User Manual

DA9313 Performance Board

UM-PM-019

Abstract

This document describes the hardware and software used to evaluate the DA9313 converter.
It is applicable to both the 317-03-B standalone performance board and the 317-04-B master/slave performance board

Contents

Figures

Tables

1 Terms and definitions

BOM	Bill of Materials
CRC	Cyclical Redundancy Check
DUT	Device Under Test
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HW	Hardware
INI	Initialization
LED	Light- Emitting Diode
LS	Level Shifter
OTP	One Time Programmable
PC	Personal Computer
PCB	Printed Circuit Board
PCM	Power Commander Mode
PVC	Power Voltage Converter
SAM3U	USB I^2C Interface
TCP	Transmission Control Protocol
USB	Universal Serial Bus

2 References

[1] DA9313 Datasheet, Dialog Semiconductor

[2] 317-04-B_PerformanceBoard_SCH.pdf, Dialog Semiconductor

[3] 317-04-B_PerformanceBoard_PCB.pdf, Dialog Semiconductor

[4] 317-03-B_PerformanceBoard_SCH.pdf, Dialog Semiconductor

[5] 317-03-B_PerformanceBoard_PCB.pdf, Dialog Semiconductor

3 Introduction

The DA9313 performance board, shown in Figure 1, has been produced to allow measurement, evaluation, and programming of the DA9313 device.

Dialog's control software package SmartCanvas™ uses a simple graphical user interface, allowing DA9313 to be controlled via the USB port of a PC. The mini USB connection is visible on the left side of the board. When the cable is connected to the USB port of the PC the green LED D1 is on.

The board has jumper links, see Figure 3, to provide access to alternative configurations and measurement test points, only a few of these needs to be altered to evaluate most standard operating modes.

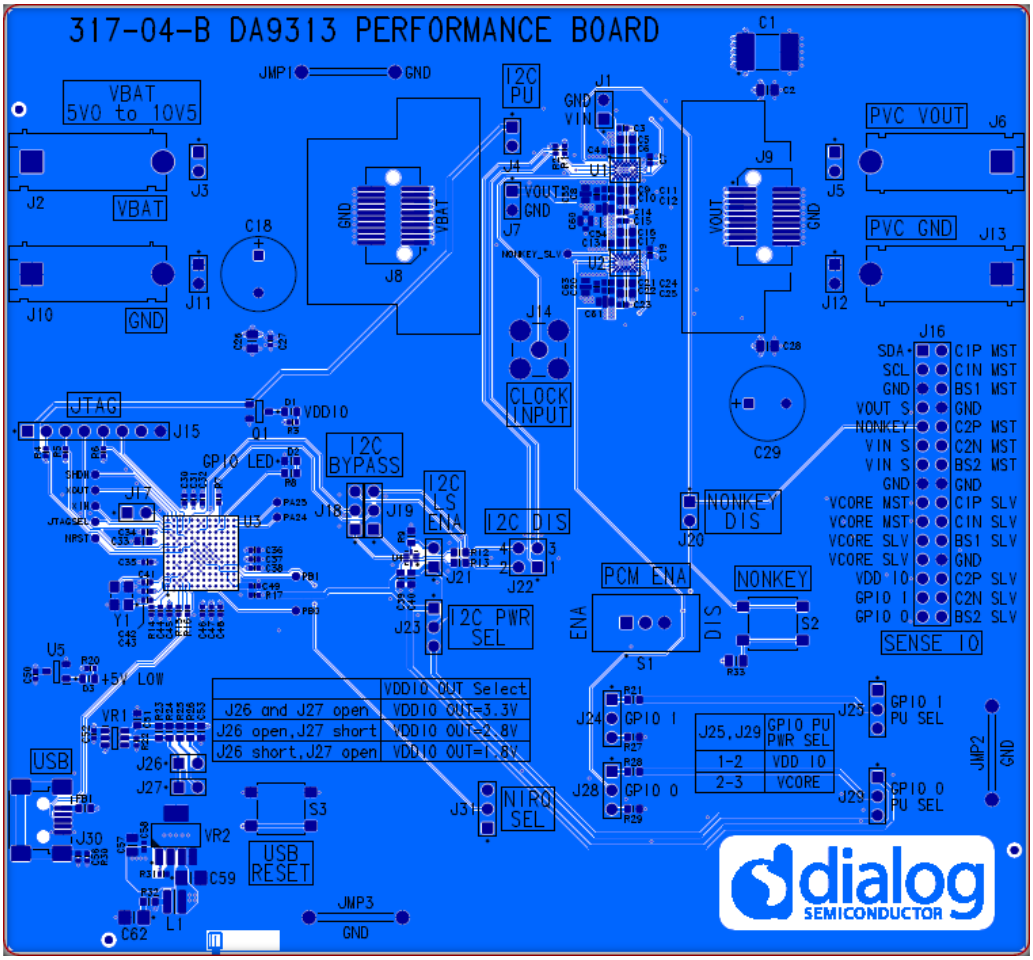

Figure 1: DA9313 Performance Board (317-04-B Shown)

4 DA9313 Performance Board Hardware

The DA9313 performance board functionality can be broken down into five discrete sections, as shown in Figure 2:

1. VBAT power input
2. Device(s) under test (DUT)
3. PVC output
4. USB interface (I^2C communication) and USB reset
5. GPIOs, nONKEY and control switches

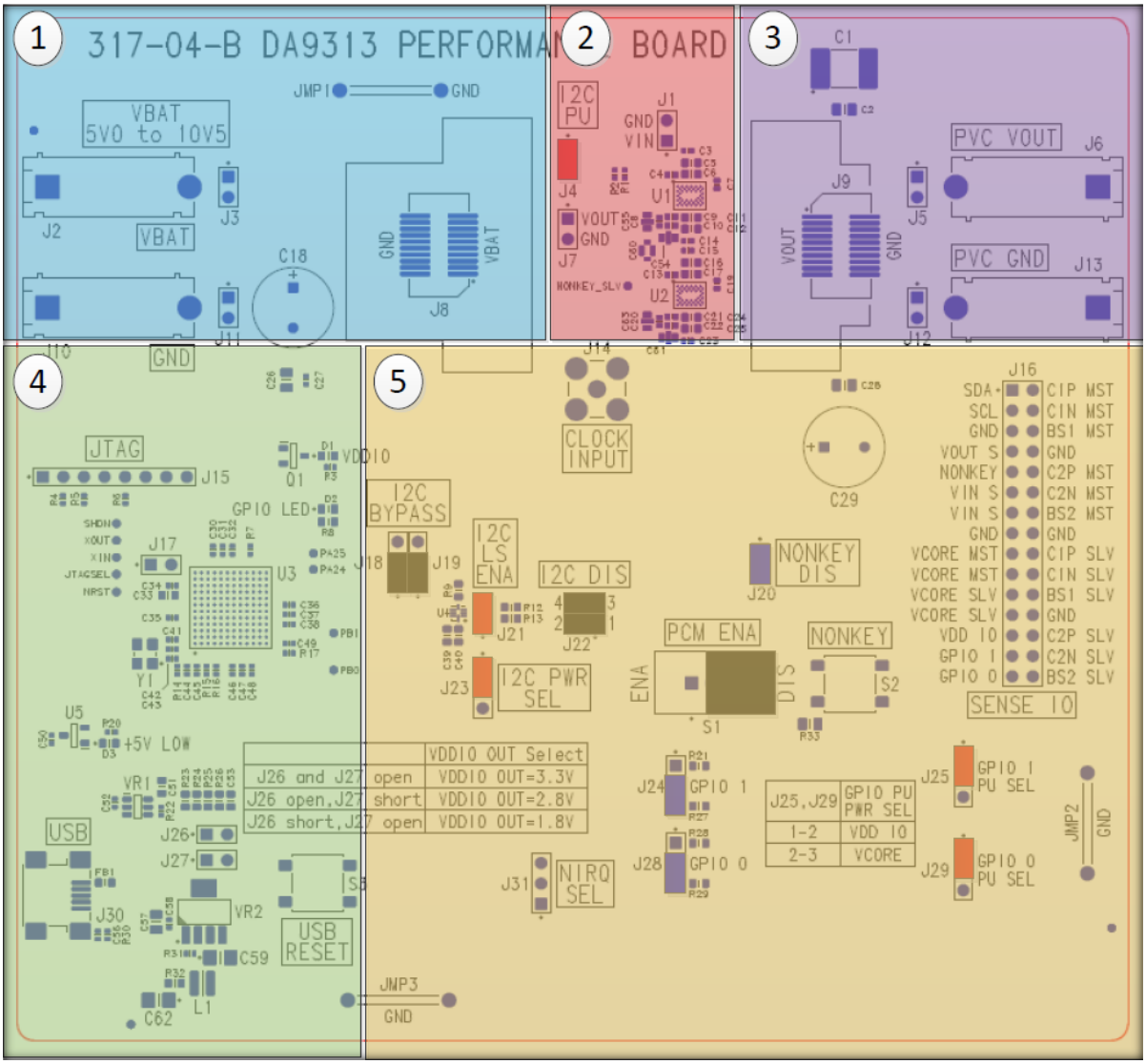

Figure 2: DA9313 Performance Board Sections (317-04-B Shown)

4.1 Default Link Positions and Connector Definitions

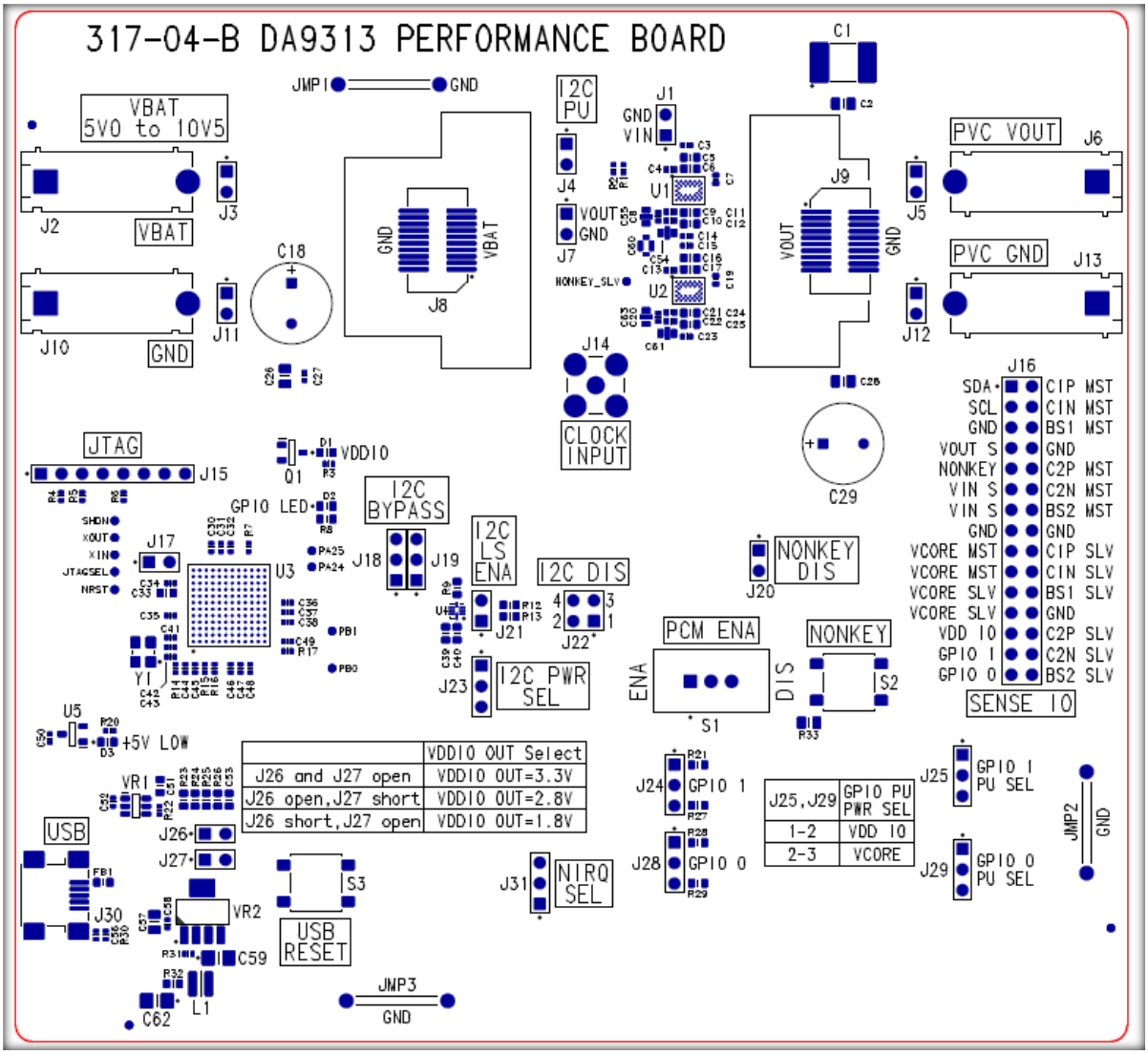

Figure 3: DA9313 Performance Board Default Link Positions (317-04-B Shown)

Table 1: Connector Definitions

Reference Designator	Position	Function
J1	1	VIN sense
	2	GND sense
J2	NA	VBAT
J3	1,2	VBAT sense
J4	1-2	I^2C pull-up
J5	1,2	PVC VOUT (sense)
J6	NA	PVC VOUT (force)
J7	1	VOUT sense
	2	GND sense

DA9313 Performance Board

Reference Designator	Position	Function
J8	NA	Line transient connector
J9	NA	Load transient connector
J10	NA	GND
J11	1,2	GND sense
J12	1,2	PVC GND (sense)
J13	1,2	PVC GND (force)
J14	NA	Clock input connected to GPIO_1
J15	NA	JTAG: reserved
J16	NA	Measurement header
J17	1-2 (OPEN)	Important: This erases the USB IC firmware. DO NOT SHORT.
J18	1-2 (DEFAULT)	I^2C bypass: SAMU3U SCL connected to LS input
	2-3	I^2C bypass: SAMU3U SCL connected to LS output (bypass)
J19	1-2 (DEFAULT)	I^2C bypass: SAMU3U SDA connected to LS input
	2-3	I^2C bypass: SAMU3U SDA connected to LS output (bypass)
J20	1-2 (DEFAULT)	nONKEY connected to S2 switch
J21	1-2 (DEFAULT)	I^2C level shifter IC enabled
J22	1-2	SCL from LS/SAM3U connected to DA9313
	3-4	SDA from LS/SAM3U connected to DA9313
J23	1-2 (DEFAULT)	I^2C LS powers selection: connected to VDDIO
	2-3	I^2C LS power selection: connected to VCORE
J24	1-2	GPIO_0: PU enabled
	2-3 (DEFAULT)	GPIO_0: PD enabled
J25	1-2	GPIO_1 PU selection: VDDIO
	2-3	GPIO_1 PU selection: VCORE
J26	1-2 (UNPOP)	Sets VDDIO to 1.8 V (if J27 open)
J27	1-2 (UNPOP)	Sets VDDIO to 2.8 V (if J26 open)
J28	1-2	GPIO_0 PU selection: VDDIO
	2-3	GPIO_0 PU selection: VCORE
J29	1-2	GPIO_0 PU selection: VDDIO
	2-3	GPIO_0 PU selection: VCORE
J30	NA	USB inlet
J31	1-2	GPIO_0 connected to SAM3U port B3
	2-3	GPIO_1 connected to SAM3U port B3
S1	1-2	ENA: Power Commander mode enabled (+5 V)
	2-3 (DEFAULT)	DIS: Power Commander mode disabled and S2 connected to GND.
S2	NA	nONKEY push button
S3	NA	Push button to reset the USB IC

4.2 USB Interface

The DA9313 performance board uses an ATMEL® SAM3U® microcontroller as the USB transceiver; programmed to deliver the following functionalities:

- I²C control interfaces
- Discrete digital IO control (General Purpose Input Output (GPIO) and dedicated functions)

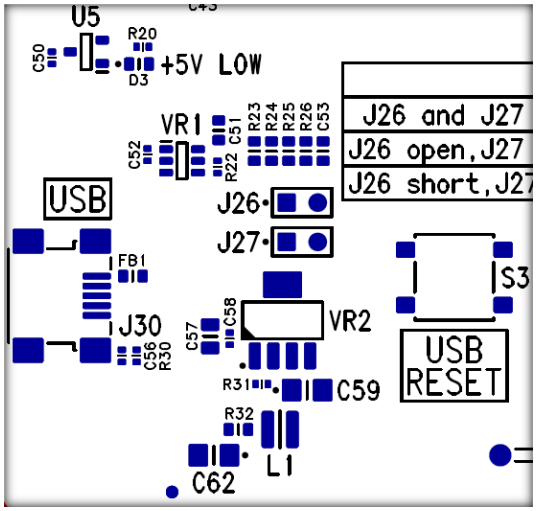

Figure 4: USB Interface Connector and Reset Switch

The USB is powered by an on-board regulator (VR2), as shown in Figure 4.

4.3 Power Supplies

The DA9313 performance board is powered up when a power source is connected to VBAT (J2). VBAT operating range is +5 V to +10.5 V.

The VDDIO voltage is generated by an on-board regulator (VR1) supplied from the +5 V USB. By default, the onboard generated VDDIO is +3.3 V (J26 and J27 jumpers are UNPOP).

- To change the onboard VDDIO to +1.8 V, J26 must be fitted while J27 is UNPOP.
- To change the onboard VDDIO to +2.8 V, J27 must be fitted while J26 is UNPOP.

Pressing the USB RESET switch S3 shuts down the on-board regulator VR2, powering down the supply for the SAM3U (USB I²C interface). This resets the GUI communication with the performance board.

NOTE
It is recommended that the DA9313 performance board is connected to a USB port capable of supplying 500 mA as correct operation cannot be guaranteed if connected to a USB hub that provides only 100 mA.

Table 2: Power Configuration Switch

Switch Name	Reference Designator	Type	Function
USB RESET	S3	Push button	Shuts down the on-board regulator that generates the interface supply

5 DA9313 Performance Board Software

The board is controlled using the SmartCanvas software graphical user interface (GUI), which requires a PC operating Windows® 2000/XP/Vista/Windows 7 with a USB1.1 or USB2 interface. The GUI allows the user to:

- Configure the DA9313 device in Power Commander mode or Normal mode.
- Perform write and read operations to all control registers.
- Monitor the device status.

5.1 SmartCanvas GUI Installation

The files required to install the software are available on the supplied USB drive. To install the DA9313 SmartCanvas software:

1. Run **setup_DA9313_GUI_1.1.x.x.exe**. The program default install location is: **C:\Dialog Semiconductor\Power Management\DA9313 GUI**.
2. On completion, plug in the performance board and apply VBAT. The software must be started after the USB board is plugged in or communication with the board may fail.
   a. For first time users, Windows should detect the attached USB device. If this is not the case, it may be necessary to install the driver by navigating to the required driver file in the USB driver directory, see section 5.2. For guaranteed operation, it is recommended that a PC reboot be carried out after installing the driver.
3. Once installation is complete, run the DA9313 SmartCanvas software: **DA9313 GUI.exe**.

5.1.1 SmartCanvas GUI Installation Step-by-Step Guide

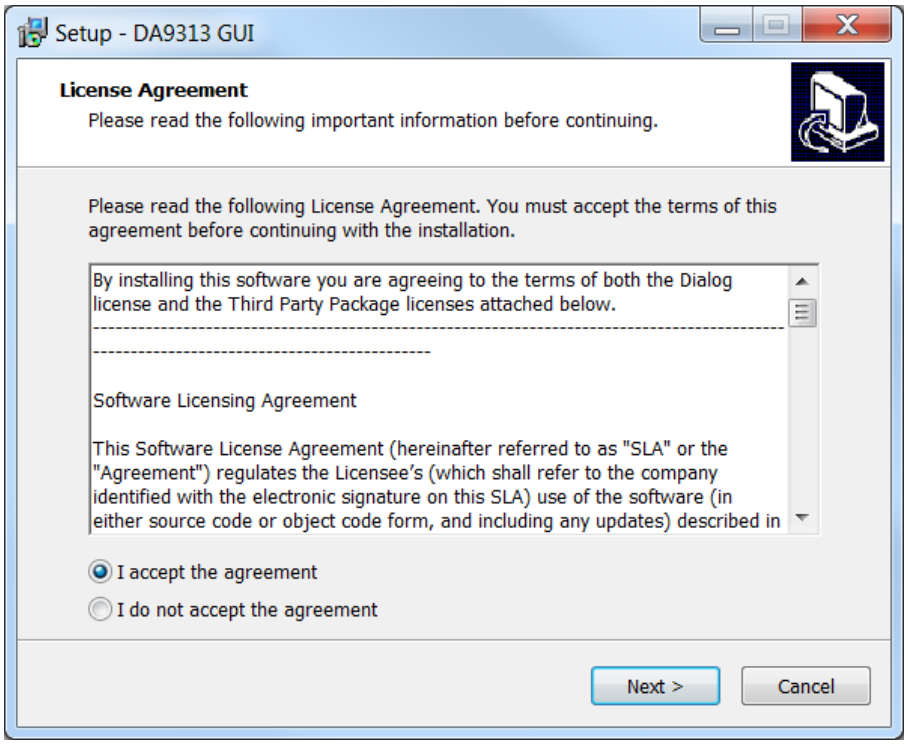

Figure 5: GUI Setup License Agreement

1. Select **I accept the agreement** then click **Next**.

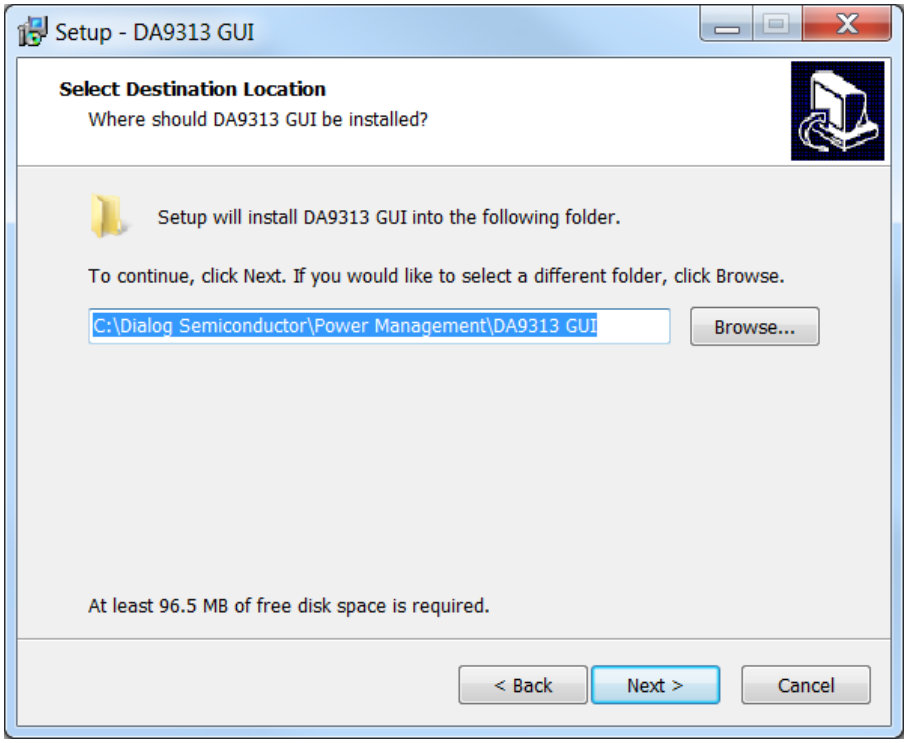

Figure 6: GUI Setup Destination Location

2. Click **Next**.

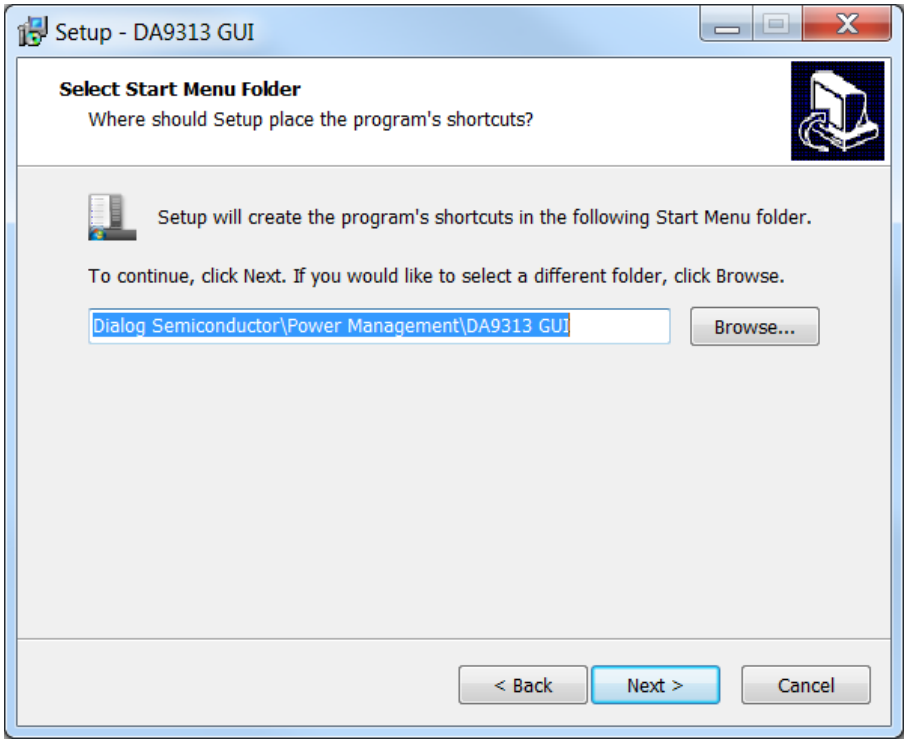

Figure 7: GUI Setup Shortcuts Location

3. Click **Next**.

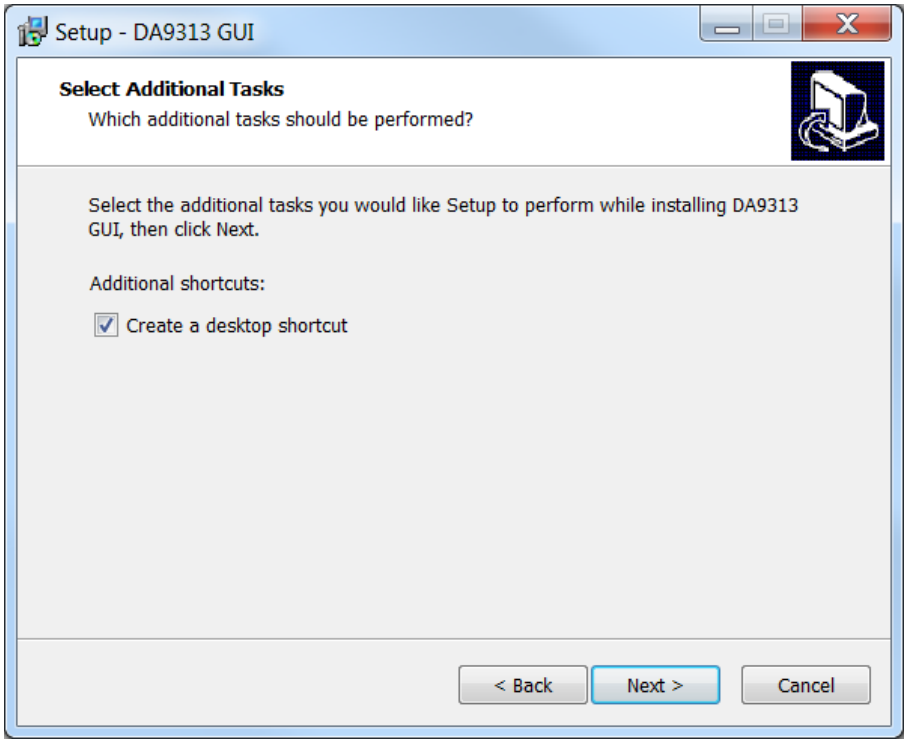

Figure 8: GUI Setup Create Shortcut

4. Select the **Create a desktop shortcut** check box and click **Next**.

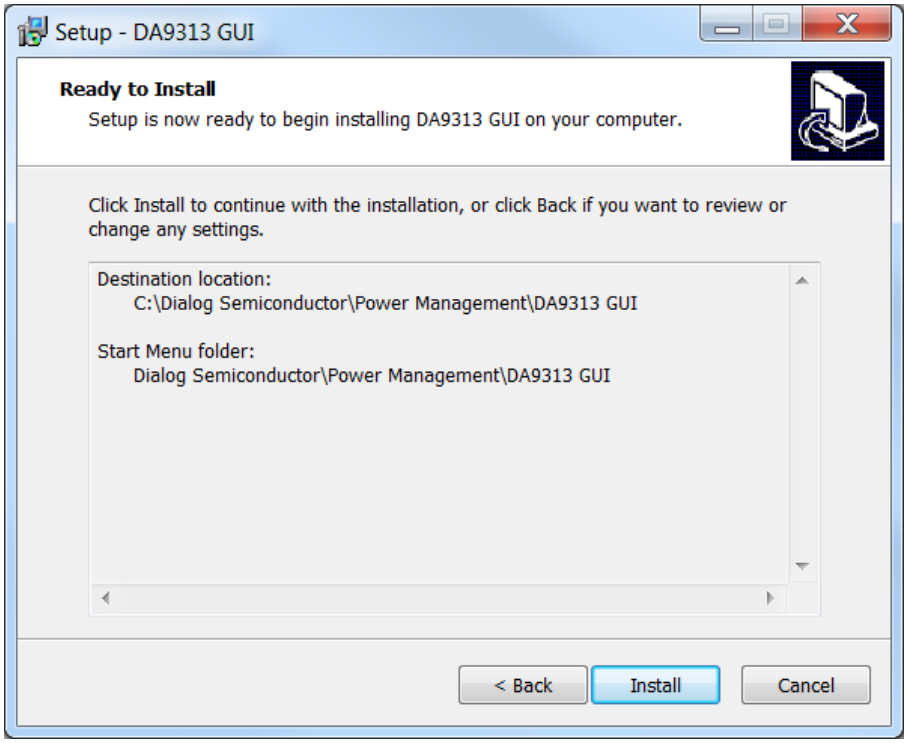

Figure 9: GUI Setup Installation

5. Click **Install**.

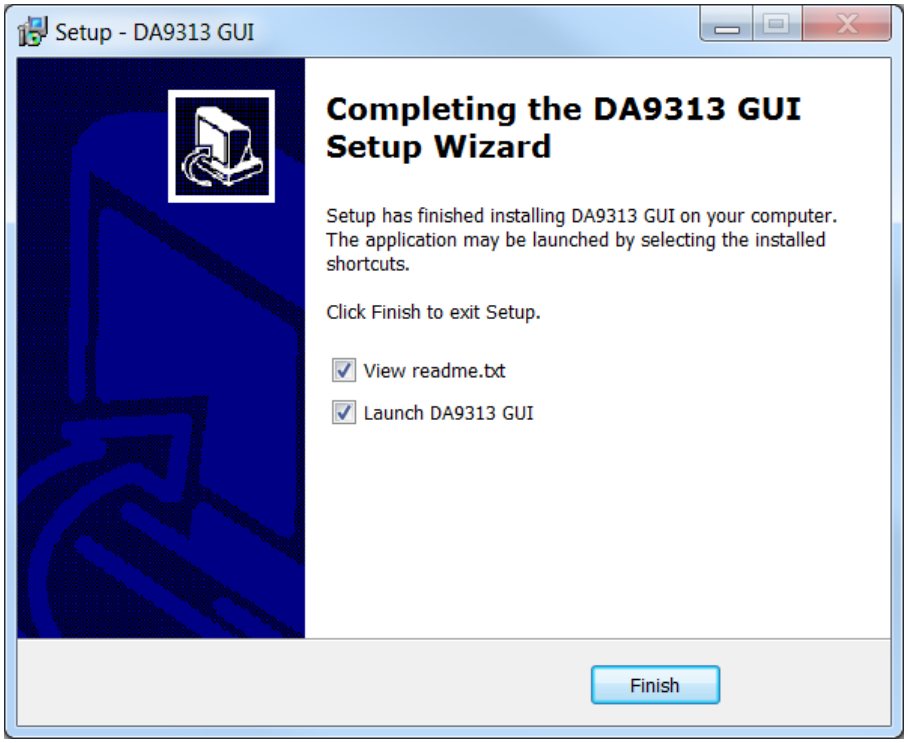

Figure 10: GUI Setup Finish and Launch

6. Click **Finish**.

Once the installation is complete, you may need to restart your computer.

When the software is installed, insert the USB cable before applying VBAT. The DA9313 SmartCanvas software can be started after power up.

5.2 Initial USB Connection

On connecting the USB to the PC for the first time, the SAM3U USB driver will request driver updating/installation from the Windows operating system. On Windows 7 (32-bit) operating systems (OS) the driver usually installs automatically. On Windows 7 (64-bit) machines it is common for the complete driver installation to fail. If this happens you must install the driver manually by following these steps:

1. **Control Panel** → **Devices and Printers** (double-click device with yellow exclamation sign).
2. **Update Driver**.
3. **Browse my computer for driver software**.
4. Select the **Driver** folder location: **C:\Dialog Semiconductor\Power Management\DA9313 GUI\driver**.
5. If Windows warns about the driver, select **Install anyway**.
6. Remove the USB cable then reinsert it into the performance board.

6 SmartCanvas Software

Run the DA9313 SmartCanvas software by clicking the shortcut on the appropriate item in the **Start menu** (**All Programs → Dialog Semiconductor → Power Management → DA9313 GUI**). The main GUI interface is displayed, see Figure 11.

The minimum recommended setting for the PC display size is 1024x768 pixels. Font size on the PC display should be Normal (95 dpi).

NOTE
It is important to note that a display size other than the recommended setting will affect the way in which the panels appear.

6.1 Register Controls

Device registers are displayed as a group of controls. Selected bit ranges within a register make up a control. Register data is always a standard bit width dictated by the device register map, but a control can be anything from 1-bit to the full register data width.

Registers are grouped together on tabs to assist with identification of device function or registers of the same type.

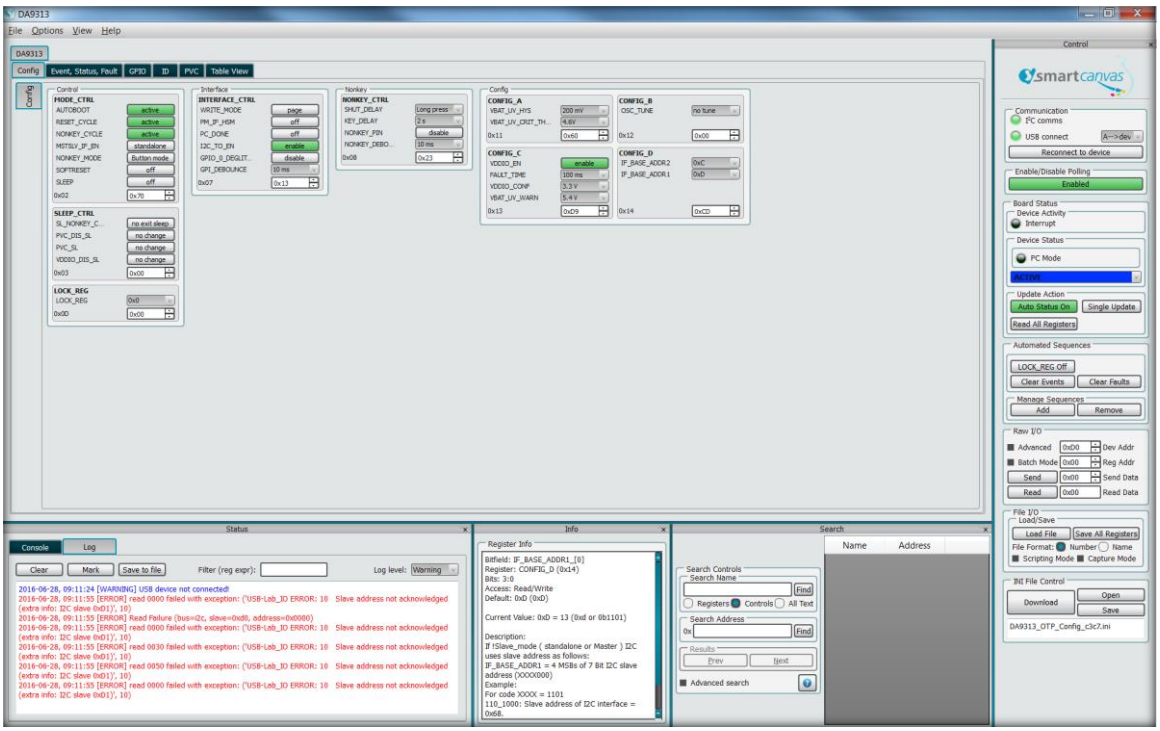

Figure 11: Main Interface

6.2 Table View Tab

The **Table View** tab, see Figure 12, shows the complete register map. From this tab all the registers can be set or read back. Clicking on any of the bit groups on the map will allow access to the full control settings; these controls will read and write the same value elsewhere on the interface.

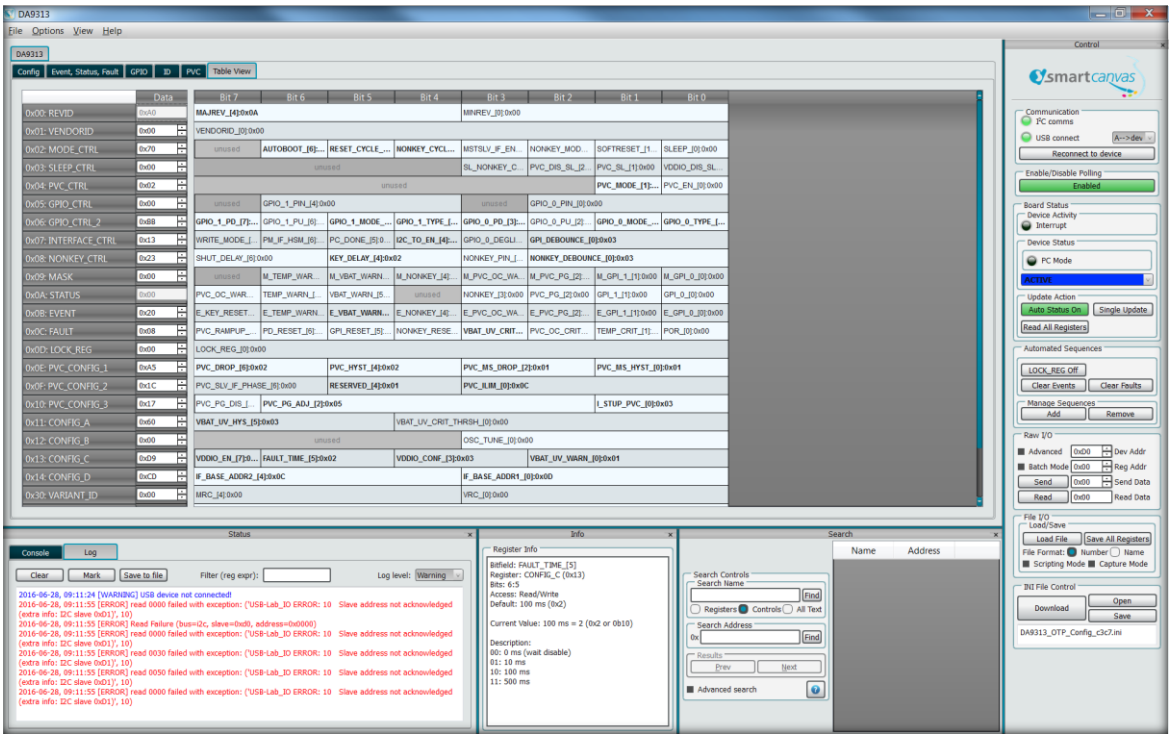

Figure 12: Table View

6.3 Control Windows

All dockable control windows, either on the right or bottom of the main window add additional functionality or monitoring to the GUI. The windows can be undocked by clicking on and dragging the title bar of that window and can be placed anywhere on the screen. If desired the **Search** and **Info** windows can be docked on the left hand side of the main window for greater convenience.

6.3.1 Communication

The **Communication** window, see Figure 13, has indicators to show when the Atmel SAM3U USB module is plugged in and when I²C communication is possible; the currently active USB device number is also shown. Most of the time the communication link will automatically connect if the USB is active and the device is powered up. On rare occasions, sequence of events may prevent recognition of the active communication link, pressing the **Reconnect to device** button will recover the link.

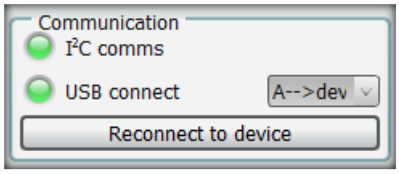

Figure 13: Communication Control

6.3.2 Enable/Disable Polling

The Enable/Disable Polling button, see Figure 14 enables or disables polling of the device registers and the refreshing of the registers controls on the GUI interface.

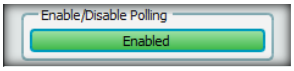

Figure 14: Polling

6.3.3 Board Status

The **Board Status** window, see Figure 15 displays the following information:

- An indicator to show when the Interrupt line has gone high.
- Indicator to show if Power Commander mode is active.
- A box to display the current device operational status.

The update actions are as follows:

- **Auto Status On** – when enabled the indicators described above are constantly valid if the hardware device registers are being actively polled.
- **Single Update** – even if the device registers are not being continuously polled then a single poll of the indicators described above can be carried out.
- **Read All Registers** – even if the hardware device is not being actively polled then all the registers can be polled once by pressing this button.

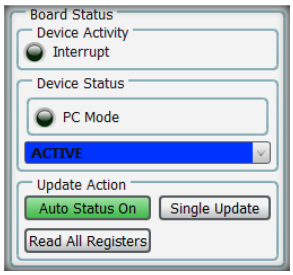

Figure 15: Board Status

6.3.4 Raw I/O

The **Raw I/O** control, see Figure 16 sends the entered device address, register address and data on the I²C communications interface. If the information sent is not valid then the I²C message will return NACK an error message will be displayed in the **Status** window.

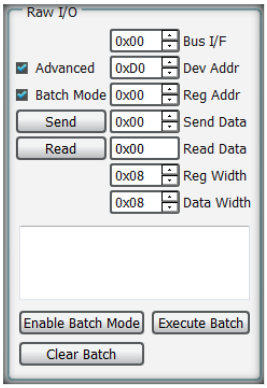

Figure 16: Raw I/O

There are two optional modes selected by check boxes **Advanced** and **Batch Mode**.

6.3.5 Automated Sequences

Predefined automated sequences may be present in this window, see Figure 17. These can be either single-shot buttons or latched buttons, allowing separate on and off scripts as required.

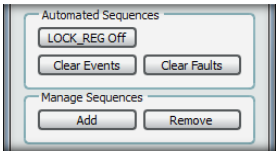

Figure 17: Automated Sequence

It is also possible to add sequences which can be defined in the **Create New Button** window, see Figure 18. If the button is checkable then an off script, a monitoring register, control name, and monitoring value that tests 'True' also need to be assigned. This value will be monitored to ensure that the button state is displayed correctly if device power or communication is reset.

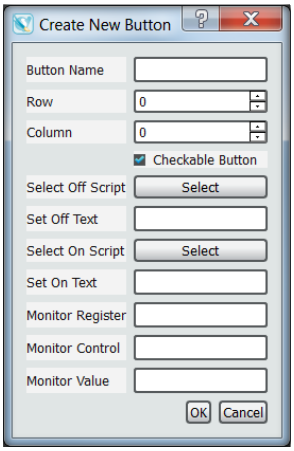

Figure 18: Automated Sequence - Create New Button

An automation button and its associated control scripts can be removed using the **Remove Button** window, see Figure 19. The assigned button name must be used to remove the button and not the on or off text; this name can be found by hovering over the button to display the tool tip.

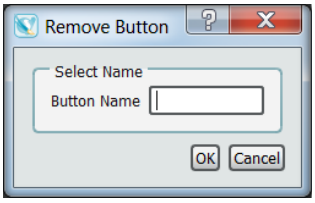

Figure 19: Automated Sequence - Remove Button

6.3.6 File I/O

The **File I/O** control, see Figure 20, **Load File** button allows formatted text files to be loaded into the device registers, which will then be reflected on the **Registers** display of the GUI interface.

The file scripting options are listed below. All numeric register addresses or data values must be in hexadecimal with prefix 0x.

- WRITE [device name][register address][register data value]
- READ [device name][register address]
- DELAY [time in milliseconds] – no time units suffix required
- USBIO [USB IO pin index][output value] – Atmel USB device output value = 0 or 1
- ACTION [text message] – the file being loaded will pause until the message dialogue has been acknowledged

An example script:

```
READ DA9313 0x06
WRITE DA9313 0x06 0xBB
DELAY 100
ACTION Stop script until cleared
```

The **Save All Registers** button will save the register values displayed on the GUI interface. If saving with the extension .txt or .csv, one of the **File Format** radio buttons can be selected to save by either register name or register number.

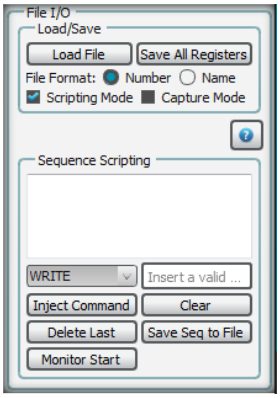

Figure 20: File I/O

An additional **Scripting Mode** makes it possible generate a correctly formatted text file using the same file formats as above.

When **Capture Mode** is enabled, the press of any button on the main control interface will be recorded onto the **Sequence Scripting** window. When the sequence has been completed it can be saved as a text file by pressing the **Save Seq to File** button.

An additional function command **MONITOR** is also present in the drop down list. This option is a special function which allows one of the controls on the interface to be monitored continuously until a defined value has been read back.

The format is:

- MONITOR [device name][register address][control name][monitoring value]

Scripting Mode generates a text file using the formats below. All numeric values must be in hexadecimal with prefix 0x.

- WRITE [device name or device address][register address][register data value]
- READ [device name or device address][register address]
- DELAY [time in milliseconds] – no time suffix required
- USBIO [USB IO pin index][output value] – Atmel USB device output value = 0 or 1
- ACTION [text message] – the file being loaded will pause until the message dialogue has been acknowledged

NOTE
If the GUI only refers to one device, then for WRITE and READ entries it is only necessary to enter the register values as the device address or name will be entered automatically

6.3.7 INI File Control

The **INI File Control** window, see Figure 21 allows the loading of .ini file formatted project files to the main interface registers by pressing the **Download** button. The **Open** button will load the chosen file path ready for download to the device. The **Save** button will save the main interface register values of the related OTP registers as an .ini formatted file.

If the device has been powered up in Power Commander mode and the file has been downloaded to the interface, the **Download** button will change to **Start**. The **Start** button can then be pressed to set the device into ACTIVE mode by following the on screen instructions.

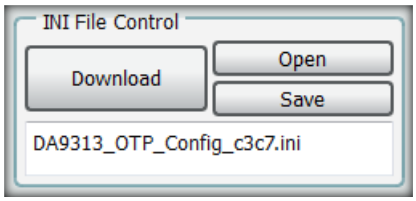

Figure 21: INI File Control

6.3.8 Search

The **Search** control, see Figure 22, allows the user to focus in upon a register by name or number, or a by particular text term contained within the GUI. An entry made in the **Search Name** box will search register names and display the instances found in the right hand side list box. Alternatively if **All Text** is selected, all entries containing the written text will be identified and listed.

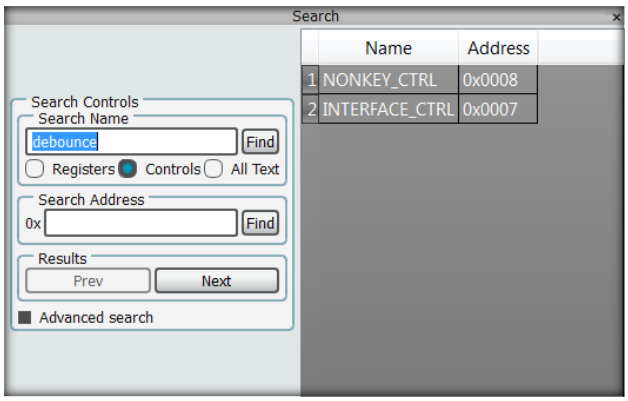

Figure 22: Search

If **Advanced Search** has been selected, selecting **Reg expr** will allow a search based upon Python® regular expression functions.

6.3.9 Info

The **Info** window, see Figure 23, will display a description of an interface control including: name, parent register and the bits to which this control corresponds, current value, whether it is read only or R/W, and finally a description of each possible setting.

For long descriptions the window may either be undocked and made larger, or docked at the left hand side of the main window.

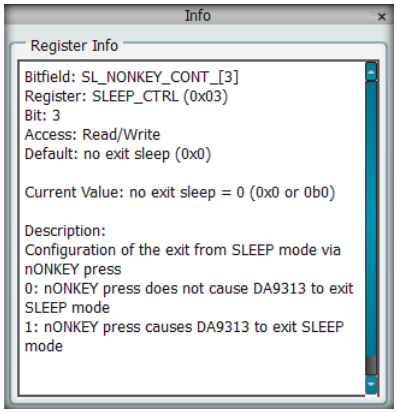

Figure 23: Info

6.4 Menu Items

There are a number of menu items found on the top line of the interface, which allows the selection of enhanced functionality. These are not required in most cases of interface use.

6.4.1 Options → Settings

The **Settings** control, see Figure 24, under **Options** in the menu, allows:

- **Numerical Base**: changes of the numerical base used throughout the GUI.
- **Polling**: enables/disables polling and allows the polling rate to be changed. There is an option to poll all or just the visible registers, however selecting **Poll Visible Only** may have adverse effects on the functionality of the automation controls and is therefore not recommended.
- **Bus Interface**: allows the selection of any of the hardware communication buses available on the board and if I^2C is selected then the frequency of the bus can be altered.
- **Comms Server**: for use with the TCP protocol, but may not be relevant to all boards.

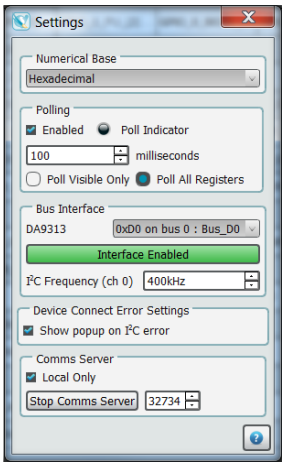

Figure 24: Settings

6.4.2 Options → OTP

The **OTP** window, see Figure 25, is an advanced tool that is used to program the one time programmable (OTP) registers in the device.

The following controls are available:

- Read OTP Registers: read the hardware device OTP register values
- Load HW Defaults: load the defaults defined for the main user interface
- Load OTP from File: open an OTP .ini file and set the OTP addon table ready for programming or editing
- Load OTP from Registers: transfer the values from the main user interface into the OTP addon table
- Save OTP to File: save the OTP values shown on the OTP addon table to an OTP .ini formatted file
- Transfer OTP to Registers: transfer the register values in the OTP addon table to the registers on the main user interface
- Program OTP: program the OTP registers if VBAT = +7.5 V

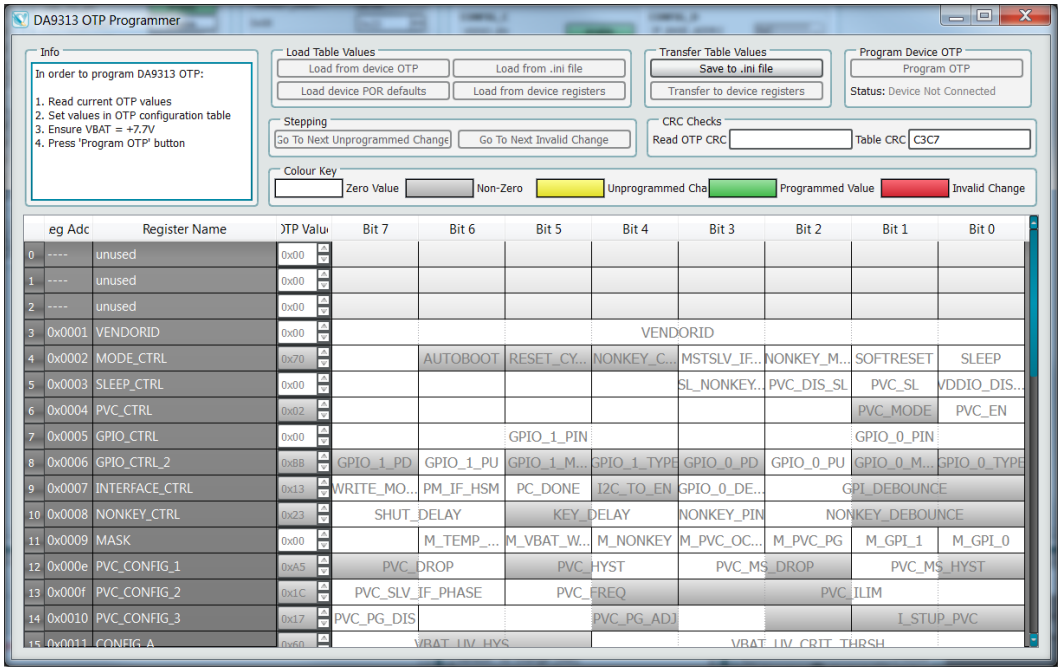

Figure 25: OTP Programmer

Register status indicators color coding:

- Grey: unassigned and unprogrammable OTP register
- White: unprogrammed bit set to value zero
- Yellow: unprogrammed bit set to value one
- Green: programmed value read from hardware register
- Red: invalid setting

6.4.3 Options → USBIO

Certain SAM3U USB IO pins are connected to the DA9313 to allow dedicated hardware control options. It is also possible to set some of the IO pins as GPIOs to configure more general functions in the device, see Figure 26. These GPIO options are:

- Input: the DA9313 drives a signal in to the USB IO pin.
- Output: the USB IO pin drives a signal in to the DA9313.
- Input: PullUp: the USB IO pin has an internal pull-up resistor the DA9313 drives a signal into the USB IO pin.
- Output: PullUp: the USB IO pin has an internal pull-up resistor and the USB IO pin drives a signal into the DA9313.
- Input: Multi-driver: the evaluation board has an external pull-up resistor and the DA9313 drives a signal into the USB IO pin.
- Output: Multi-driver: the evaluation board has an external pull-up resistor and the USB IO pin drives a signal into the DA9313.
- Input: Pull-up and Multi-driver: the pin supports multi-driver open drain option from multiple external drivers.
- Output: Pull-up and Multi-driver: the pin supports multi-driver open drain option into multiple external targets.

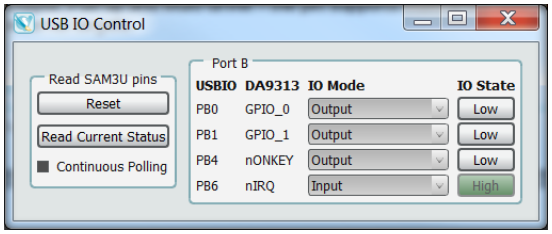

Figure 26: USB IO Control

⚠ **CAUTION**
Incorrect use of the IO settings could cause damage to the SAM3U or DA9313 devices

6.4.4 View Menu

The **View** menu reopens the docking windows if they have been previously closed.

7 Power Commander Mode

The Power Commander mode is a special mode for evaluation and configuration using the DA9313 SmartCanvas software and evaluation board. In Power Commander mode, the DA9313 is configured to load the control register default values from a control interface instead of from the OTP cells.

To enter the Power Commander mode, +5 V needs to be momentarily applied to the DA9313 nONKEY pin before VBAT is applied. The DA9313 latches this information and the +5 V can then be removed.

Using the performance board, enter Power Commander mode as follows:

1. Remove all power (VBAT and USB) from the performance board.
2. Set S1 to position 1-2:
   a. S1 (PCM_ENA) is in ENA position, see Figure 28.
3. Insert the USB cable into the performance board.
4. Press down S2 and apply VBAT at the same time. This will apply the +5 V USB to the nONKEY.
5. Run the DA9313 SmartCanvas software. The Power Commander mode is now latched by the DA9313.
6. **IMPORTANT**: Set S1 (PCM_ENA) back to position 2-3 (DIS). This allows usage of the nONKEY push button (S2).
7. The DA9313 SmartCanvas **Board Status** control, see Figure 27, will show that the device is in **WAIT_OTP_DONE** state.

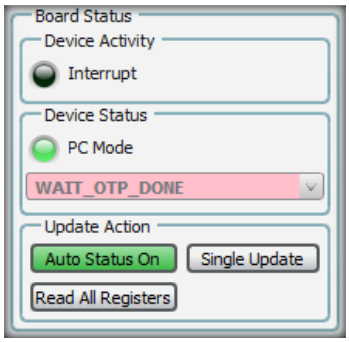

Figure 27: WAIT_OTP_DONE State

8. Open the **.ini** configuration file using the DA9313 SmartCanvas **INI File Control Open** control, see Figure 21.
9. Click on the **INI File Control Download** control to download the **.ini** file data into the registers display of the GUI interface, and then click on the **INI File Control Start Device** control.
   a. If AUTOBOOT is set, the DA9313 will progress automatically to ACTIVE mode.
   b. If AUTOBOOT is not set, the DA9313 will progress to POWER_DOWN mode. There is no I^2C communication possible in POWER_DOWN mode, press the nONKEY to progress to ACTIVE mode.

This completes the sequence to enter the Power Commander mode.

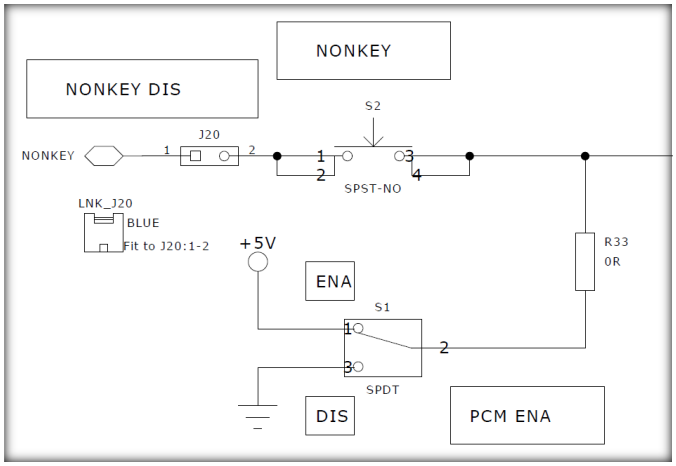

Figure 28: Power Commander Switches

8 Register Text File

The DA9313 SmartCanvas software includes the ability to save and load a text file containing command codes representing the register addresses and data. This file is principally used to save and load setup data, but may also be used to perform a small degree of automation.

The use of the **Save All Registers** in the **File I/O** control permits register contents to be transferred to the user's own software. This is a register dump of the entire device. If **Name** is selected instead of **Number**, then register names rather than register addresses are saved in the text file.

8.1 I²C Register Text File Format

The following formats are used for both read and write in the text file:

- WRITE: will write to the device: WRITE [device name] [register address] [register data value].
- READ: will read from the device: READ [device name] [register address]. The result of the read is passed to a **File Readback Values** pop-up window.
- DELAY: will implement a time delay specified in milliseconds: DELAY [time in milliseconds] – no time suffix required.
- USBIO: will control the Atmel USB device IOs: USBIO [USB IO pin index] [output value] – Atmel USB device output value = 0 or 1.
- ACTION: will allow pausing of the file being loaded until the pop-up message dialogue has been acknowledged: ACTION [text message].

Numbers are always expressed in Hex, separated by tabs. The use of 0x in front of the hex value is mandatory. Inline comments (lines beginning with //) are permitted in the file. The data will be processed in the order written and written directly to the specified device.

For example:

```
WRITE DA9313 0x06 0xBB
WRITE DA9313 GPIO_CTRL_2 0xBB
READ DA9313 0x06
READ DA9313 GPIO_CTRL_2
DELAY 1000 // delay is 1000 ms or 1 s
USBIO 0 1 // "0" refers to the index number for the USB IO pin on the "USB_Ports" tab.
ACTION Please press the OK button to continue.
```

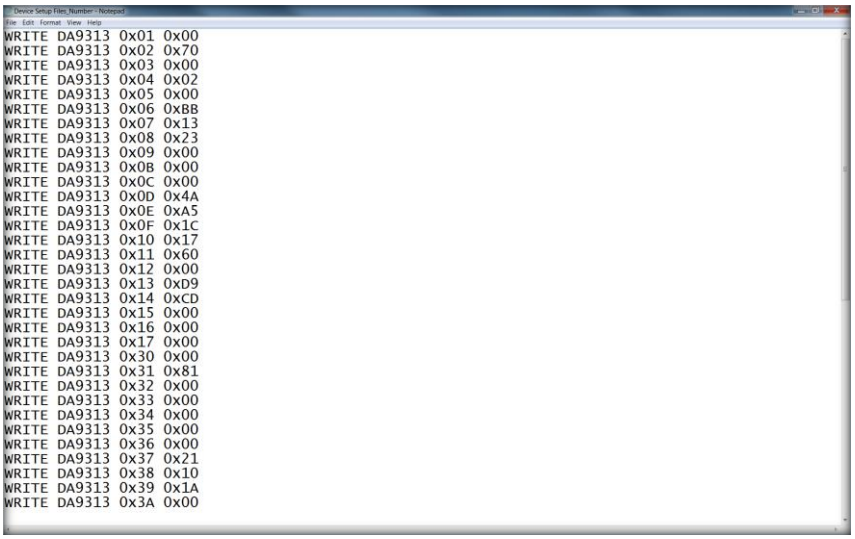

Figure 29: Register Dump (.txt File) Example

9 Configuration File (.ini)

A configuration file (*.ini) can be created using the **INI FILE Control Save** button.

Select **Save as type: DA9313 *.ini** on the **Save Register Dump** pop-up window and click on **Save** to save the register dump on the user's own software.

This will save all available registers for the DA9313 device and create a GUID that includes a unique CRC.

To make an edit to a configuration file, use the DA9313 SmartCanvas software to upload the configuration file to be modified (**INI FILE Control Open** button), make the modifications required to the register and save the new .ini file using the **INI FILE Control Save** button.

NOTE
The CRC in the new configuration file will be different to the one in the previous unmodified file

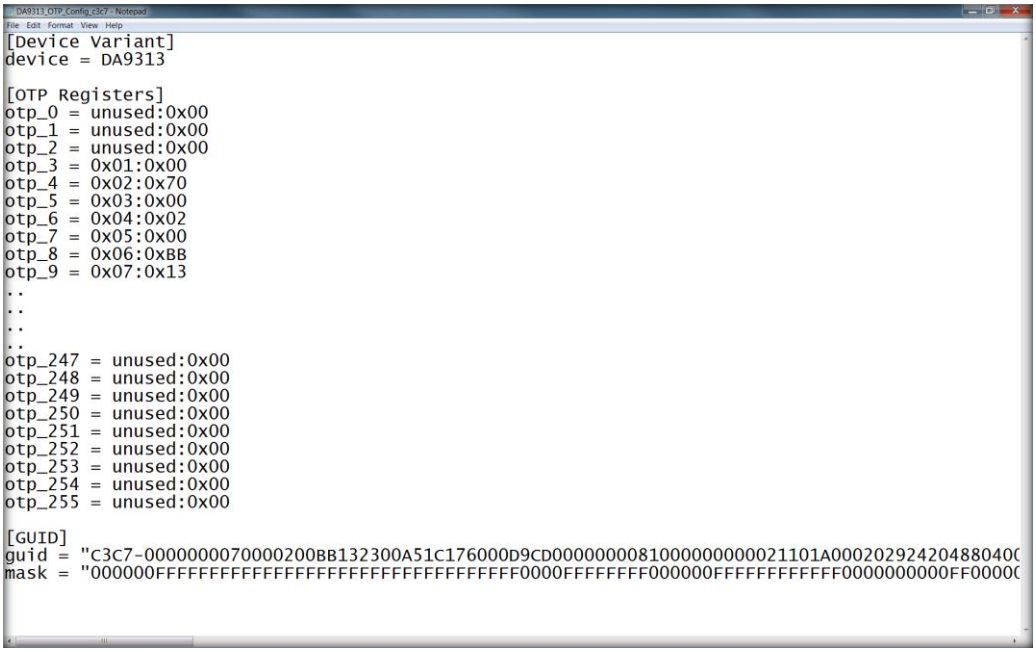

Figure 30: Configuration File (.ini) Example

The GUID string is one value (key) in the "ini" file. The GUID key should be included in a section indicated by the title [GUID]. The second key to be included in the GUID section is the MASK string.

The GUID string is a text string starting with the letters GUID and containing 512 characters, for a 256 byte OTP. The line also contains a 16-bit CRC code created from the 256 values AND'ed with the MASK string.

Each pair of characters represents the Hex value of the 8-bit register value.

Revision History

Revision	Date	Description
2.2	16-Feb-2022	Rebranded to Renesas.
2.1	21-Feb-2017	PMIC replaced by converter or removed in some instances. Abstract updated.
2.0	27-Sept-2016	Updated for 317-03 / 317-04 Rev B boards.
1.0	07-Jul-2016	Initial version.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

DA9313 Performance Board

Important Notice and Disclaimer

(Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu

Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

https://www.renesas.com/contact/

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.