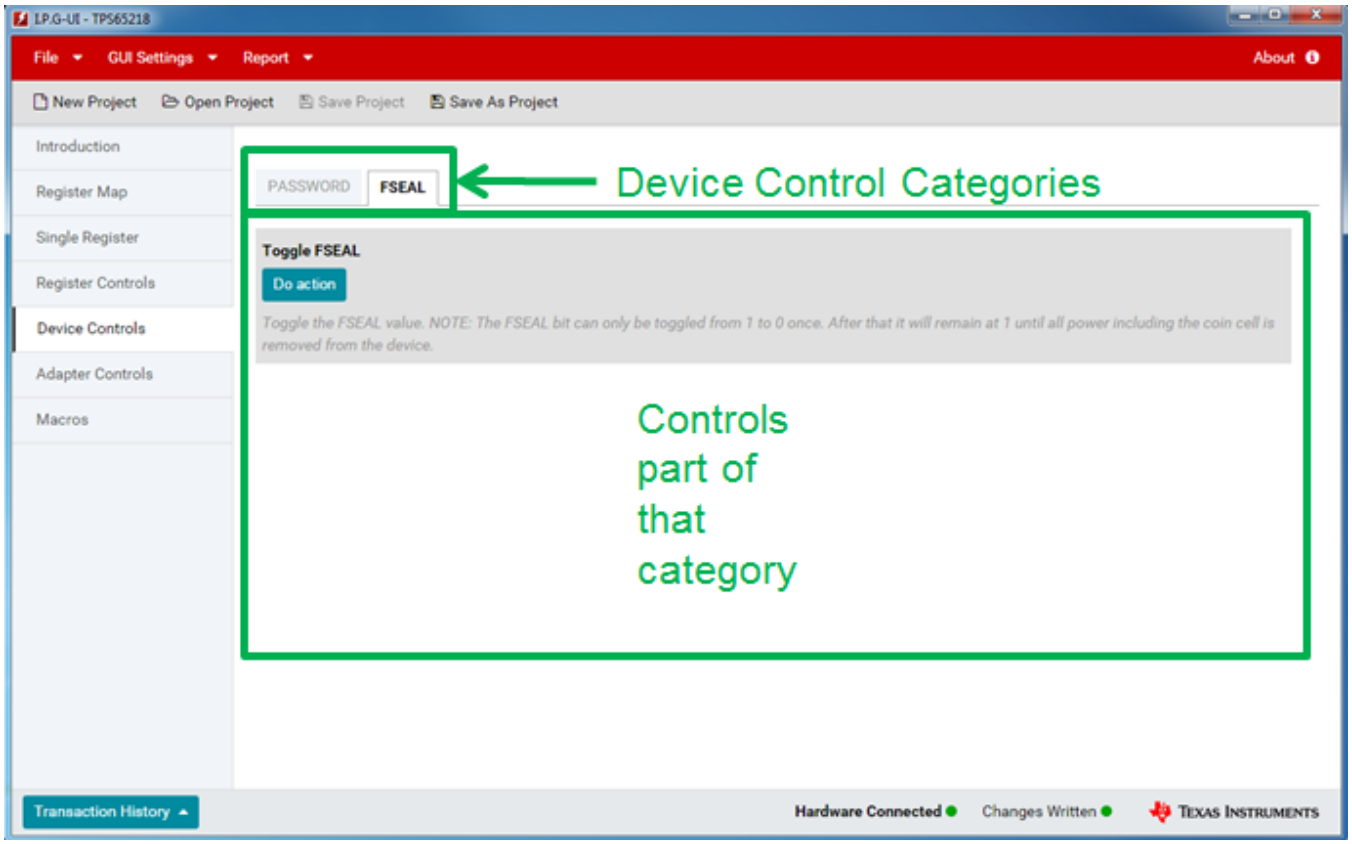

Figure 29. Device Controls

14 Macros

The Macros view allows the user to run any of the pre-defined macros for the device (if any) or to create new macros from the transaction history to save commonly-repeated command sequences. The options available are:

1. Run the macro to execute the sequence of commands
2. Edit/View the macro to adjust what commands are being run. Edit is only available for user-created macros. Pre-defined macros only allow viewing the commands.
3. Clone the macro to give a starting point for defining a new macro
4. Delete the macro. This is only available for user-defined macros. Pre-defined macros cannot be deleted.

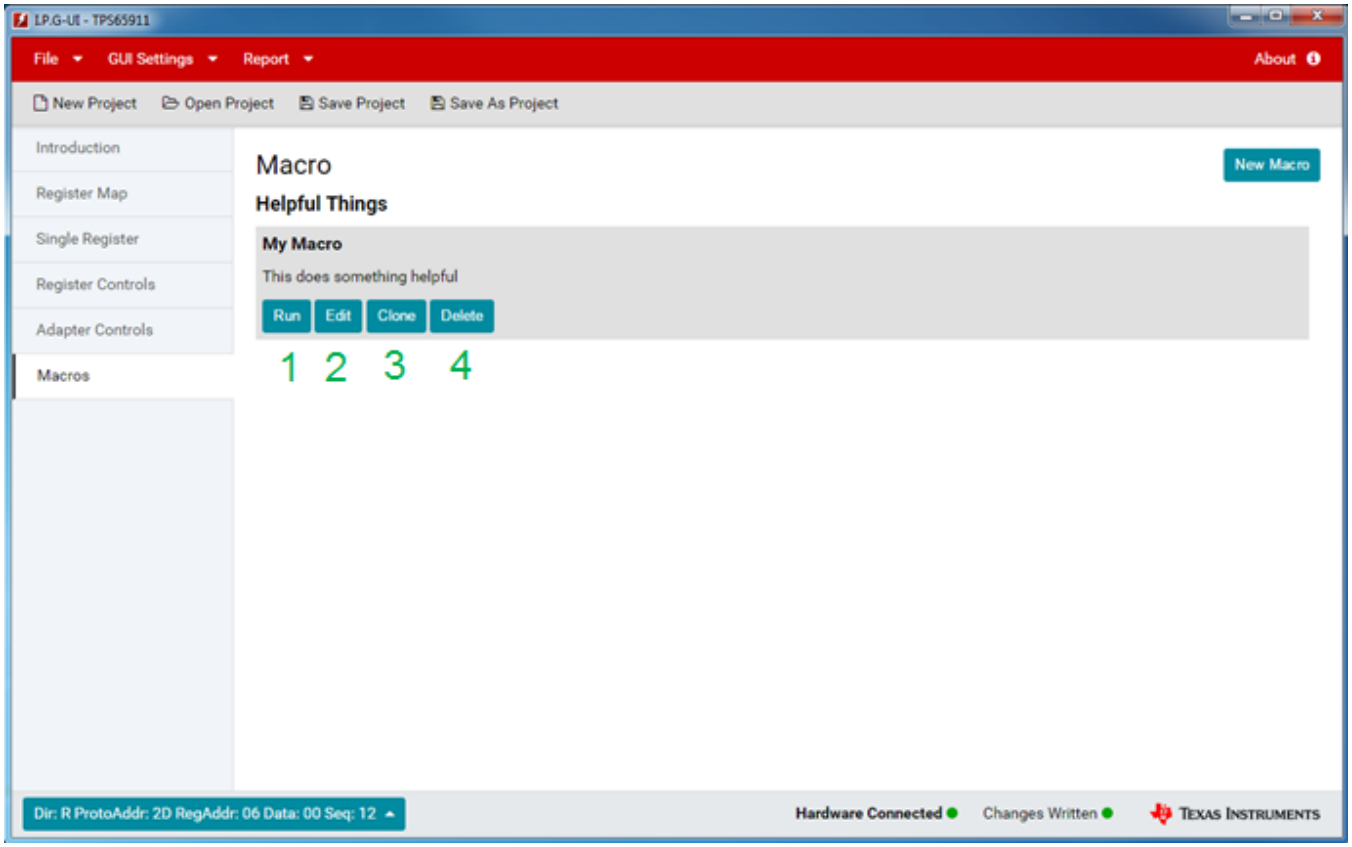

Figure 30. Macros View

14.1 Creating a New Macro

Clicking the “New Macro” button takes the user to the macro creation screen. Here the user has the ability to:

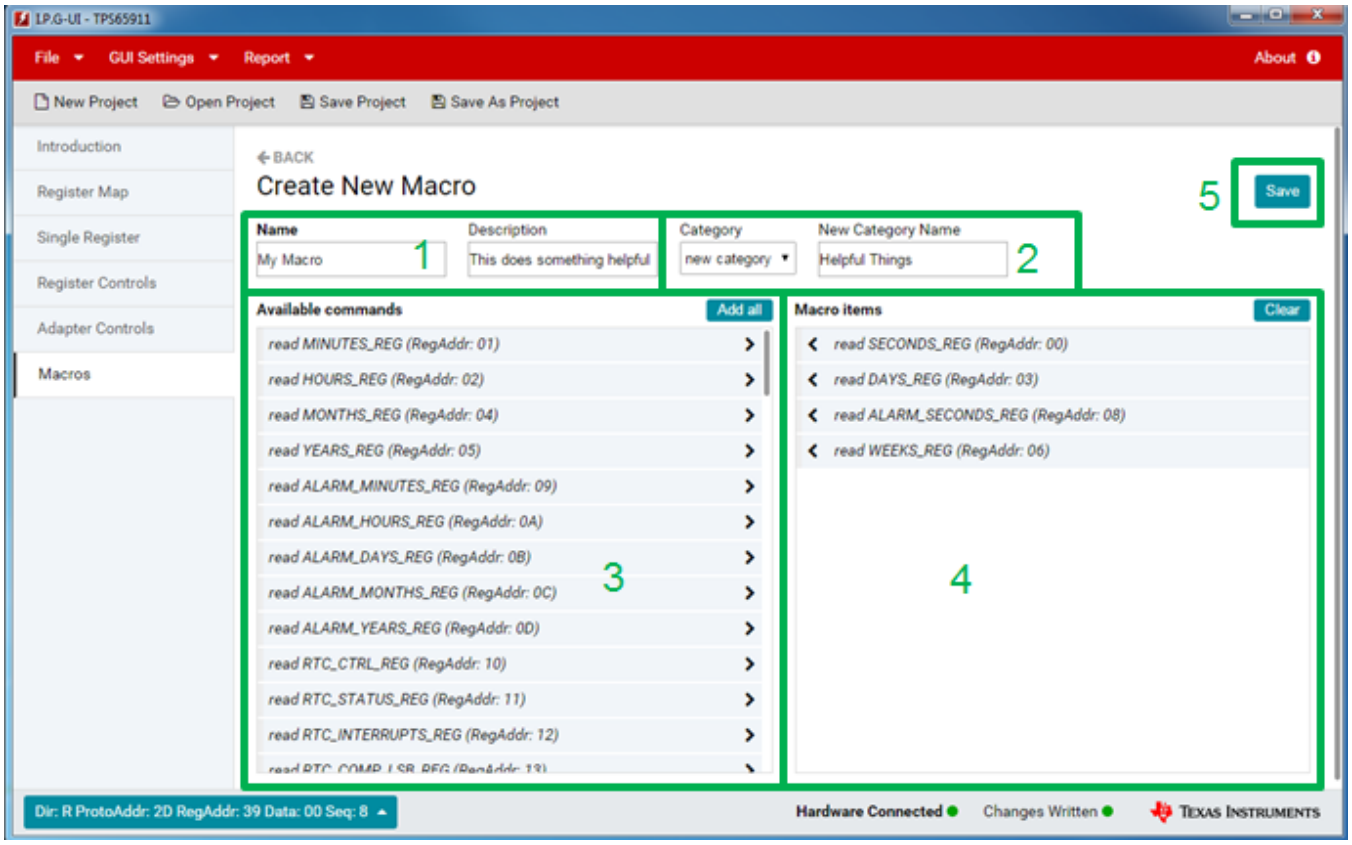

Figure 31. New Macro Creation Screen

1. Assign a name and description to the macro
2. Choose which category to add the macro to or create a new category
3. Select items from the transaction history to be added to the macro
4. Determine the order of the commands to run in the macro by dragging and dropping them into the desired order
5. Save the macro to be run again later

The back button can be used to return to the Macros view without saving the macro.

NOTE: Not all items in the transaction history are available to be added to macros. When a transaction item is created by the GUI automatically, this item will not be selectable for macros. An example of this is if automatically generating the password requires writing to a password register before writing the register the user selected, then the write of the user-selected register will be available for the macro, but the auto-generated command to write the password register will not. This is done on purpose as selecting the write of the register will regenerate the auto write command so there is no need for the user to add that to the macro directly.

14.2 Transaction History

The Transaction History lists all of the commands sent to and from the device as well as any device control actions that have been run. The commands are grouped together to show which were performed as part of a single operation. Errors are also displayed in the transaction history. Clicking on the transaction history will expand it so that the user may scroll through the history of commands and errors.

With the addition of advanced controls, it is possible for pre and post operations to be performed automatically for the user. To differentiate these from normal commands, they are tabbed over to the right. Items that are tabbed to the right below the primary command are run before the primary command. Items that are tabbed to the right above the primary command are run after the primary command. These items will not be selectable in the Macros view as they will be run automatically when the user selects the primary command for the macro.

15 Adapter Controls

The **Adapter Controls** allow the user to change the USB2ANY adapter configuration for their device. The default values are assigned by the device description but can be overridden by the adapter controls. However, the default values are set to work for the EVM from TI and should not need modification. The following controls are available to the user:

- **I²C Address** – This control allows the user to select whether to use 7-bit or 10-bit I²C slave addressing. For example when using the single register view with the blank device as covered previously this would allow the user to communicate with devices that use a 10-bit I²C slave address rather than the default 7-bit addressing.
- **3.3V pullup** – This enables the 3.3-V pull-ups on the USB2ANY adapter for devices that do not have a pull-up already on the board.
- **5V pullup** – This enables the 5-V pull-ups on the USB2ANY for devices that do not have a pull-up already on the board.
- **I²C Speed** – This allows the user to change the I²C bus speed for the device. The available speeds are 10, 100, and 400 kbps and are determined by the adapter hardware, not by the PMIC.
- **Protocol** – This allows the user to choose the I²C protocol to use. Some devices do not support the repeated start protocol for operations like a register read. Instead they require a stop/start protocol to be used.
- **Transfer Mode** – This allows the user to specify whether block mode transfers should be used when communicating with the device. When **Block** mode transfers are used the IPG-UI will consolidate contiguous registers into a single read or write (up to the USB2ANY limit) and transmit them with a single command. In **Single** mode each register is read or written one at a time. Not all devices support block mode and single mode must be used.

16 Status Bar

The status bar is located at the bottom of the IPG-UI and provides the user with the information about GUI operations

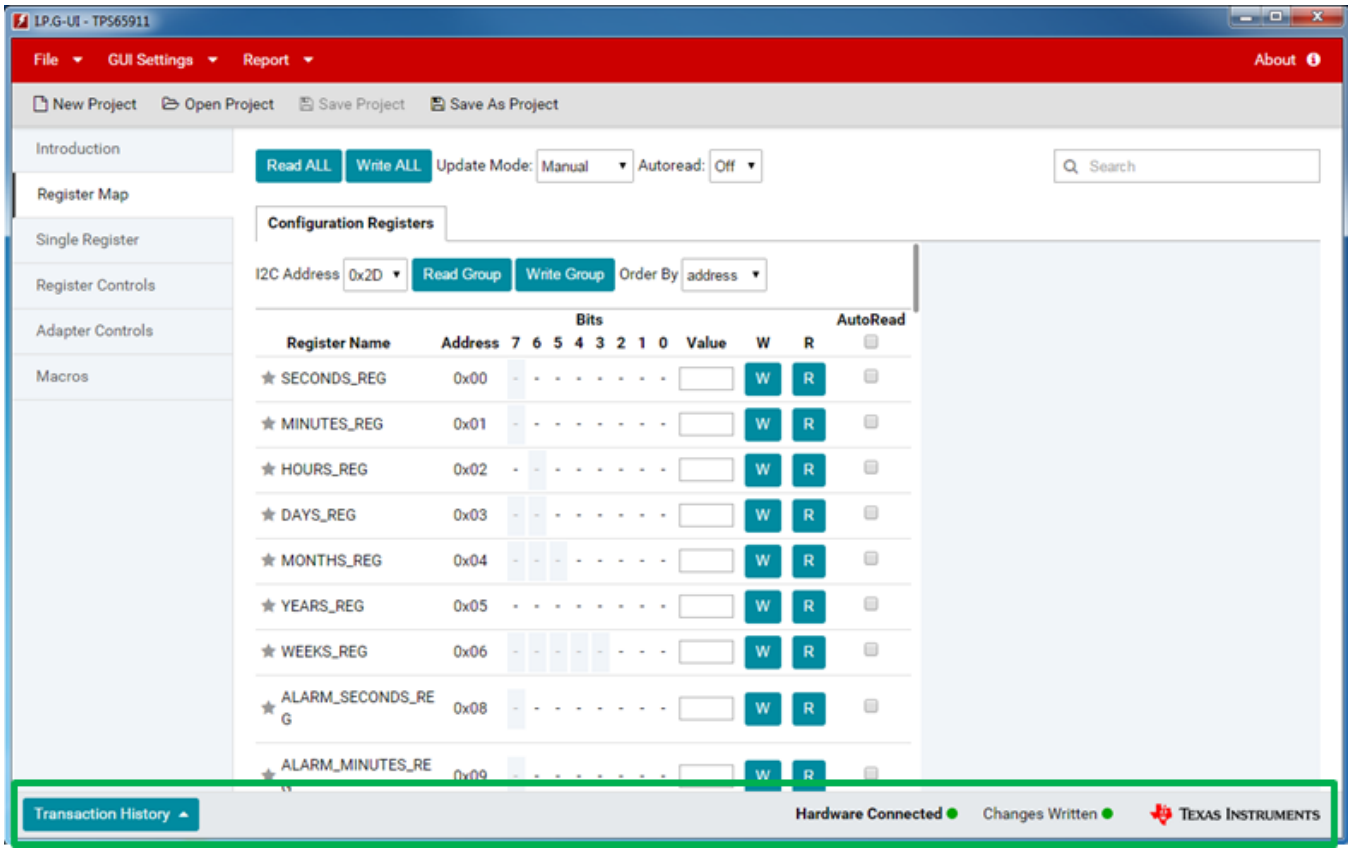

Figure 32. Status Bar

16.1 Transaction History

The **Transaction History** lists all of the commands sent to and from the device. The commands are broken out by register even when the device is in block mode. Errors are also displayed in the transaction history. Clicking on the transaction history will expand it so that the user may scroll through the history of the commands and errors.

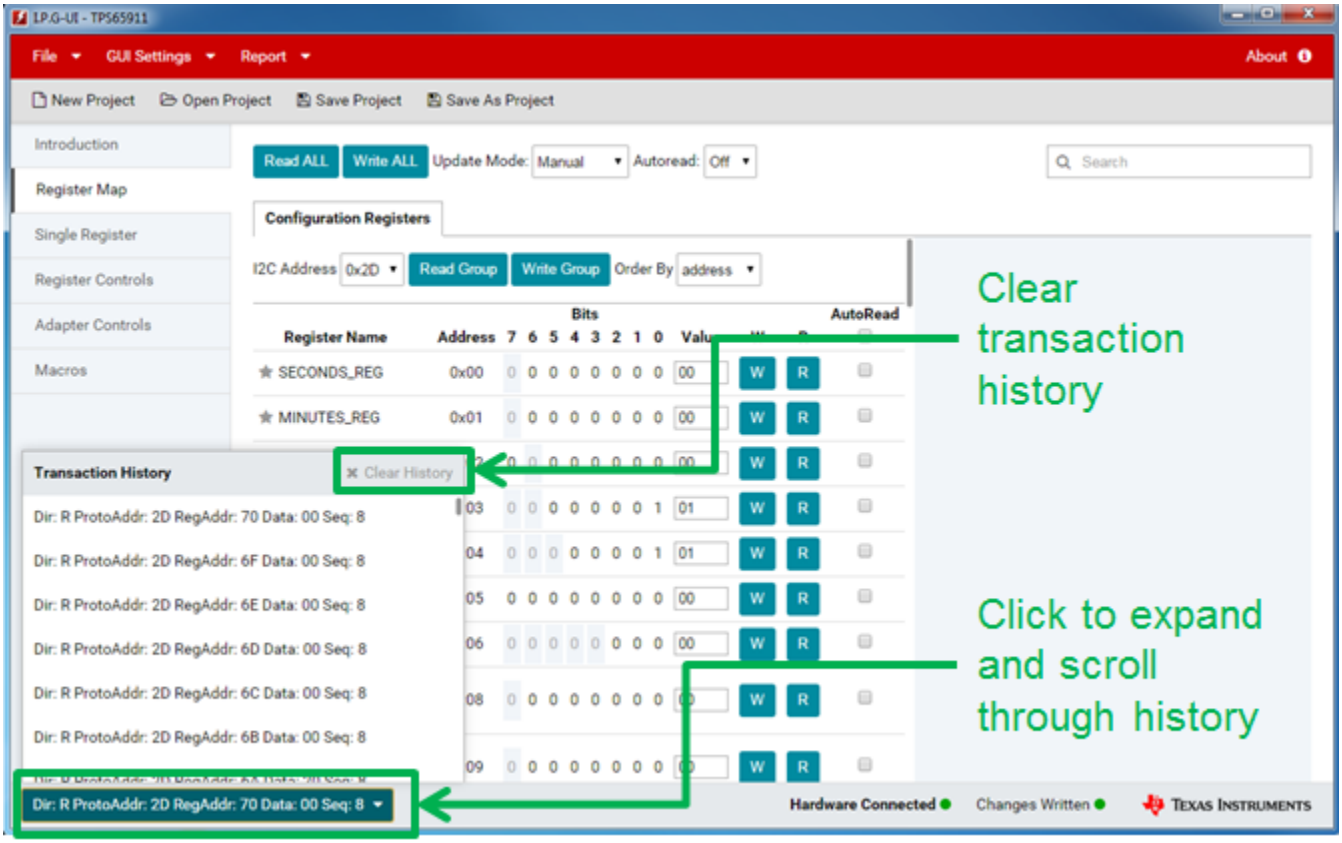

Figure 33. Transaction History

The user may also clear the transaction history when it is expanded using the Clear History button.

Read and Write operations have the following format in the transaction history:

- **Dir** – This indicates if the command was a read (R) or a write (W)
- **ProtoAddr** - This represents the I^2C slave address for the command
- **RegAddr** – This represents the register address for the command
- **Data** – This represents the data read from or written to the register
- **Seq** – This is the command sequence which indicates the order in which the commands were sent

Errors listed in the transaction history will provide the error code returned by the USB2ANY adapter and what the code represents.

16.2 Hardware Connected

The **Hardware Connected** item on the status bar indicates whether the USB2ANY adapter is connected. A green circle indicates the adapter is connected while a red circle indicates it is not connected. It should be noted that this does not indicate whether the EVM is connected to the USB2ANY. If an EVM is not connected then the read and write commands will throw an error.

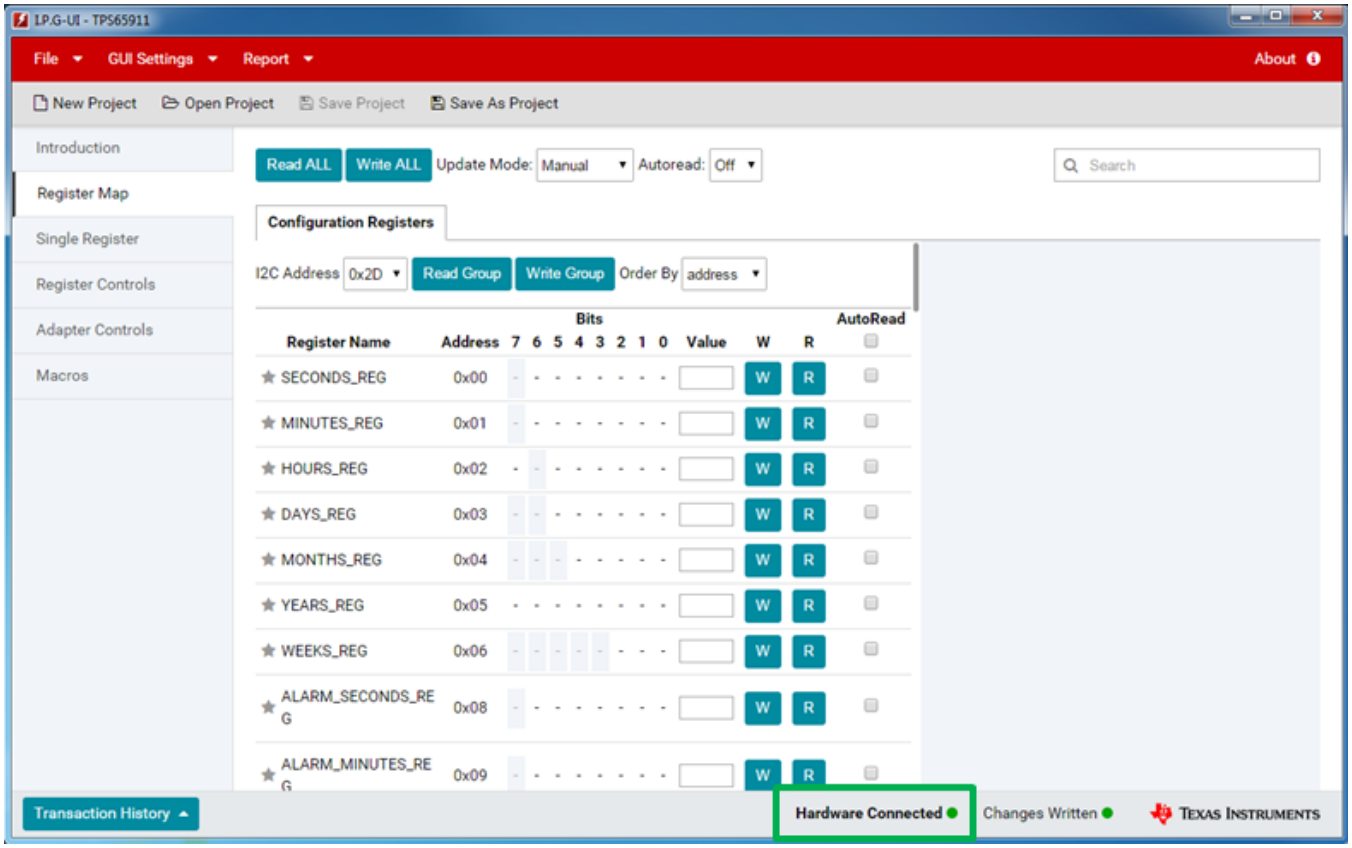

Figure 34. Hardware Connected Status

16.3 Changes Written

The **Changes Written** item on the status bar indicates whether the user has made changes to the register settings that have not been written to the device. A green circle indicates that all changes have been written while a red circle indicates that the user has not written the changes. This is slightly different from the bit changed value in the register map in that when the registers are first read the bits are shown as changed to let the user know which bits were set to 1 instead of 0, but the **Changes Written** status will not show any unwritten changes because the user has not made any changes to the bits in the register.

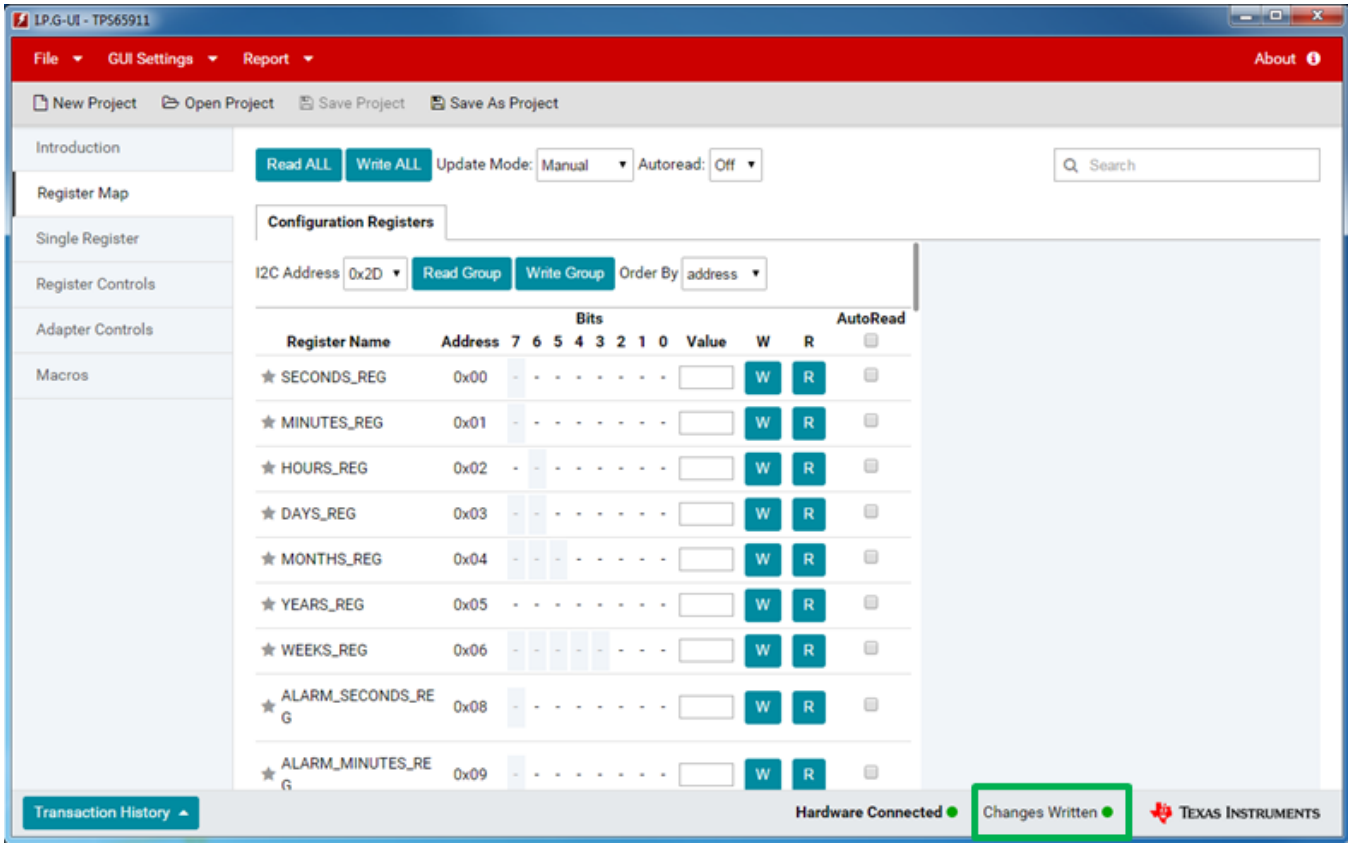

Figure 35. Changes Written Status

17 Quick Access Controls

The **Quick Access Controls** can be found in the gray bar along the top of the GUI. These provide the user a simple one-click method for common GUI actions.

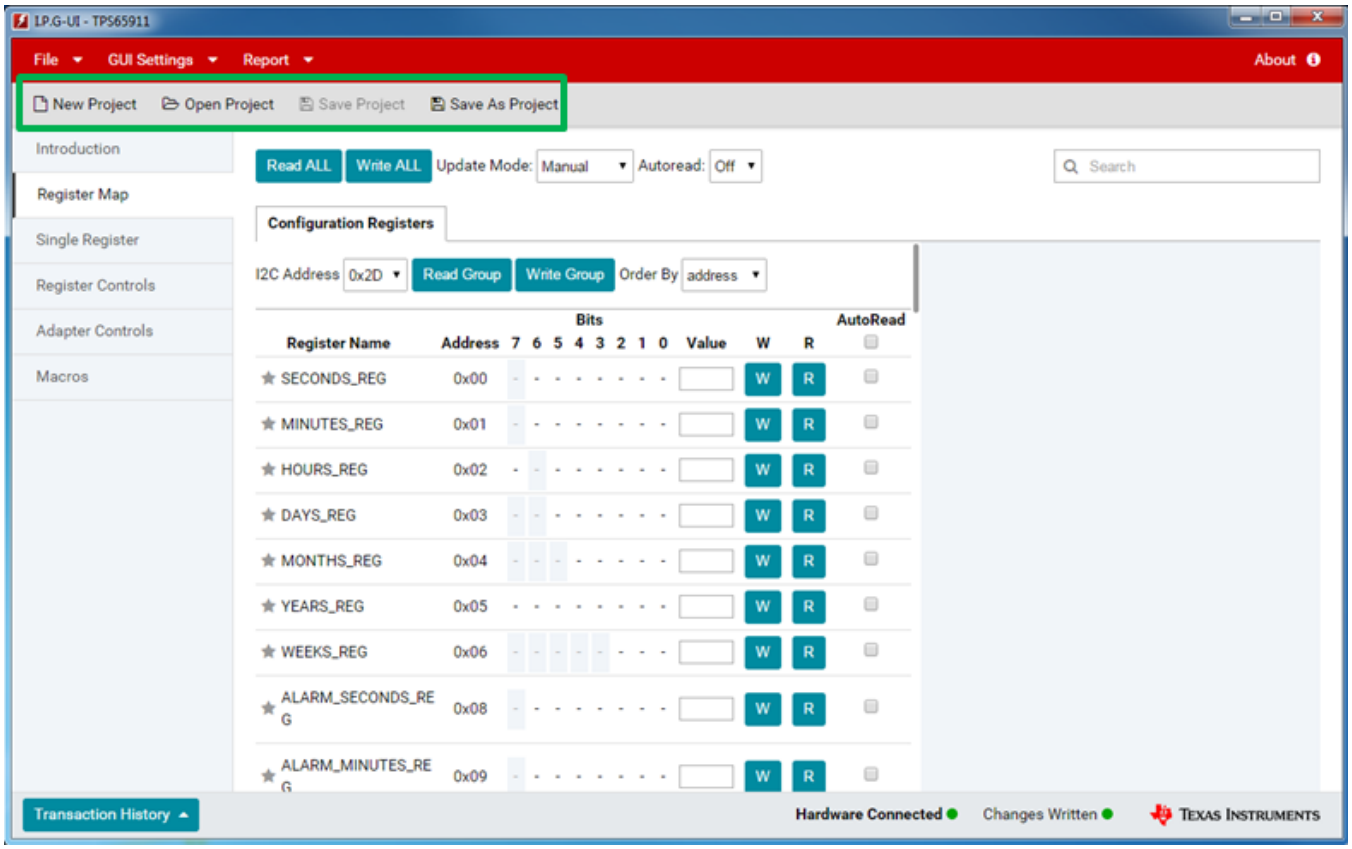

Figure 36. Quick Access Controls

The controls available are:

- **New Project**—Clicking on **New Project** will take the user back to the Landing Page to create a new project. If the user has made changes in the current project they will be prompted to either save the project, discard the project, or cancel the new project request.
- **Open Project**—Clicking on **Open Project** will open a file-browsing dialog where the user can select a previously saved project file to open. The user is not prompted to save the current project or discard changes when this is selected.
- **Save Project**—Clicking on **Save Project** will save the current project to the already selected project name. If the project has not already been saved using the Save As Project functionality then the Save Project option will not be clickable.
- **Save As Project**—Clicking on **Save As Project** will open a file-browsing dialog to let the user select a path to save the project to as well as the project file name. This path and name will be used by **Save Project** to save any updates to the already selected path and name. The default project save path is the <user's home directory>/Documents/Texas Instruments/IPG-UI/Projects

18 File Menu

The **File Menu** can be found in the red bar at the top of the GUI. Hovering over the File item will expand the menu and provide the user items they can click on to perform the listed action. Currently the file menu shows the same options as the quick access menu.

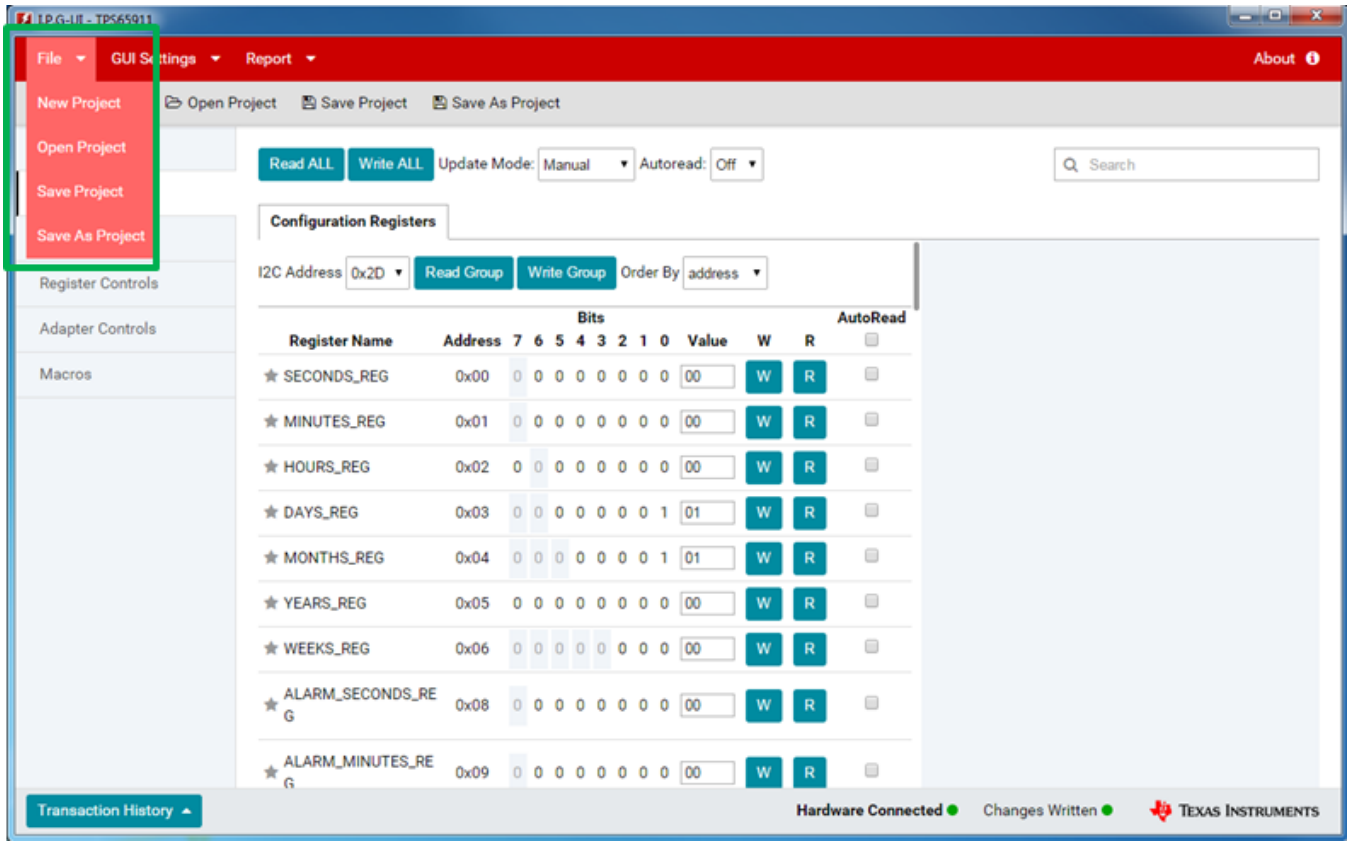

Figure 37. File Menu

19 GUI Settings

The GUI settings menu allows control over GUI level behavior. The two items available are Confirm Write and Autosave Project. A check next to the item indicates that the option is set and clicking the item will toggle the option.

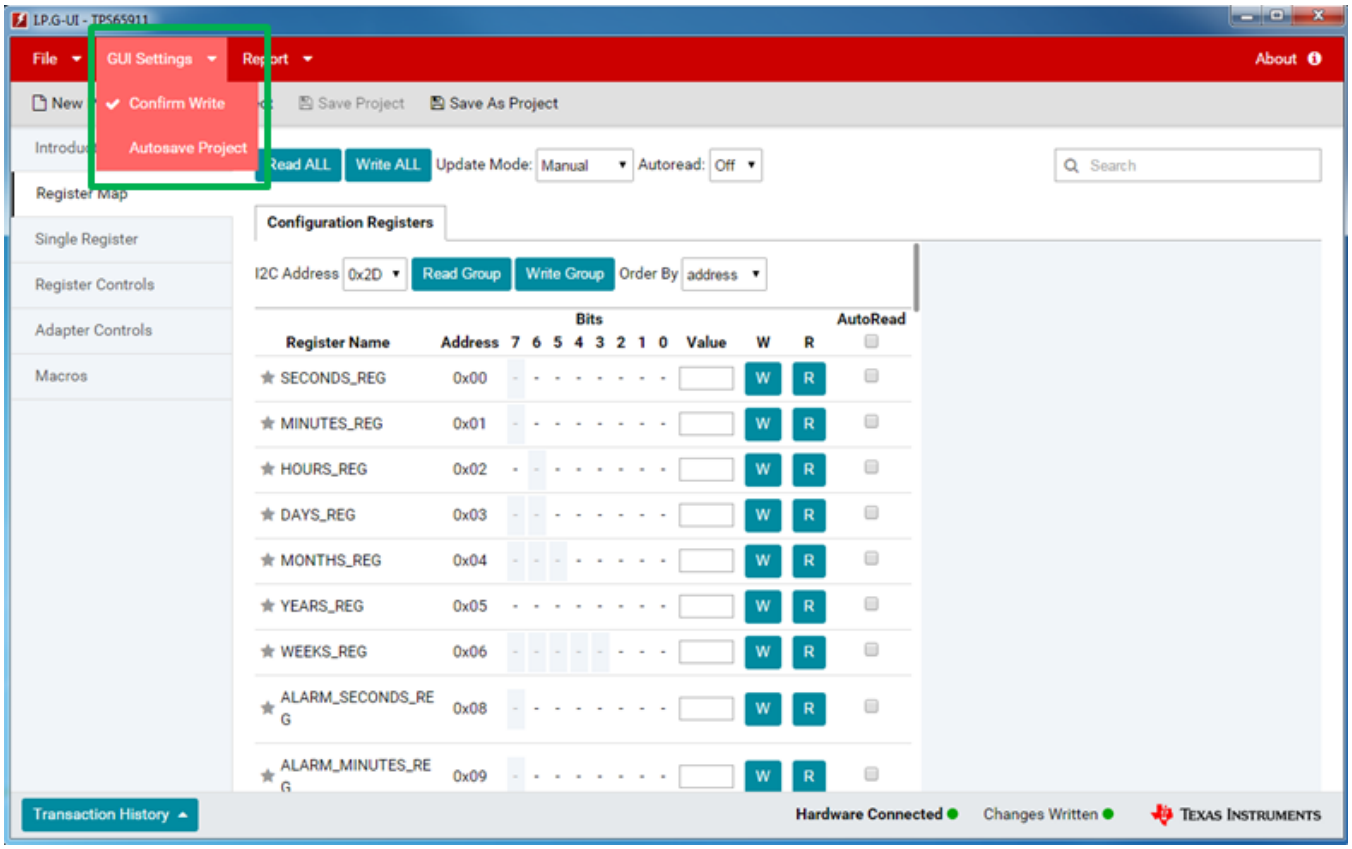

Figure 38. GUI Settings

19.1 Confirm Write

When this option is set to on and the user attempts to write an unknown value to a register they will be prompted with a notification of which register(s) and which logical group(s) have unknown values. At this point the user can determine whether they want to continue with the write or exit and fix the issues. Turning this control off will avoid this notification and the values will be written.

19.2 Autosave Project

When the autosave option is checked any time the user makes a change to the project settings or register configurations those changes will be saved. If the user has not previously saved the project then on the first change they will be asked to provide a file name to save the project to.

20 Report

The Report menu item allows the user to generate reports for sharing with TI. The current option supported is to create a Transaction History report. Clicking this option will prompt the user to specify a file name to save the report to which they can send to TI or anyone else for review.

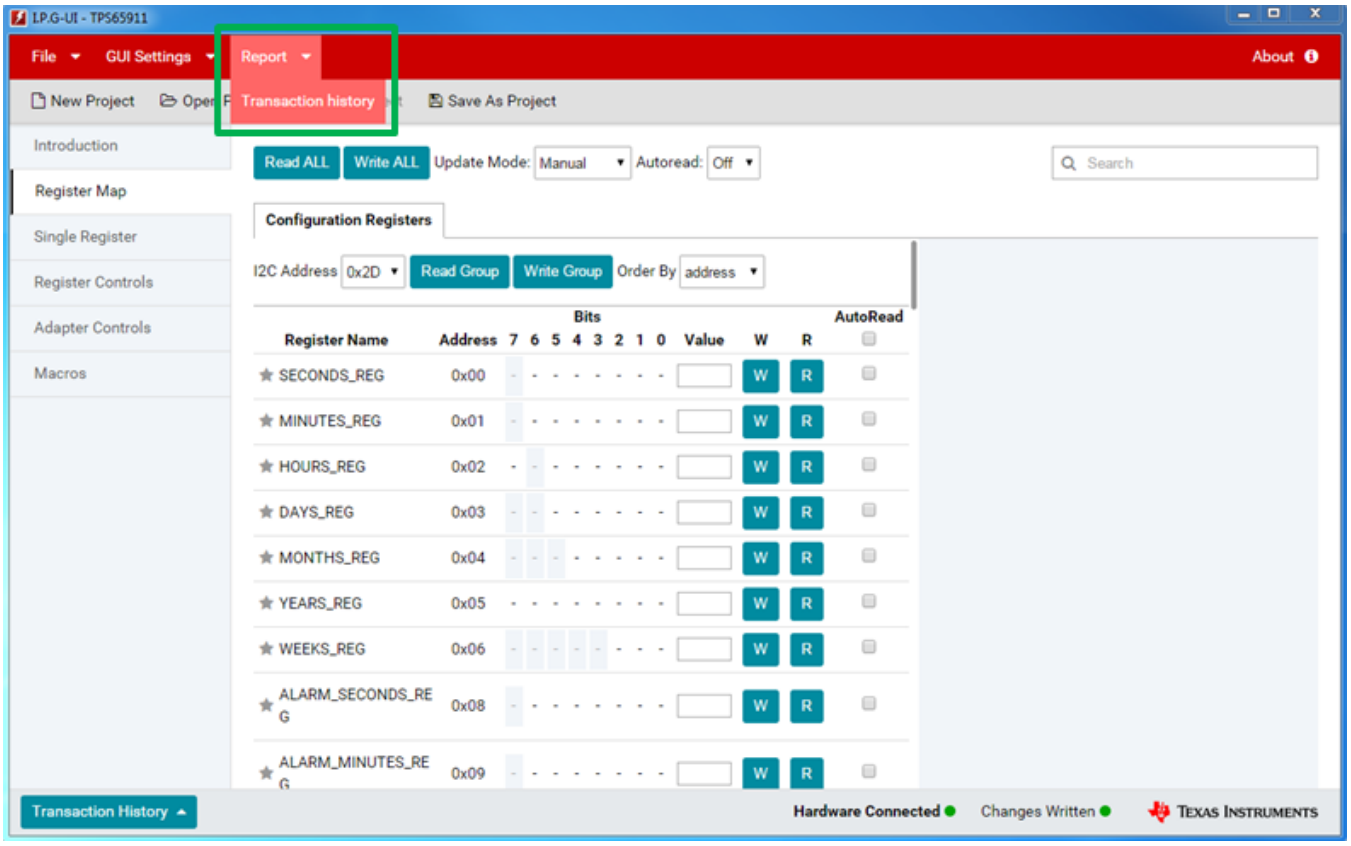

Figure 39. Report

21 Errors and Notifications

The user is notified of Errors and other status notifications in the upper right part of the GUI. Errors are displayed as a red block with the error details. When an error occurs it will remain until the user clicks the close icon to dismiss it. If more than one error of the same type occurs only one notification will be shown with a count of the number of times the error occurred.

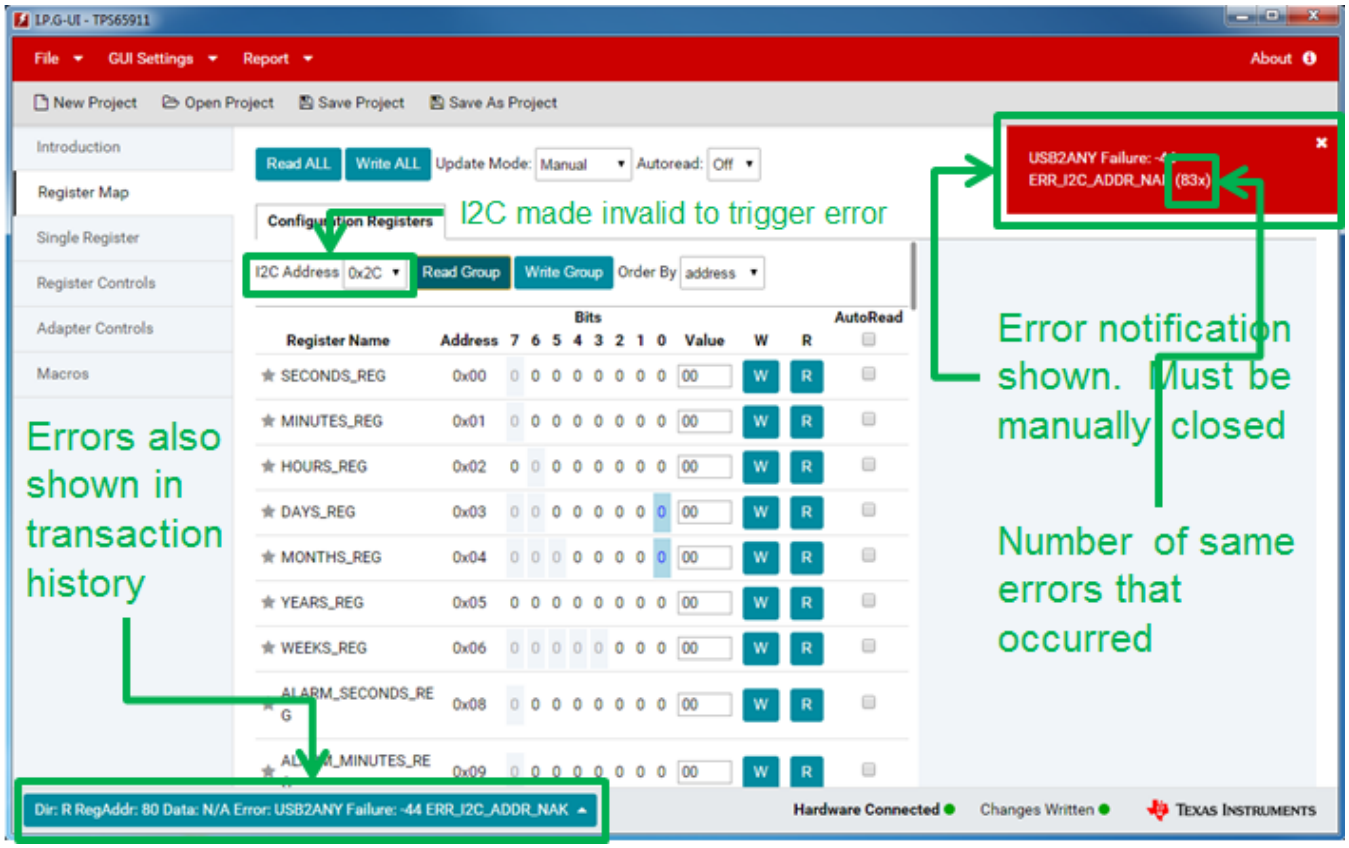

Figure 40. Error Notifications

Notifications are shown as a blue box in the same area. They behave the same as errors with the difference that a notification will automatically disappear after 5 seconds if the user does not dismiss it.

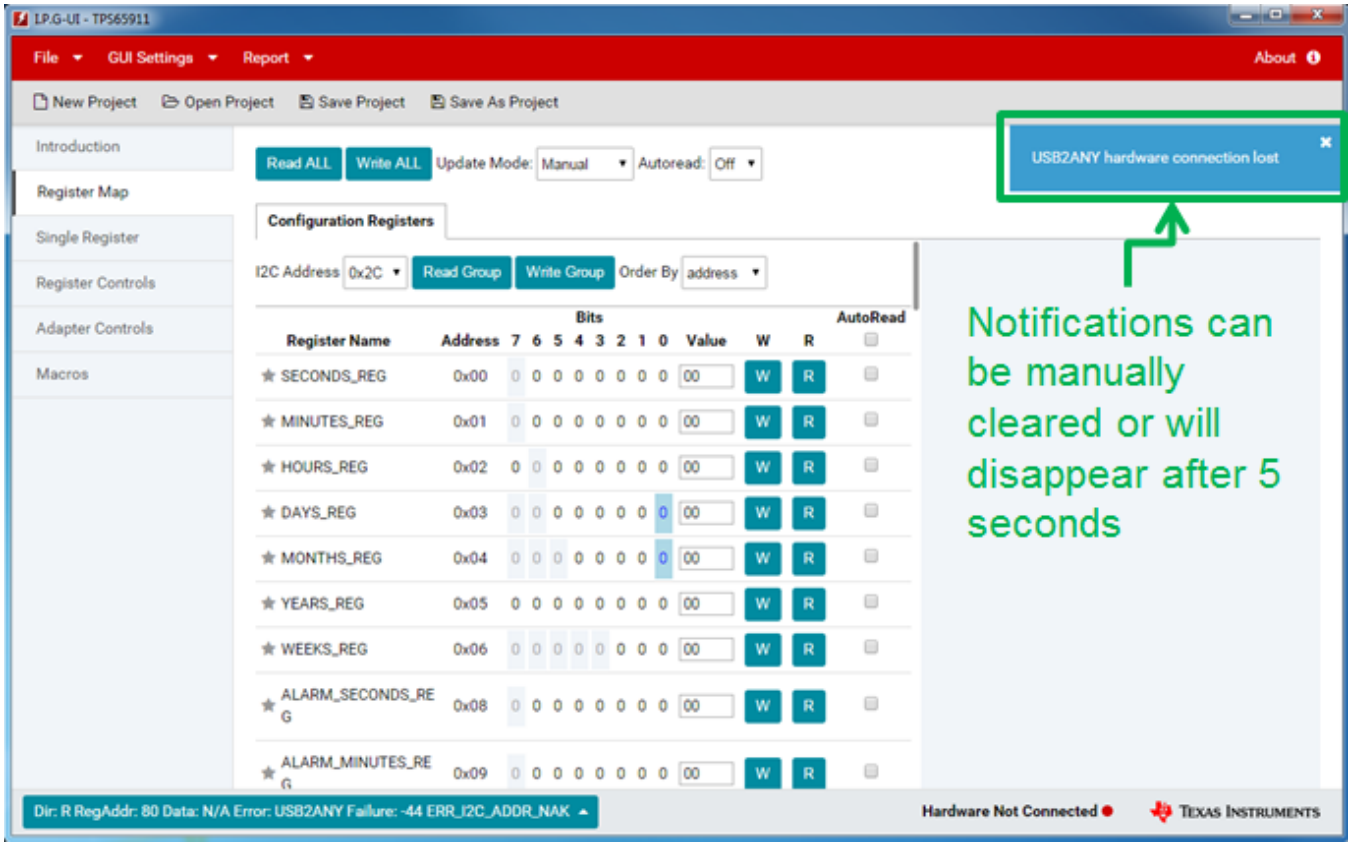

Figure 41. Notifications

22 Search

In the **Register Map** view the user has the option to search the registers and bits for a specified string. The search is located in the upper right corner of the view.

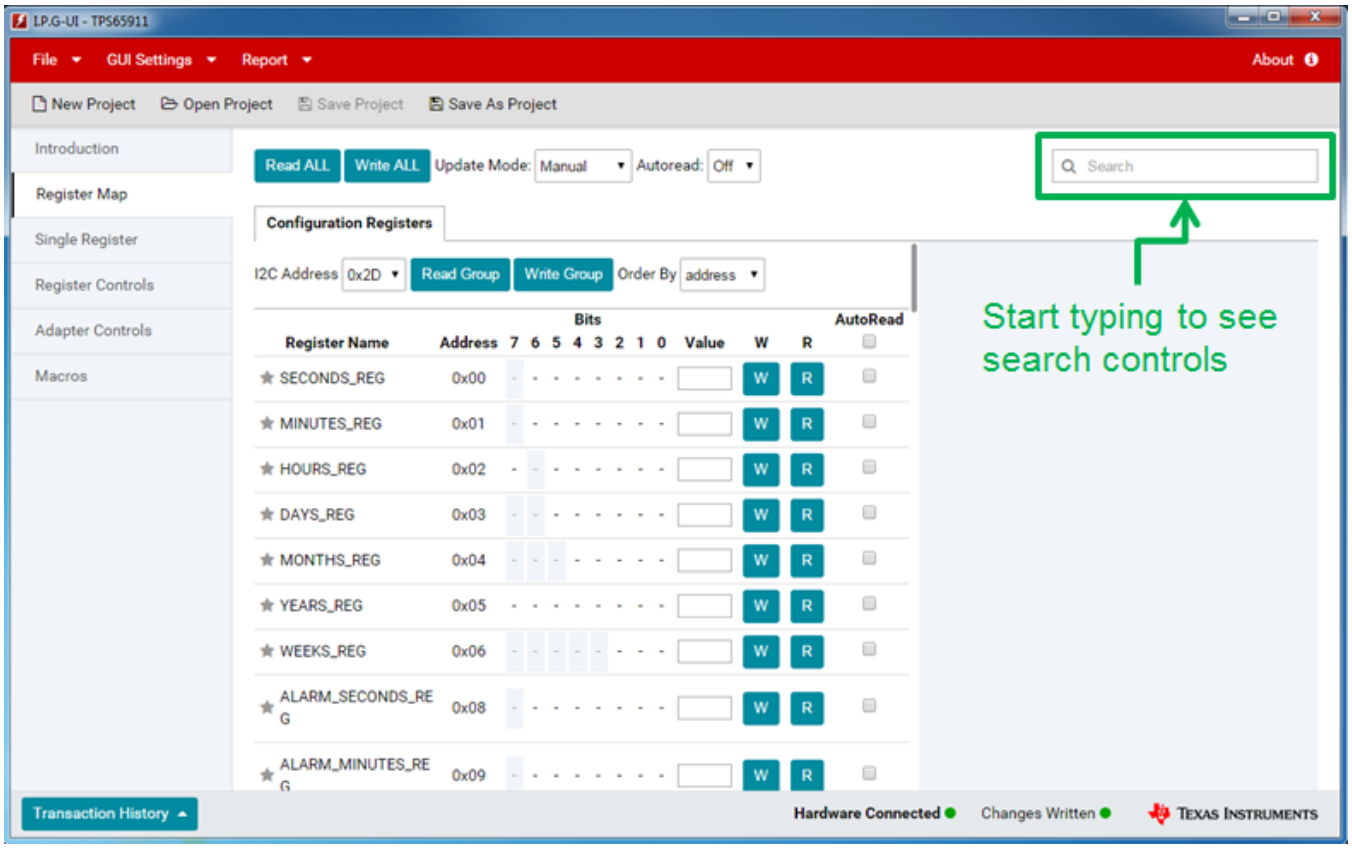

Figure 42. Search

The values searched are the register name, register description, bit name, and bit description. The following search options are available:

- **Search** – This is the string to search for.
- **Search Bits** – Whether to search only register name and description or also search bit names and descriptions.
- **Match Case** – Whether or not to perform a case sensitive search.

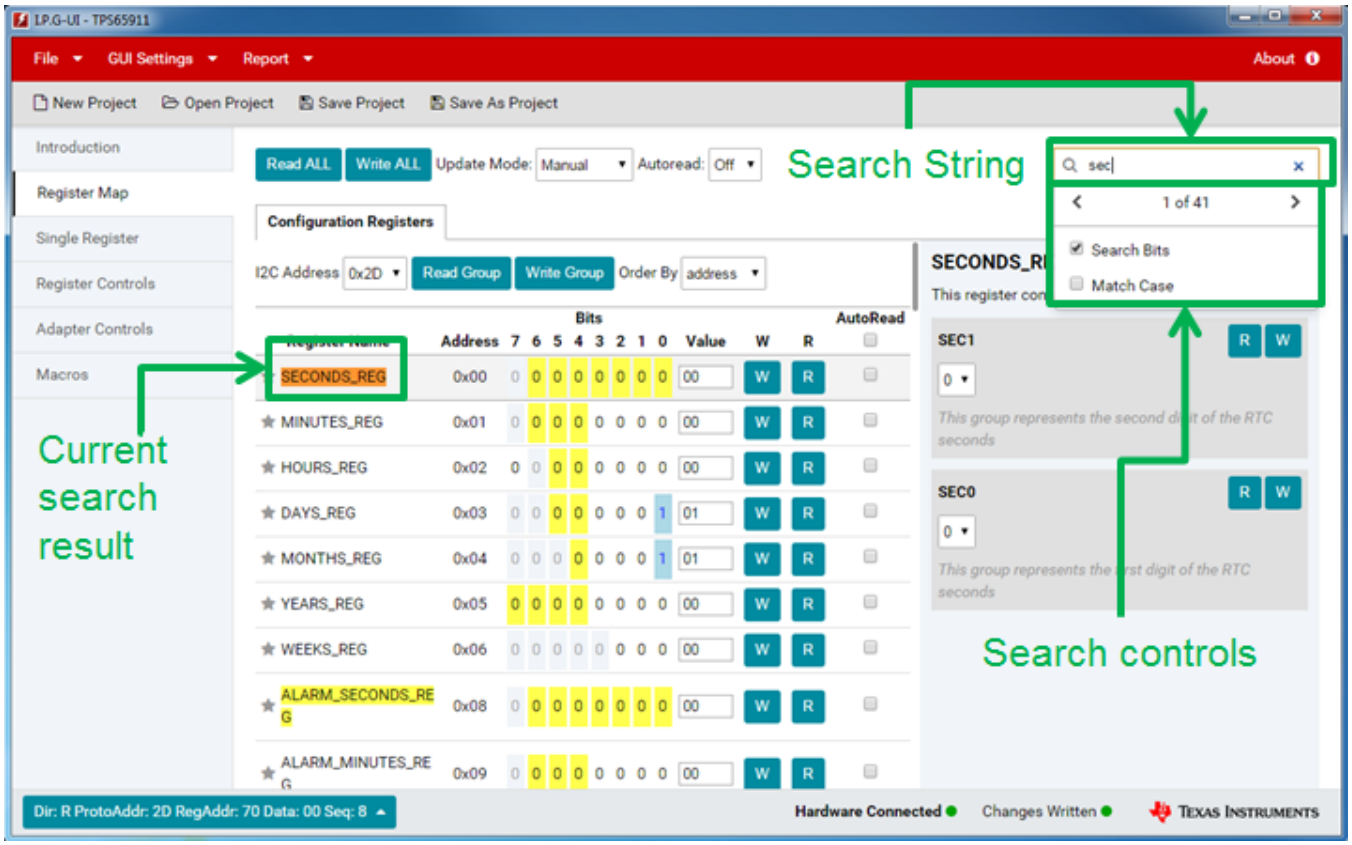

Figure 43. Search Controls

After a search has been performed if there are any results the user can use the controls that appear to step between each search result or clear the search results. The user is also notified which search result out of the total number of matches they are looking at. As the user moves between search results the register map will be highlighted to show the current result and will automatically scroll the table or switch between register bank tabs to show the current result.

23 Frequently Asked Questions (FAQ)

1. How do I make the splash screen go away when the GUI starts?

   Check the USB2ANY adapter. If the LED is blinking this means the adapter is locked and needs to be unplugged and replugged to reset it. During initialization if the adapter is frozen, the GUI waits for the adapter to be reset in order to finish initializing. If the LED is not blinking, try resetting the adapter to clear the issue.

2. How do I determine the version of the IPG-UI and USB2ANY firmware?

   The version of the IPG-UI as well as the version of the USB2ANY firmware can be found by clicking the **About** button in the upper-right corner of the GUI. This will pop up a window showing the GUI and adapter versions along with other information.

3. How do I determine which device the GUI is configured for?

   When a project is loaded or created the device the GUI is currently configured for can be determined by looking at either the title bar of the GUI which will be updated with the device name or by navigating to the Introduction view.

4. How do I uninstall the IPG-UI?

   For the Windows system: The IPG-UI can be uninstalled using the **Programs and Features** control in the Windows **Control Panel**. Select the IPG-UI program and click the **Uninstall** button to remove the IPG-UI software from the host PC.

   For Linux and OSX systems: The uninstall program located in the IPG-UI installation directory can be used to remove the IPG-UI software from the host PC.

5. What version of the USB2ANY firmware is supported?

   The IPG-UI supports USB2ANY firmware 2.7.0.0.

6. Can I update the USB2ANY firmware?

   The IPG-UI does not currently support updating the USB2ANY firmware and relies on a separate software package for firmware updating. The additional software package is the USB2ANY explorer and can be found on the same download page as the IPG-UI.

7. I do not have a USB2ANY adapter. How can I get one?

   Contact your local TI sales team for assistance in obtaining a USB2ANY adapter.

Revision History

Changes from A Revision (July 2015) to B Revision **Page**

- Added additional supported systems 3
- Added installation instructions for other operating systems 4
- Added instructions for launching the GUI for other operating systems 6
- Updated screenshots 6
- Added uninstallation instructions for other operating systems 51

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Revision History

Changes from B Revision (August 2015) to C Revision **Page**

- Added 2.0.0.0 release notes. 4
- Added a third prerequisite and supporting text to the *Linux* section. 5
- Added note to the end of the *Create a New Project* section. 9
- Changed the number os listed files in the recent files list from 5 to 3 in the *Recent Projects and Devices* section. 11
- Added *Device Controls* and *Macros* sections. 12
- Added *Register Controls Auto Read* section. 34
- Added *Device Controls* section. 35
- Added *Macros* sections. 36
- Added *GUI Settings* section. 45
- Added *Report* section. 46

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265