User Manual

DA9053 GUI and EVB User Manual

UM-PM-026

Abstract

This document describes the hardware and software used in Dialog Semiconductor to test and evaluate the DA9053 Power Management Controller.

Contents

Figures

Tables

1 Terms and Definitions

ADC	Analog-to-Digital Converter
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
LDO	Low DropOut voltage regulator
LED	Light Emitting Diode
NTC	Negative Temperature Coefficient (thermistor)
OTP	One Time Programmable (memory)
PC	Personal Computer
PCB	Printed Circuit Board
PFM	Pulse Frequency Modulation
PWM	Pulse Width Modulation
RTC	Real Time Clock
USB	Universal Serial Bus

2 References

[1] DA9053, Datasheet, Dialog Semiconductor.

3 Introduction

The DA9053 Evaluation Board has been produced to allow measurement, evaluation and programming of the DA9053 device. It is supplied together with a USB stick containing documentation and driver files. The driver software uses a simple graphical interface, allowing DA9053 to be controlled via a USB port of a PC. The Evaluation Board:

- has been designed to accept the device either in a socket (U1) or soldered directly to the board,
- has a large number of jumper links to provide access to many configuration and measurement test points, but few of these need to be altered for most normal operations.

4 Hardware

The hardware solution is based upon PCB numbered 44-179-115-01-B or 44-179-115-03-B.

The software uses a PC operating Windows 2000/XP/Vista/Windows 7 with a USB1.1 or USB2 interface.

The DA9053 device plus the USB Interface consume approximately 35 ma in the standby state. The USB cable should be plugged directly into a 500 ma capable USB port, since it may consume more than 100 ma in operation. It is not guaranteed to operate in a hub, see the section 4.1.

The software permits configuration of the device using one of several pre-prepared templates, write and read operations to all control registers, and provides monitoring of device status.

The DA9053 Evaluation Board comprises the DA9053 device, a USB-I^2C bridge for communication with the device, and a few external active components to reduce the requirement for external equipment.

Notes and limitations:

- The PCB is primarily intended to allow evaluation using the device mounted in a socket (max contact resistance 50 mOhms). *This socket imposes a limit on charger current available. This has been limited at 900 mA*. An alternative reference PCB is available with a soldered sample.
- The use of a socket may preclude measurement of detailed device parameters such as maximum load, load regulation, and noise.
- As the socket impedance and trace lengths affect the operation and stability of the buck converters, tantalum output capacitors have been used on the evaluation PCB instead of the usual ceramic devices. This will give rise to a larger than usual ripple voltage on the buck converters, which will not be present in a soldered PCB.
- To minimize inductance the passive components needed by the LDO regulators have been placed underneath the PCB, as close as possible to the socket pins.
- The board has been designed to use a minimum track width and spacing of 3 mil (0.75 mm).
- Gerber data for the board is available on request.
- All power outputs are available as pairs of pins, to permit true Kelvin sensing of output voltage under load conditions. Common impedance has been minimized apart from the socket resistance.
- All control inputs and outputs to the device are available for direct connection by removing the appropriate jumpers.

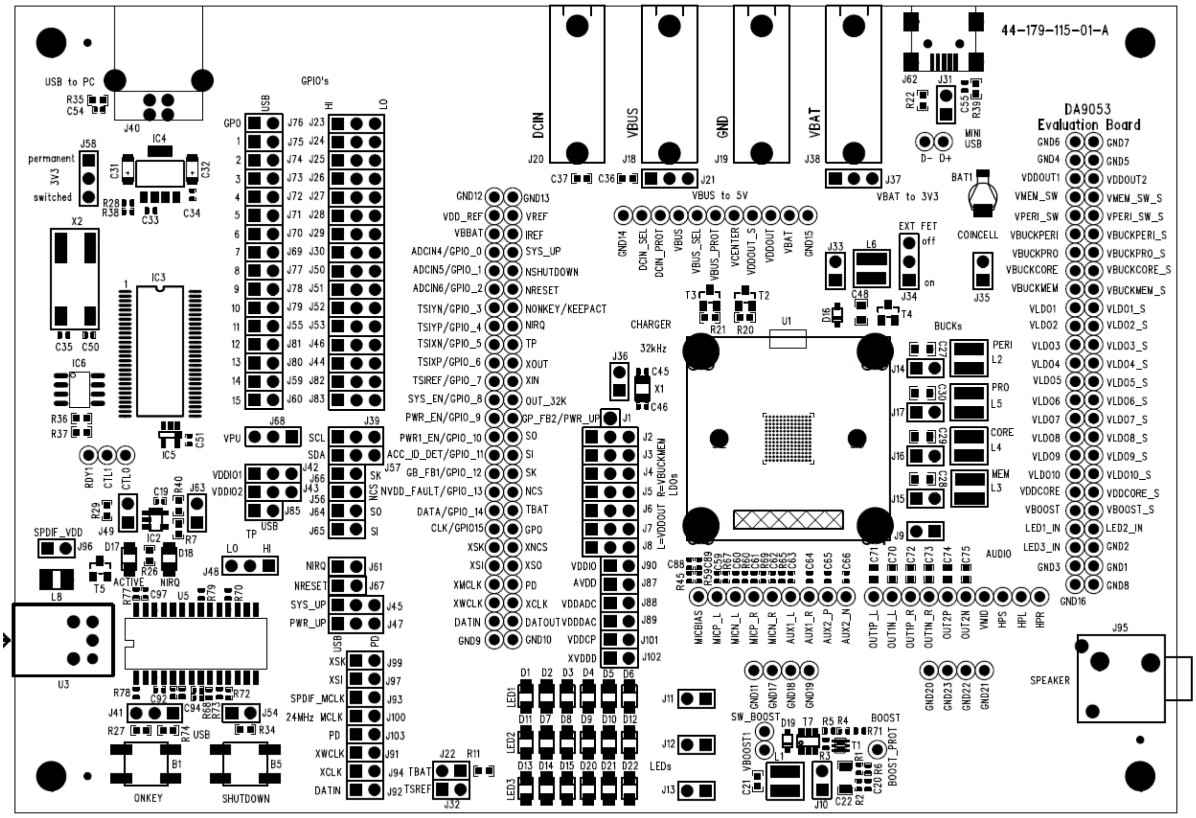


Figure 1: Evaluation Board Assembly Diagram

4.1 Power Supplies

The board is intended to be supplied by a single 3.6 V supply (nominal) plugged into the 4 mm sockets VBAT and GND. If charging operation is to be examined, a Lithium Ion battery may be used, or a “sourcemeter” instrument, such as Keithley 2400, Hameg 8143 or similar, which needs to be used with source and sink current capability.

Certain devices on the board will be powered from the 3.3 V supply produced by the USB interface regulator, such as the LEDs, the USB device itself, and the VDDIO voltages.

⚠	CAUTION
	For demonstration purposes, the 3.3 V supply can be connected to VBAT via link JP37 to allow the use of the Evaluation Board without a power supply. However, this should be used with some care, as this configuration gives no measure or indication of the current taken, which could be too large for the USB host to support (>500 mA) if certain features are enabled, for example, white LED. If the device has been inserted into the socket incorrectly, the socket or the device could be destroyed. This self-powered operation should not be the default mode of use. If using this mode, ensure that jumper J58 is in position 1 (permanent).

The DCIN and VBUS terminals are available to allow investigation of charging performance if using a Li-on battery or 4-quadrant sourcemeter as VBAT supply.

4.2 Links

Table 1: Device Links

Link	Position 1 (left)	Position 2 (where applicable, right)	Function
J76	To USB		Controls GPIO0 from USB device Note 1
J23	Hi	Lo	Connects GPIO0 to fixed voltage Note 2
J75	To USB		Controls GPIO1 from USB device
J24	Hi	Lo	Connects GPIO1 to fixed voltage
J74	To USB		Controls GPIO2 from USB device
J25	Hi	Lo	Connects GPIO2 to fixed voltage
J73	To USB		Controls GPIO3 from USB device
J26	Hi	Lo	Connects GPIO3 to fixed voltage
J72	To USB		Controls GPIO4 from USB device
J27	Hi	Lo	Connects GPIO4 to fixed voltage
J71	To USB		Controls GPIO5 from USB device
J28	Hi	Lo	Connects GPIO5 to fixed voltage
J70	To USB		Controls GPIO6 from USB device
J29	Hi	Lo	Connects GPIO6 to fixed voltage
J69	To USB		Controls GPIO7 from USB device
J30	Hi	Lo	Connects GPIO7 to fixed voltage
J77	To USB		Controls GPIO8 from USB device
J50	Hi	Lo	Connects GPIO8 to fixed voltage
J78	To USB		Controls GPIO9 from USB device
J51	Hi	Lo	Connects GPIO9 to fixed voltage
J79	To USB		Controls GPIO10 from USB device
J52	**Ground**	Lo	Connects GPIO10 to fixed voltage
J55	To USB		Controls GPIO11 from USB device
J53	Hi	Lo	Connects GPIO11 to fixed voltage
J81	To USB		Controls GPIO12 from USB device
J46	Hi	Lo	Connects GPIO12 to fixed voltage
J80	To USB		Controls GPIO13 from USB device
J44	Hi	Lo	Connects GPIO13 to fixed voltage
J59	To USB		Controls GPIO14 from USB device
J82	Hi	Lo	Connects GPIO14 to fixed voltage
J60	To USB		Controls GPIO15 from USB device
J83	Hi	Lo	Connects GPIO15 to fixed voltage

Link	Position 1 (left)	Position 2 (where applicable, right)	Function
J58	USB powered always	USB disabled when no VBAT	USB power
J21	External VBUS	USB 5 V supplies VBUS	VBUS supply
J37	External VBAT	USB 3.3 V supplies VBAT	VBAT supply
J68	VDDCORE	VDDOUT	Pullup supply
J42	3.3 V USB	VDDCORE	VDDIO1 voltage Note 3
J43	3.3 V USB	VDDCORE	VDDIO2 voltage Note 4
J85	To USB		TP output Note 5
J48	**On**	**Off**	**Power Commander Mode** Note 6
J41	Pushbutton	USB	nONKEY source
J54	Pushbutton		SHUTDOWN source
J47	To USB	Pulldown	PWR_UP output
J45	To USB	Pulldown	SYS_UP output
J67	To USB		NRESET output
J61	To USB		NIRQ output
J65	To USB		SI input
J64	To USB		SO output
J56	To USB		NCS input
J66	To USB		SK input
J57	USB to DATA/GPIO14	USB to SO	SDA input/output Note 7
J39	USB to CLK/GPIO15	USB to SK	SCL input Note 7
J22	10K pulldown		TBAT input
J32	VLDO9		TSREF voltage
J49	100K to VDDOUT		NRESET pullup resistor
J63	100K to VDDOUT		NIRQ pullup resistor
J2	VDDOUT	BUCKMEM output	LDO1 supply input
J3	VDDOUT	BUCKMEM output	LDO2 supply input
J4	VDDOUT	BUCKMEM output	LDO3_4 supply input
J5	VDDOUT	BUCKMEM output	LDO5 supply input
J6	VDDOUT	BUCKMEM output	LDO6 supply input
J7	VDDOUT	BUCKMEM output	LDO7_8 supply input
J8	VDDOUT	BUCKMEM output	LDO9_10 supply input
J11	Connected		LED1 LEDs
J12	Connected		LED2 LEDs
J13	Connected		LED3 LEDs
J10	Connected		Boost inductor
J9	To VDDOUT		VDDBUCK supply
J15	Connected		BUCKMEM inductor
J16	Connected		BUCKCORE inductor
J17	Connected		BUCKPRO inductor
J14	Connected		BUCKPERI inductor
J35	Connected		Coincell connection
J34	In circuit	Disabled	External FET
J33	Connected		Charger buck inductor
J31	Connected		USB detect 200 ohm resistor
J36	Connected		Crystal oscillator

Note 1 Jumper pairs J76/J23 etc. are not normally used at the same time. Either select SUB control, or fixed level, or none.

Note 2 Fixed levels are via 100K resistor to ground, or 100K resistor to VDDIO2, to allow for bidirectional use.

Note 3 Normally set to position 1 for USB control.

Note 4 Normally set to position 1 for USB control -- set to position 2 to test IO level control.

Note 5 Connected to allow USB to detect Power Commander mode.

Note 6 Power Commander Toggle Switch.

Note 7 If set to Position 2, Jumpers 64 and 66 should be removed to prevent logic clash.

Figure 2 shows the default locations of all the jumper links.

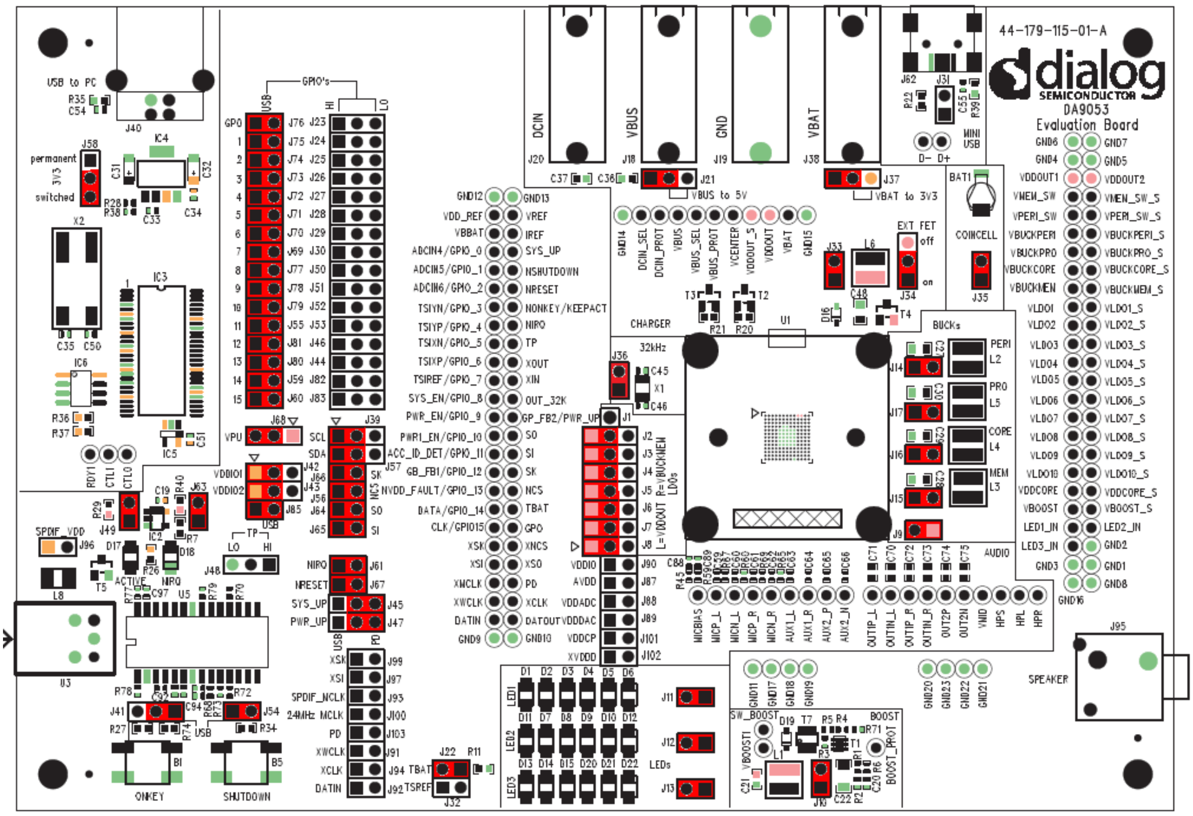


Figure 2: Default Link Locations

5 Evaluation Board Features

5.1 USB Interface

The USB Interface IC3 is used here for two purposes:

1. As a source of I^2C and uWire/SPI control signals, such as SCL.
2. To provide the discrete signals GPIO<x>.

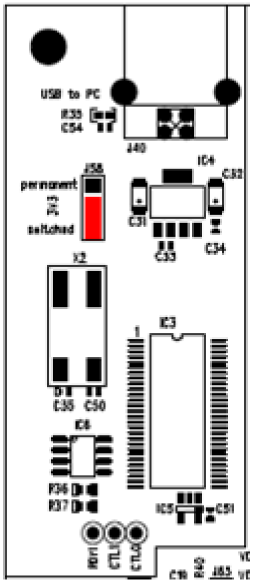


Figure 3: USB Interface

The USB device is powered via the USB bus cable via a fixed 3.3 V regulator.

The USB Interface can be removed from circuit by removing several of the jumpers listed above, to allow direct access to all signals from external circuitry. The USB Interface can also be used to supply the VBAT voltage (at 3.3 V) to the Evaluation Board, although this must be used with some care.

The USB Interface implements multimastering on its I^2C interface, permitting concurrent operation with any other multimastering controller. This allows the software to control a DA9053 device which is already part of the user system, and under control of the system processor.

5.2 Control and IO Signals

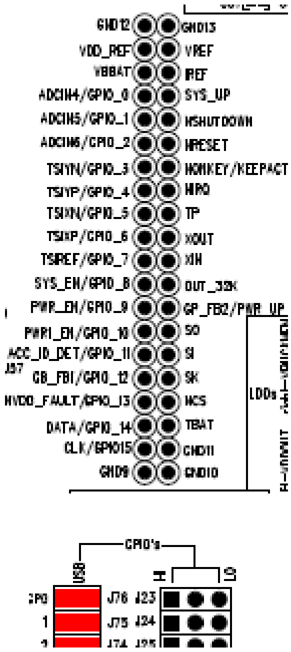


Figure 4: Control and IO Signals

All of the Control and IO signals from the DA9053 appear on the headers pins shown on the left. These are the most useful monitor points for debugging purposes.

If required, a suitable connector can bring these signals to the system board for integrated development. A standard 50 way connector will fit without interference. If used in this way, the other links which also control these pins should be removed to avoid logic or voltage clash.

The arrangement of the jumper links to the GPIO pins allows for a very flexible use of these resources. The jumper may be moved between the left-hand location (USB control/monitor), the right-hand terminals which pull up to VDD_IO2, down to ground via 100K resistors, or left open for external connections.

This is shown in Figure 5.

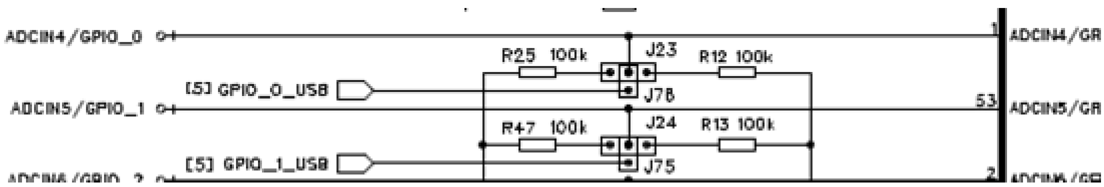
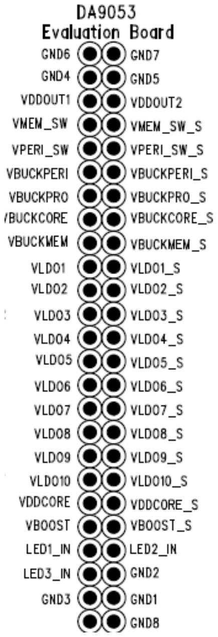


Figure 5: Jumper Movement

5.3 Voltage Outputs



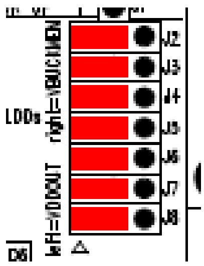
The LDO, buck regulator and boost regulator are available on the pairs of terminal pins shown on the left. They are arranged in pairs to allow meaningful measurements under load conditions.

Each pin is connected separately to the regulator output, to a point as close as possible to the device pin.

The inner connections (VLDOx) are routed with wide PCB traces, while the outer connections (VLDOx_S) are routed with narrow PCB traces. The outer connections are intended as Kelvin sense points, to allow measurement of the regulator outputs without effect of PCB resistance.

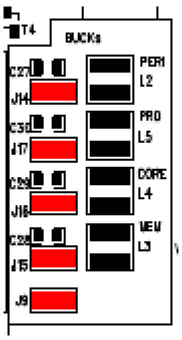
This is not perfect, however, since the socket that the device is mounted in adds about 50 mohms to each pin output, producing a small error when under load.

If required, a suitable connector can bring these signals to the system board for integrated development. A standard 50 way connector will fit without interference.



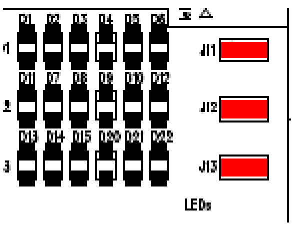
This group of links allow the LDOs to be individually powered from either VDDOUT (default) or the buck regulator BUCKMEM for lower power operation.

To power an LDO from a regulator other than BUCKMEM, the link may be removed and a suitably terminated wire connected to the center pin and the chosen regulator output.



Each buck regulator output has a jumper link in series with the inductor; This allows:

- inspection of the switching voltage and frequency,
- a small wire or resistor to be inserted to measure inductor current. This might be necessary if a different inductor coil were being assessed in the application.



The 3 jumper links shown left allow access to the white LEDs fitted to the board. If desired, external LEDs may be connected instead between VBOOST and the right-hand terminal of each jumper.

Please note the requirements of the specification with regard to balancing of the voltage drop in each string.

6 Software

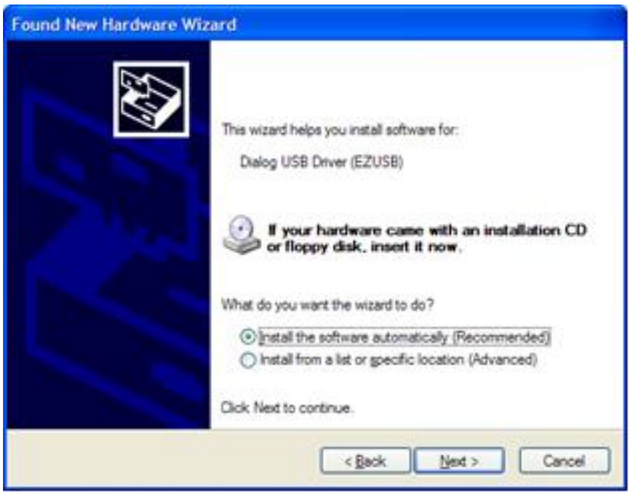
6.1 Installation Procedure

Insert the USB stick containing the software. If the installation does not start automatically, run the program **setup.exe**, found in the **Software** directory on the stick. An automated script installs the program on your PC. By default, the directory:
C:\Dialog Semiconductor\Power Management\Power_Commander_Vxx (version number) is used.

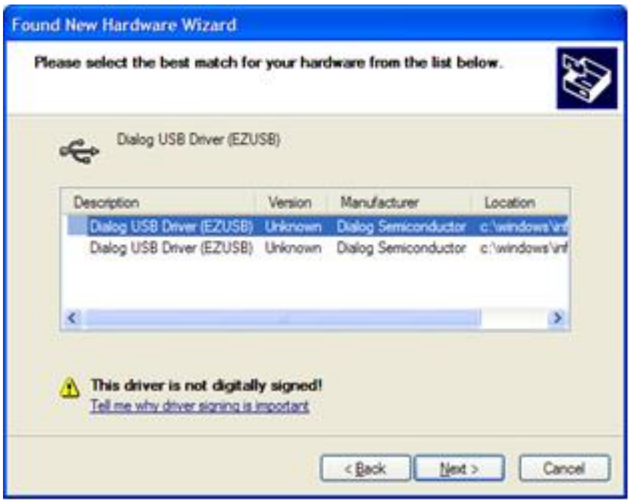
Plug in the USB cable, Windows detects the USB device and prompts for the drivers. These should automatically be located on the root directory of the USB stick. The setup file is Dlgezusb.inf.

The following images show the procedure step-by-step:

1. From the **New Hardware Wizard**, select **No, not this time** and then click **Next**.

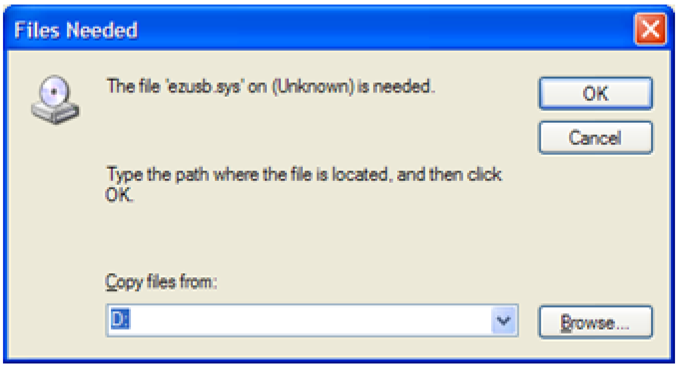


2. Select **Install the software automatically**.



3. Select **Dialog USB Driver** and click **Next**.

4. Click **Continue Anyway**.



5. This step is not always produced, if not continue to step 6. If prompted, select **D:** (or your USB stick drive letter) then click **OK**.

6. Click **Finish**.

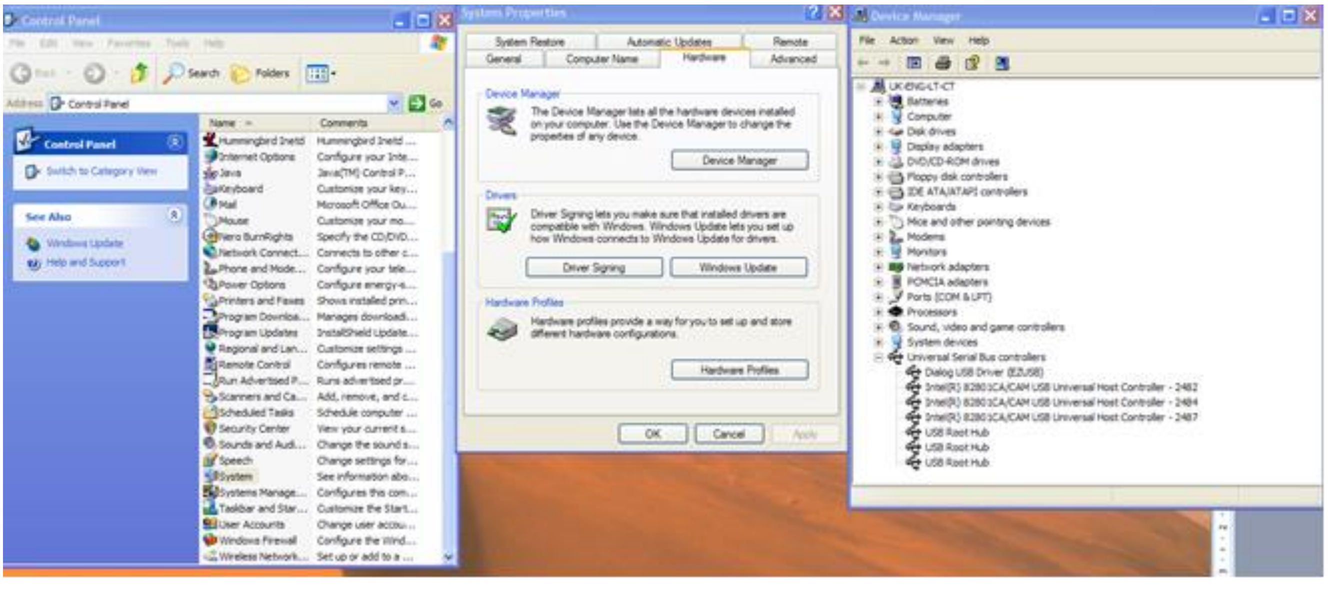


Figure 6: Installation Procedure

Notes:

- If installed correctly, Dialog USB Driver (EZUSB) should be listed on **Device Manager** as shown in Figure 6.
- If you are using Windows XP, you may get a message saying that a USB2 device is attached to a USB1.1 port. This can safely be ignored.
- To uninstall the software please use the Windows **Add/Remove Programs** function that can be found in **Control Panel**.

6.2 Starting Power Commander

Run the DA9053 program by clicking the shortcut from the **Start** menu. The best setting for the PC display size is 1024x768 pixels or above. The font size on the PC display should be normal (95dpi). It is important to note that a display size other than the recommended setting may affect the way in which the tabs appear.

Make sure that the TP toggle switch is in the **left** position to select **Power Commander** mode. The **Initial Interface** appears, with the **USB OK?** LED lit if the USB interface is correctly connected and functioning, as shown in Figure 7.

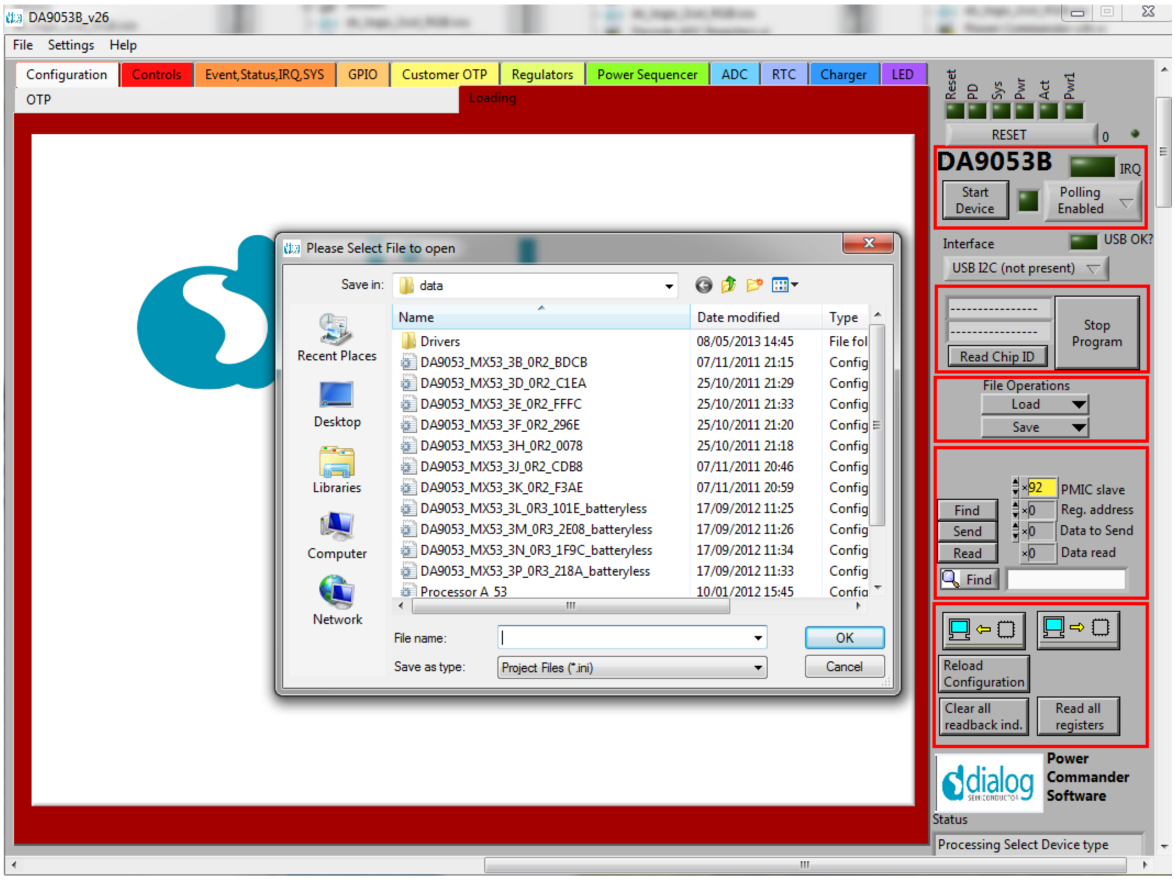


Figure 7: Initial Interface

The device is now in **Power Commander** mode, and is waiting for a project file to be selected. Project files shipped with the software are either live variants of example files. Select one of the files to begin.

The Status LEDs at the top of the screen indicate the device status. If the Reset LED is blinking yellow, it indicates that the device is not yet communicating via the I^2C interface. See section 7 for more details.

If **Autostart** mode is selected in **Settings** (default), pressing the **Start Device** button, which sends the device directly into **Active** mode, sends data and pulses the nONKEY control as necessary. If **Autostart** is not selected, the **Download** button will be visible. Pressing this will progress the device from **Reset** mode to **Powerdown** mode, and depending upon the programmed startup mode, the device may progress automatically to Active mode, or remain in **Powerdown** until a wakeup event is received. Pressing the **Start Device** button toggles the nONKEY control as long as jumper J41 is in its rightmost position. **Start Device** will also commence monitoring of device status, IRQ status and RTC at 1 second intervals, and load the RTC and alarm register with current local time and date. Deselecting **Autostart** allows more options for evaluating wakeup conditions, but is more complicated to use.

The command button **Stop Program** ends the DA9053 program.

6.3 Configuration Panel

This tab is used to select a suitable template file for the processor, allow modifications and store the modified file as a new project. The aim of the configuration process is to create a group of setup conditions which will be permanently programmed into the production device to allow a startup configuration which is perfect for the processor and all associated peripheral devices.

This includes definition of logic interfaces, LDO and buck startup voltages and timing, ADC alarm limits, startup conditions, and so on.

NOTE
Please note that ONLY changes made from this Configuration panel are stored in the Project file (project.ini). Changes made to other registers on different tabs are NOT stored in the file and are considered temporary.

After loading a project, a temporary project file temp.tmp is made, and all changes are stored there. The temporary file can then be saved as a new or existing project using the **Save** menu.

The supplied template files are read-only so that they are not inadvertently overwritten.

A project file, which is a standard Windows ini file, can be imported into the program by copying it into directory c:\Dialog Semiconductor\Power Management\Power_Commander_Vxx\Data. The file can then be opened inside the program from the **File**, **Open** menu.

It is not recommended to edit the ini file using Notepad or any other text editor.

If the LED next to the device selection is amber, it indicates that there has been an unsaved change or changes to the device configuration. Select **File**, **Save project** or **Ctrl**+**S** to save the changes.

Pressing the **Project Select** button produces a list of recently used projects with their full path names, see Figure 8.

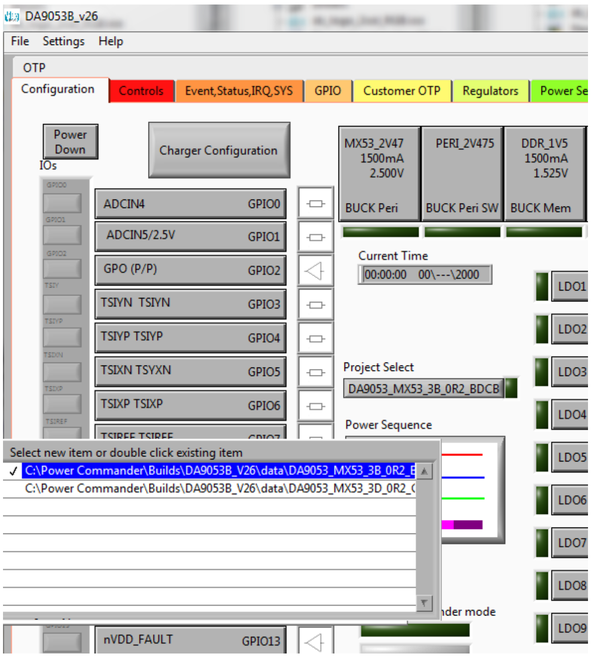


Figure 8: Project Select Dialog

A double-click on the current project reloads its data.

6.4 LDO Configuration

From the **Configuration Panel** click on the text of one of the LDO regulators to open the **LDO Configuration** dialog as shown in Figure 9.

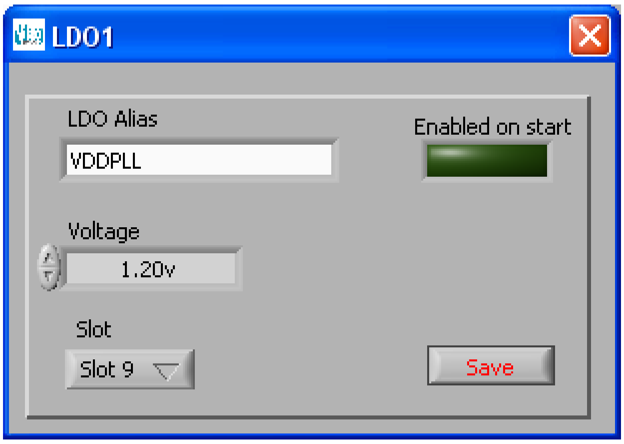


Figure 9: LDO Configuration Dialog

The LDO Configuration dialog controls are:

LDO Alias (optional) This is an alternative name for the regulator to be known by. This is an aid to visualizing the connection of the device to the system. The alias appears in the Waveform screen also.

Voltage This is the target output voltage of the regulator.

Slot (optional) This is the timing of the start of the LDO on powerup. It can be configured here, or in the Waveform screen.

Enabled Allows the LDO to be enabled immediately on startup (not under Power Sequence control).

Save Stores the changes in the project file and sets the appropriate control on the panel (Note: changes are **not** transmitted to the device at this time, only when **Start Device** is executed after powering down the device to the Reset state).

 Exits **without making any changes**.

On exit, the **Configuration Panel** is updated with the LDO number, the maximum current, the programmed voltage, and the LDO alias.

6.5 GPIO Configuration

From the **Configuration Panel** click on the text of one of the GPIO controls to open the **GPIO Configuration** dialog.

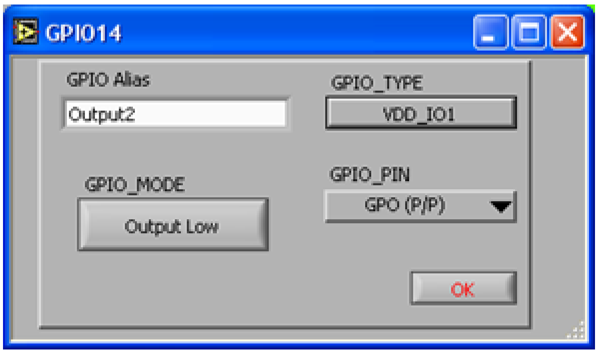


Figure 10: GPIO Configuration Dialog

The GPIO Configuration dialog controls are:

GPIO Alias (optional) — This is an alternative name for the GPIO to be known by. This is an aid to visualizing the connection of the device to the system.

GPIO_PIN — This is the type of the GPIO. Each GPIO has a primary function, either analog or digital, and a general purpose input (GPI) or general purpose output (GPO push-pull, or GPO open-drain) functions. The GPIO_TYPE, and GPIO_MODE control will change depending on the selection.

GPIO_MODE — If the GPIO is configured as a GPO, this will set the output to be low or high by default, If the GPIO is configured as a GPI, it will enable or disable debouncing and Wakeup event.

GPIO_TYPE — If the GPIO is configured as a GPO, this will set the output voltage to either VDDIO1 or VDDIO2. If the GPIO is configured as a GPI, it will set the active edge for a wakeup event.

OK — Stores the changes in the project file and sets the appropriate control on the panel (Note: changes are **not** transmitted to the device at this time, only when **Start Device** is executed after powering down the device to the Reset state).

 — Exits **without making any changes**.

On exit, the basic configuration is shown by the text string, and the icon changes to one of:

- Analog (a resistor symbol)
- GPI (a logic input facing right)
- GPO P/P (a logic input facing left)
- GPO O/D (a MOSFET symbol facing left)

6.6 Buck Configuration

From the **Configuration Panel** click on the text of one of the Buck controls to open the **Buck Configuration** dialog.

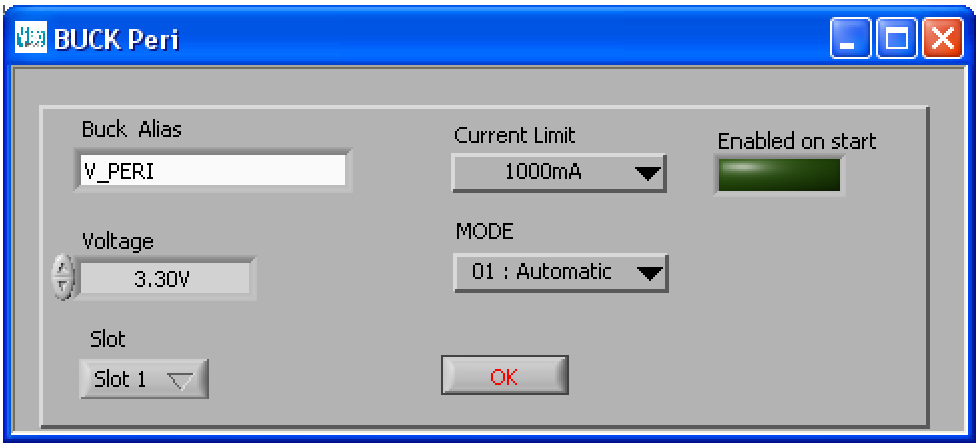


Figure 11: Buck Configuration Dialog

The Buck Configuration dialog controls are:

Buck Alias (optional) — This is an alternative name for the regulator to be known by. This is an aid to visualizing the connection of the device to the system. The alias appears in the Waveform screen also.

Voltage — This is the target output voltage of the regulator.

Slot (optional) — This is the timing of the start of the buck on powerup. It can be configured here, or in the Waveform screen.

Enabled — Allows the buck to be enabled immediately on startup (not under Power Sequence control).

Current Limit — Sets the internal current limit of the inductor current. This should be set to match the maximum specified saturation current of the inductor used. It is not the same as the maximum output current of the buck, but is related to current ripple amplitude.

Mode — Allows the buck to operate in low power Pulse Frequency Modulation (PFM), high power Pulse Width Modulation (PWM) or automatically switch between the two. Automatic force PWM mode and allows faster switching between the two modes.

OK — Stores the changes in the project file and sets the appropriate control on the panel (Note: changes are not transmitted to the device at this time, only when **Start Device** is executed after powering down the device to the Reset state).

 — Exits **without making any changes**.

6.7 Buck Switch Configuration

These are controlled switches that allow peripherals to be independently controlled from devices on the main buck outputs.

From the **Configuration Panel** click on the text of one of the buck controls to open the **Buck Switch Configuration** dialog.

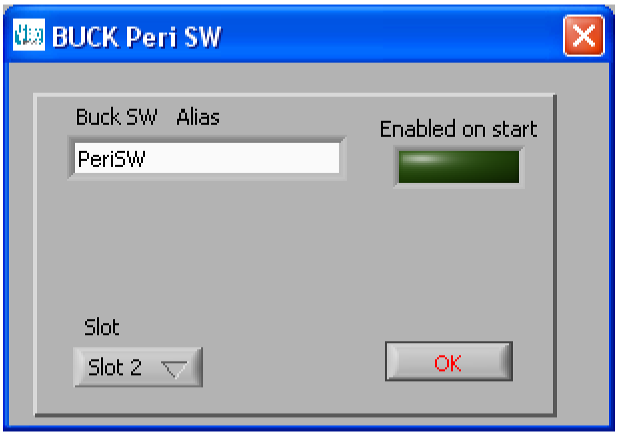


Figure 12: Buck Switch Configuration Dialog

The Buck Switch Configuration dialog controls are:

Buck SW Alias (optional) — This is an alternative name for the switch to be known by. This is an aid to visualizing the connection of the device to the system. The alias appears in the Waveform screen also.

Slot (optional) — This is the timing of the start of the switch on powerup. It can be configured here, or in the Waveform screen. The slot should not be lower than the related buck regulator else it will not have the desired timing relationship.

Enabled — Allows the switch to be enabled immediately on startup (not under Power Sequence control).

OK — Stores the changes in the project file and sets the appropriate control on the panel (Note: changes are not transmitted to the device at this time, only when **Start Device** is executed after powering down the device to the Reset state).

 — Exits **without making any changes**.

6.8 Change Configuration Register

From the **Configuration Panel** click on the **Change Configuration Register** button to bring up the **Control** dialog.

This allows changes to the default configuration of other registers that are not included in one of the other panels. They are copies of the control on other tabs, but the difference is that the value on exit is saved to the Project file. Providing copies of the registers in this way, reinforces the concept that changes made to other panels are temporary, and are the equivalent of host commands.

The **Configuration Panel** is intended to allow entry of the hardware defaults settings.

Future versions of this software will reduce the number of these extra configuration registers, moving them to more useful task GUIs, for example, ADC.

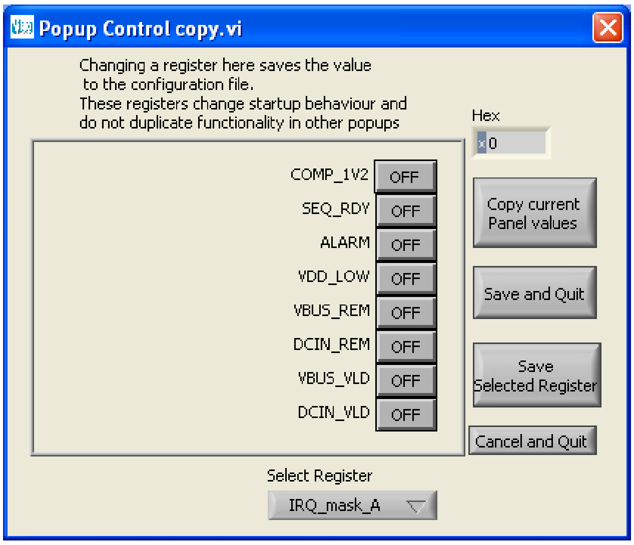


Figure 13: Change Register Control Dialog

The controls available within the Change Register Control dialog are:

Select Register	Allows selection of one of the valid configuration registers. The controls above are populated accordingly.
Register controls	Labels and text of these will depend on the selected register. The **Hex** box indicates the equivalent code for the switch states.
Copy current Panel values	Loads all current panel contents so that they can be saved. This enables faster change to multiple registers by allowing them to be set in the other tabs.
Save Selected Register	Logs the change to an individual register.
Save and Quit	Stores the changes in the project file and sets the appropriate control(s) on the panel (Note: changes are not transmitted to the device at this time, only when **Start Device** is executed after powering down the device to the Reset state).
Cancel and Quit	Closes the dialog and discards any changes made.
	Exits **without making any changes**.

6.9 Power Sequence

From the **Configuration Panel** click on the **Power Sequence** button to bring up the **Control** dialog.

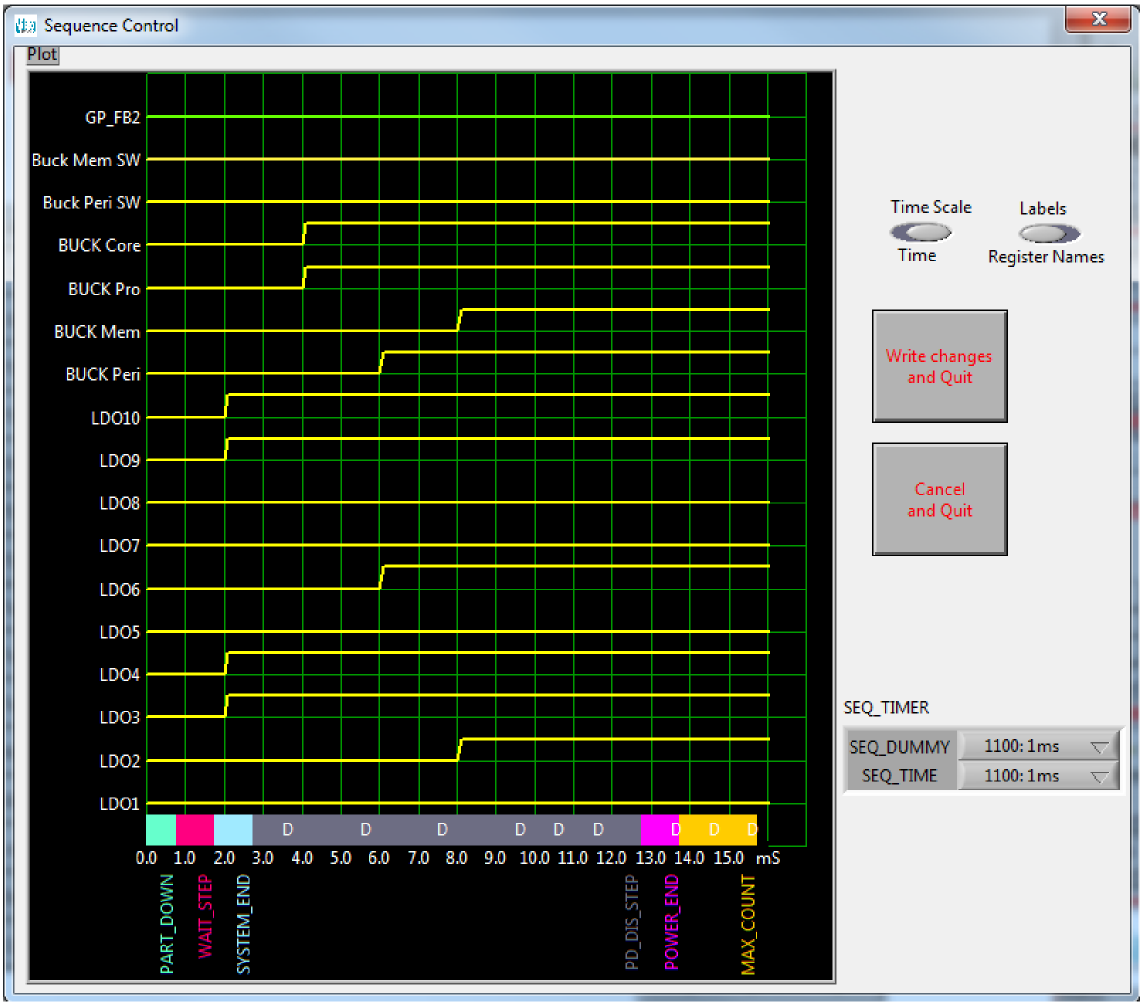


Figure 14: Power Sequence Control Dialog

This graph shows the relative timing of all the LDOs, buck regulators and switches, their placement in the slots and the positions of the control markers. Further information is available in [1].

Changes to relative positions are achieved by clicking the appropriate edge and dragging it left or right.

The control markers (located along the X axis) are changed by dragging the edge of the colored band.
Note that the system markers are shown in the middle of the slot, to indicate that they complete at the end of the slot. It is possible for markers to lie on top of each other.

The Dummy slots are calculated automatically and marked with a “D”. Dummy slots are slots where no programmed transition takes place, and whose duration is controlled by the **Seq_Dummy** timing control.

Note that signal GP_FB2 has 4 transitions possible (2 rise, 2 fall).

Any LDO or buck that is not used should have its transition set to Slot 0.

Any unused regulators are shown as permanently low. Click on any part of the waveform to change this, or drag the rising edge to zero to remove the regulator from startup.

Avoid placing multiple regulators into the same slot as this can cause greater inrush currents.

The controls available within the Power Sequence Control dialog are:

Labels	Switches the labels on the left-hand Y axis between Alias names and regulator names.
Time Scale	Switches the labels on the lower X axis between slot numbers and calculated elapsed time.
SEQ_TIMER	This controls the slot duration for occupied slots and dummy slots independently. The effect can be seen when **Time Scale** is switched to **Time**.
Write Changes and Quit	Stores the changes in the project file and sets the appropriate control on the panel (Note: changes are not transmitted to the device at this time, only when **Start Device** is executed after powering down the device to the Reset state).
Cancel and Quit	Closes the dialog, but does not save the result in the project file.
	Exits **without making any changes**.

6.10 Download

This button writes all the register changes made if **Power Commander mode** is set, and the device is ready for the download. The device signals this by holding pin nVDDFAULT low.

The download is made on the currently activated port of the device (HS2, I^2C, or uWire/SPI Mode 0) overriding any programmed mode until **Power Commander mode** is exited.

If the **Autoboot** configuration is selected, a second download is made to place the device in Active mode, else it will wait in power down mode for a wakeup event. A second download is then necessary to reach Active mode.

6.11 Charger Configuration

From the **Configuration Panel** click on the **Charger Configuration** button. Note that this screen is temporary and will be hidden once access has finished. It is only accessible via this button and not via the tabs. The reason being to reinforce the concept that the panel sets the *initial* value of registers that are stored in the OTP. Changes to these registers will prompt a save to the configuration file on the **Configuration panel**.

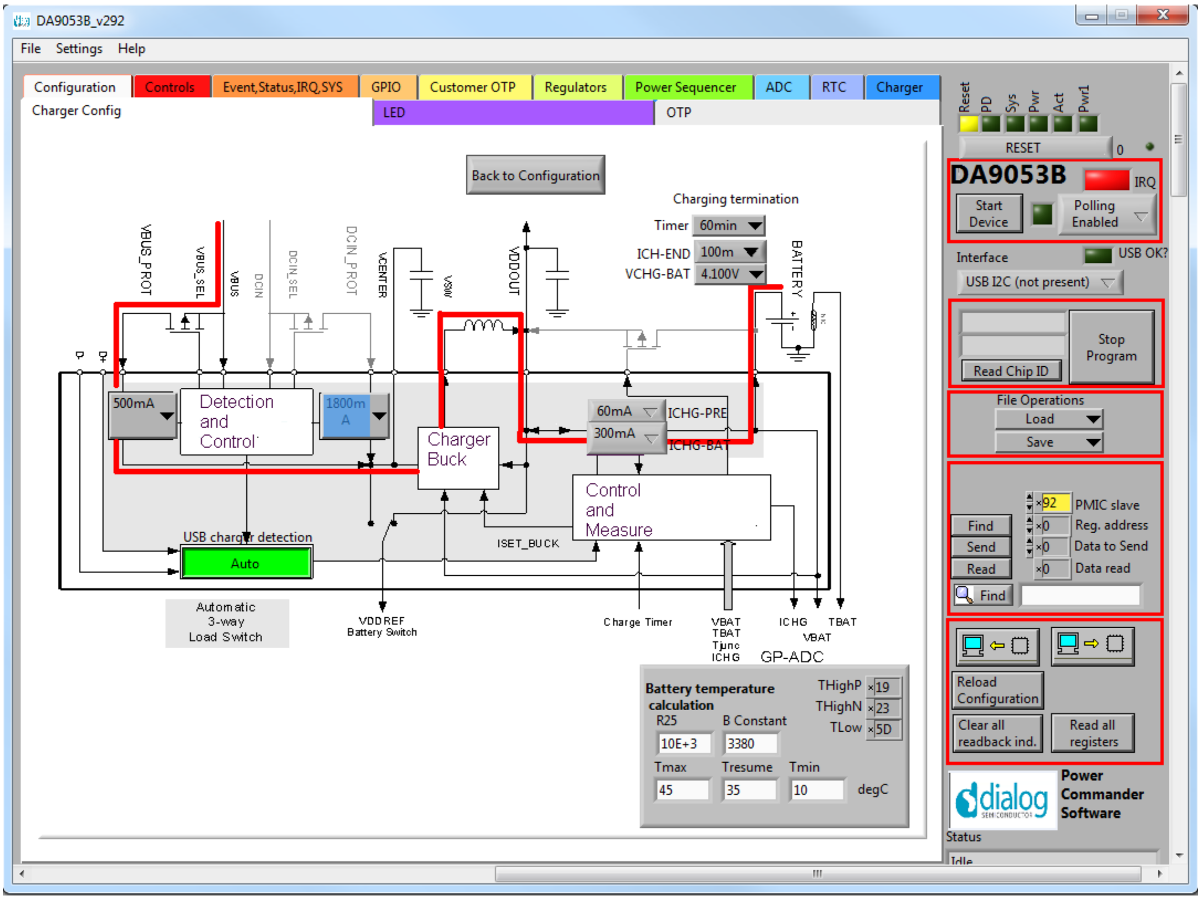
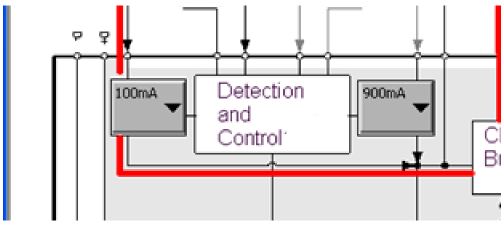


Figure 15: Charger Configuration

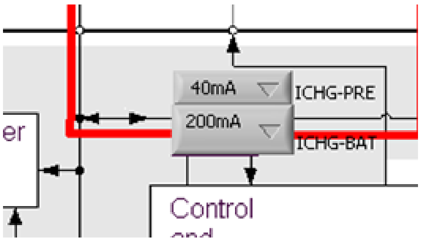
The panel shows a simplified schematic of the charger circuitry, with the power path from an external USB charger highlighted. The function of the appropriate controls can be seen with reference to this path.

A panel is also provided to calculate the settings for battery protection by measurement of its temperature by the inbuilt NTC resistor.

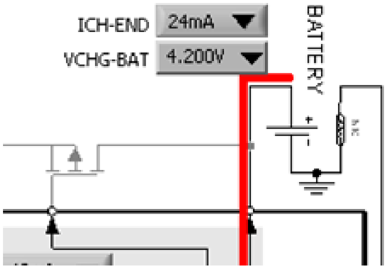


These two buttons control the *input* current limit for the USB and DCIN inputs respectively.

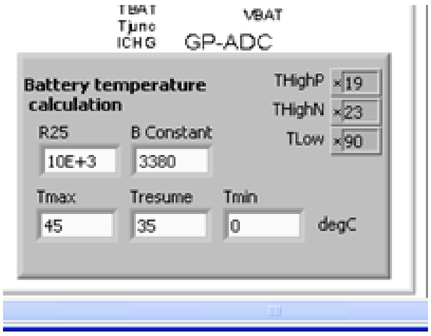
Note that these are limited in Power Commander to 900 mA each simply to protect the socket from excess currents. See Dialog Use in Table 2 for details.



These two buttons set the initial battery charging current in precharge and fast charge respectively. Note again that the fast charge current has been limited to 900 mA for socket protection.



These two buttons set the terminal current and voltage respectively. Charging will complete when measured battery current is below the ICH-END set point while voltage is above VCHG-BAT.



This panel allows calculation of suitable values for the battery temperature protection function. This is based upon measurement of the NTC resistor contained within the battery pack. The measurement is made automatically by the GPADC of the voltage across the NTC when excited by a current of 50 uA.

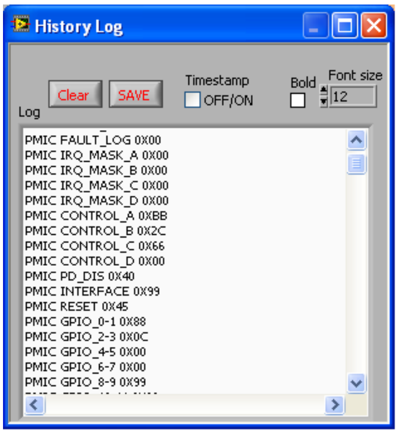
The NTC specifications: resistance at 25 °C and B constant, which describes its gain, are entered into the appropriate boxes. The required trip temperatures Tmax, Tresume, and Tmin are then entered in degrees Celsius.

The values 45 °C for stopping charging, 35 °C for resumption of charging, and 0 °C for low temperature protection, are typical.

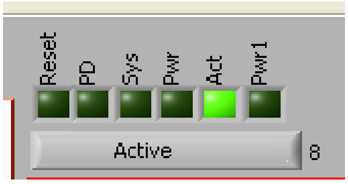
6.12 Menu Items

Table 2: Menu Items

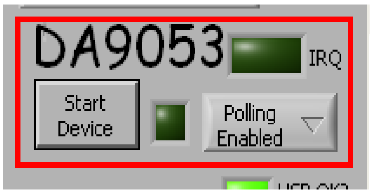
Menu/Submenu	Description
File->	
New Project	Creates a blank project, not using a template, with all registers set to zero.
Open Project	Selects a project to open. Right-clicking on a filename allows editing, copying, and pasting. Otherwise this option is the same as selecting a project from the dropdown Device selector.
Save Project	Save the current temporary file with a new name, or replaces an existing filename.
Delete Project	Deletes the selected project. This should be used with some care, and is included to allow cleanup of several trial versions.
Check Project	This verifies that the project configuration file is complete. This includes all registers that the user may not to define an initial condition ("Don't care"), and GP_ID registers. If any are listed as missing, the most convenient method to include these is as follows: Close the message popup. Select **Change Configuration Register** control on white **Configuration** tab. Select **Copy Current Panel Values**. Select **Save and Quit** From the **File** menu, select **Save project**.
Check Trim	This checks to see if trim values are stored in the device. Sample are normally delivered tested and trimmed. If this is the case, a file "trim.txt" will be generated which will ensure that correct trim values are used instead of default values for all subsequent operations. This ensures that best accuracy is achieved during evaluation. *It is recommended that this operation is repeated if a new sample device is used.*
Exit	Closes the program. If there are unsaved changes a dialog box is displayed.
Settings->	
Autostart	Allow single button startup of the device.
Binary Indicators	Changes all indicators from Hex to binary mode if ticked.
Dialog Use->Test Pages	Allows view of internal test pages (only for Dialog Semiconductor use). A password is necessary for access.
Dialog Use->Ignore NVDDFAULT	Allows software to operate if NVDDFAULT signal is not available on some external PCBs (only for Dialog Semiconductor use).
Dialog Use->Enable High Currents	As the socket on the EVB will not support currents greater than 900 mA, charger settings are not available using this EVB. If an external soldered device is in use, with a suitably sized inductor, this item will allow access to the higher currents.
Reg Names in File	If activated, uses register names instead of numbers in text files. See Appendix A.

Menu/Submenu	Description
History Log 	This enables a popup dialog which will log all manual and file activity. This dialog is extremely useful to capture operations for later scripting, or to verify and understand schematic operations on the codec. This log is interactive. Comments can be added using "//" either as separate lines or inline. Lines can be deleted or modified. The entire contents can be cleared or saved to a file. The log also receives the results from read operations in a text file. See Appendix A. The History Log window can be resized and repositioned, and the font size and bold type can be selected. A timestamp is optional.
Help->	
Show Context Help	Opens a floating information which updates with information on each control under the mouse cursor. This should be left enabled by default to receive information and comments on all registers.
User Guide	Opens a PDF copy of this document (a suitable PDF reader must be installed).
About	Displays version information and contact details.

6.13 Status and Controls



Device Status information indicates the current mode of the device. A flashing yellow indicator on Reset indicates that the status is invalid, probably due to communication not being established yet. Device status, ChipID, IRQ status, and RTC are read at 1 second intervals.



Start Device
This button initializes the device (by pulsing ONKEY low, with J41 in its rightmost position) and providing the data downloads automatically.
At the end of startup, a file "Host_configuration.txt" is loaded and ran to emulate the host processor writing immediately to the device.

Polling Enabled

If pressed, Chip ID, IRQ status and RTC readback are suppressed. This is used to force the communication over the bus to be silent. If this is set to *automatic*, the program will only poll the device while the application is the topmost window. If obscured by another program or window, polling will be disabled.

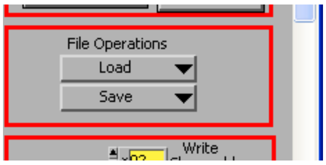
LED
IRQ

If device is active this is green, else red.
This is an indicator of interrupt status, visible from all tabs. Pressing this LED clears the interrupt events.

Indicates version and trim status when device is active. When inactive, version and trim status will not be correct.

Stop Program

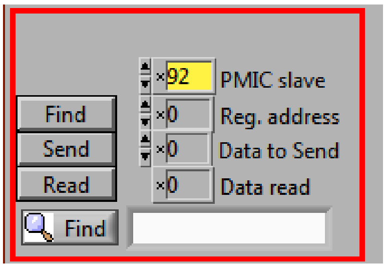
This terminates the program but leaves it inactive on the screen. If there are unsaved changes a dialog box is displayed.



Load: Loads previously saved text file, send all Registers and read back all registers.
or
Save: Save current panel state to text file. Selecting **Register Dump** option saves current register values to the text file. See Appendix A.

Note the difference between **Save** and **Register Dump**. **Save** dumps the contents of all panel controls to the file (a save state operation), while **Register Dump** reads the device contents (including status registers) into the file.

Interface: Select between USB I^2C control, offline mode and uWire/SPI. Switching to offline, then back to USB reinitializes the USB interface.
USB OK?: Indicates that the USB is OK and communicating.



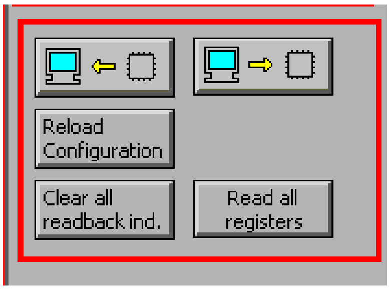
Slave Address: Set slave address of device. This affects all I^2C communications. In Power Commander mode this should be 0x92. In normal mode it should match the port in use (Register INTERFACE).
Note that this is the 8bit value (92h for Write, 93h for Read).

Send: Send a single byte data to I^2C device using Slave Address, Register Address, and Data to Send.

Read: Read single byte data from I^2C device using Slave Address, and Register Address.

Find: Find a control matching a full or partial register name, a control bit name, a register number (e.g. R23 or 17h). Pressing **Find** repetitively will step through all matching items.

NOTE
If Device Address does not match the port numbers on the device, this can be used to control/read any other device on the I^2C bus.



Synchronize Panel from Device
Reads all the register contents of the device and updates the panel to match.

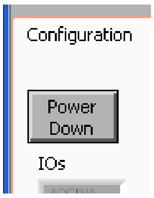
Synchronize Device from Panel
Writes all the device registers to match the panel. (Refresh operation)

Reload Configuration
Resets registers to values specified in configuration file.

Clear all readback indicators
Sets all readback indicators to 0

Read All Registers: Read all registers, comparing with the panel controls

6.14 Discrete Controls



Power Down
Invokes a Power Down by sending Deep_Sleep Register bit (CONTROL_B bit 6) to the device.

The controls on the Configuration tab, and repeated on the GPIO and Event tabs, indicate the discrete logic levels to and from the USB interface towards DA9053.

The grayed out controls are indicators. They show the level at the interface but do not drive it.

The control is only active when the GPIO is programmed as a GPI input, or the default mode is an input.

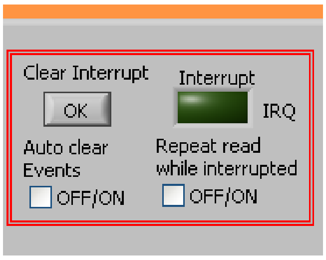
The names of the controls are read from the Alias names of the GPIO, or the GPIO number if no alias is present.

The controls will only have an effect if the appropriate jumpers on the board are in the default positions.

6.15 Event Status

This tab indicates the status of the device, wakeup and interrupt events, and provides the ability to mask events from causing an interrupt.

Note that masking an interrupt does not prevent the event from producing an entry in the being flagged. It is assumed that the application software will gate events with the mask register.

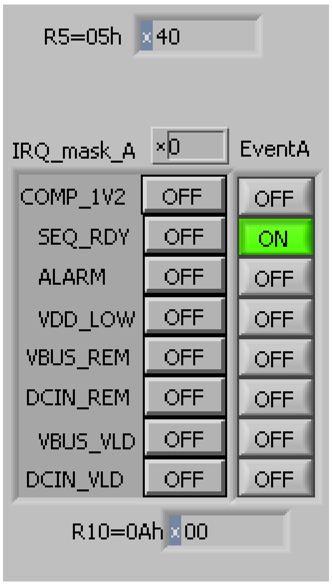


Interrupt indicator
This will not be active if polling is disabled.

Pressing **Clear Interrupt** writes all 1s to the Event registers to clear the interrupt.

If the **Auto clear Events** control is set, this is done automatically. The Interrupt pulse will be very short, but the system is ready for interrupt without manual intervention.

If the **Repeated read** control is set, the software will read event and status registers repeatedly while the **Interrupt indicator** is active.



Event Register, labelled as Register number and its hex equivalent.

Hex equivalent of mask register bits

Name of mask register bits, and also event names

Event bit set. Press to clear and reset.

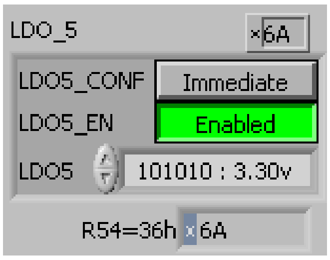
Readback of mask register, visible as hex or binary by using **Binary Indicators** in the **Settings** menu.

6.16 Control Tabs

The tabs GPIO, Customer OTP, Regulators, Power Sequencer, ADC, RTC, Charger, and LED all have the same format.

Each register cluster comprises a control with a mixture of Boolean toggle buttons, multivalue ring controls, or slide controls; a hex indicator showing the total equivalent value, a readback indicator showing the current contents of the register. The readback indicator is labelled with the register number in both decimal and hex.

Readback indicators can be switched individually by clicking on the “x” to decimal, octal, hex or binary, or they may all be changed at once between hex and binary by the **View**, **Binary Indicators** menu item.



Hex indicator

Boolean control Bit 7
Boolean control Bit 6

Ring value control Bits 0-5

Readback indicator

Changing a register control immediately sends the value to the selected register, and reads the value back again, comparing the result with the hex indicator. Note that all bits of the register are sent at once. Therefore, this does not allow changing multiple bits simultaneously.

If the readback indicator is red, it indicates that the current value does not match. This might be because the value has not been downloaded yet (for device selection), or because the supply has been interrupted.

NOTE
Changes made to registers on these screens are not saved into the project file, they are purely temporary changes.

6.16.1 OTP Programming

The One Time Programming (OTP) registers in DA9053 have been left blank to allow the user to set default startup and voltage settings.

The devices are tested and trimmed. The software can interrogate the device to ascertain whether trim values are present (by means of **Check Trim** in **Settings** menu). These trim settings will be used for all subsequent operations, giving best accuracy. If another device is used, repeat the operation.

⚠ CAUTION

Please note that these are One Time Programming registers. Once set they cannot be erased. If mistakes are made the sample may not be usable and may need to be discarded.

The existence of Power Commander software mode makes programming of OTP redundant during much of the System. Programming the OTP should be the final step in verifying that a programmed part will achieve the system objectives, and that it is intended to solder the programmed part onto the final system board.

It is necessary to start up the DA9053 Evaluation software with the board powered and a connection to the PC via the USB cable. A 7.5 V Supply should also be available to connect to the VBUS pin of the board to program the OTP.

The Evaluation board should be powered from an external power supply set to 3.5 V applied to terminal VBAT during OTP programming procedures. We do not recommend self-powering of the board due to the high risk of incorrect device insertion.

If it is decided to continue with OTP programming, the following procedure should be used.

1. From the **Configuration Panel** click the **OTP** tab.

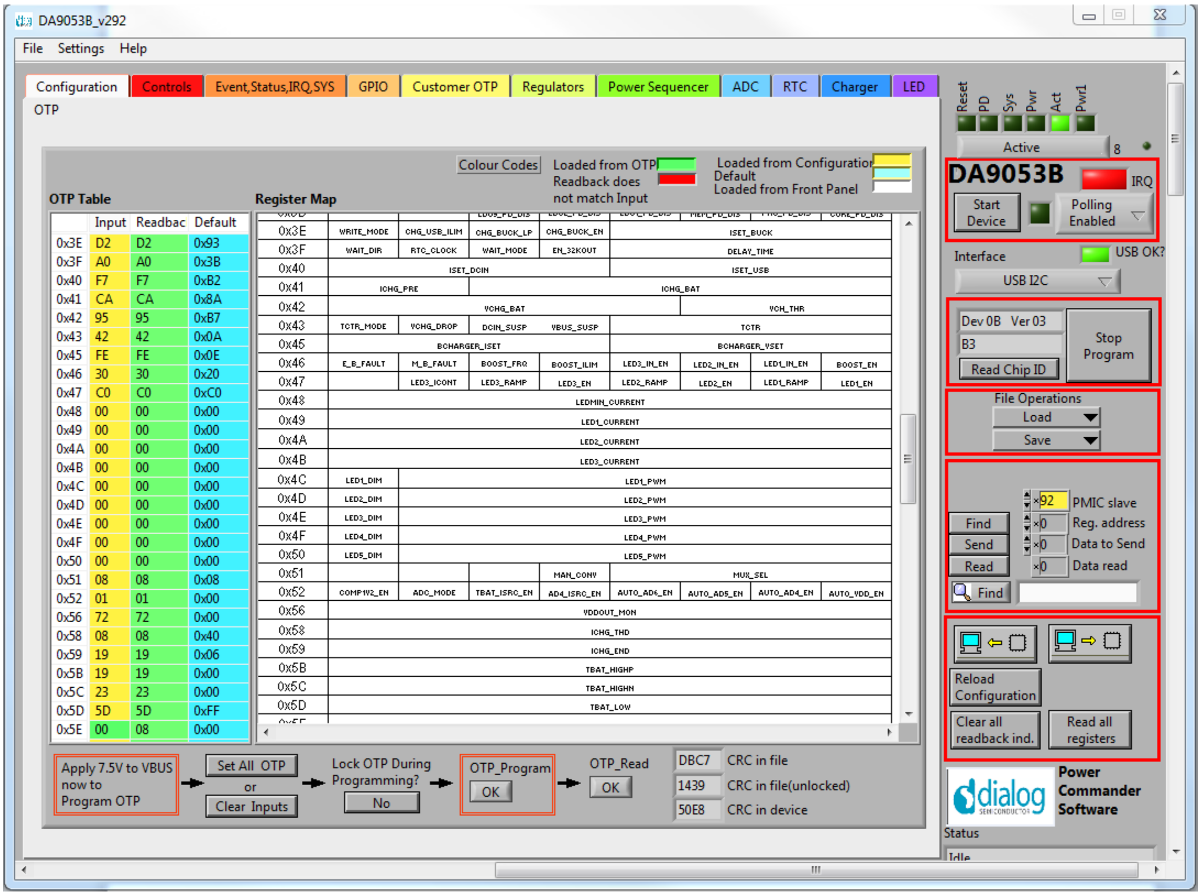


Figure 16: OTP Tab

2. Apply 7.5 v to VBUS.

NOTE
The device should be in **Power Commander** mode with the red switch in the left hand position.

3. Press **Start Device** to initialize the device into the active mode.
4. Press **Set All OTP**.

NOTES

This will read the configuration file, load the registers and apply defaults to the other registers (trim and internal configuration). The **Readback** column will turn green to indicate the current contents (if any) of the OTP, not the register contents.

To Lock the OTP and prevent further modification set the **Lock OTP during Programming?** button. This also requires that the OTP_CONF_LOCK bit is set in R131.

The OTP_CONF_LOCK bit is treated in a special way, it is only read from OTP following a POR. This means that immediately after programming, this bit will not show the programmed value. The GUI takes this into account when verifying that the programming has been successful. Immediately after programming **CRC in device** matches **CRC in file(unlocked)**. After power cycling the device it is possible to perform an OTP read and verify that **CRC in device** now matches **CRC in file**.

The numbers in the second column, labelled **Input**, are the register results of the device loading performed previously.

Verify that these values are correct by reference to the specification. If not, and the reson is unclear, it may be necessary to send the configuration file to Dialog Semiconductor Applications for debugging assistance.

5. Press the **OTP_Program** button once.

If the programming has been successful, a green popup appears indicating correct programming. The value in box **CRC in device** matches the value in **CRC in file**. If it has not been successful, a red popup appears. Check first that the 7.5 V supply is present.

If so, please compare the **Input** and **Readback** columns of the table to try to ascertain the reason. Any differences will be highlighted in red. You may need assistance if the reason for the failure to program is not obvious. All **Readback** values should be green at this point.

A few values will be light green, and have a value different from the **Input** column (registers 0x28, 0x81 in particular). This is because the device does not hold static values in the registers and are dependent upon operating mode. Light green is still a matching value and is normal.

If any register readback values are red, it indicates that the programmed value does not match the desired value in the input column. This will normally be due to one of two reasons:

1. The device has been previously programmed with a configuration different to the currently selected one.
2. One of the registers has been programmed twice, possibly during incremental programming.
   a. Click the **OTP Read** button to allow re-reading of the registers.

6.16.2 OTP Programming of Further Devices

Having successfully followed the procedure in section 6.16.1 you can continue to program another device using the following sequence:

1. Remove 7.5 V VBUS supply.
2. Remove 3.6 V VBAT supply.
3. Remove programmed device.
4. Insert new blank device.
5. Apply 3.6 V VBAT supply, ensure current is <10 mA.
6. Apply 7.5 V VBUS supply.
7. Click **Start Device**.
8. Select **Set All OTP**.
9. Click **OTP_Program**.

Under some circumstances it may be desirable to perform this procedure incrementally, for instance, when all registers do not need to be programmed at the same time.

It is possible to load only a limited number of registers. If the input value is zero no change will be made.

To do this, select **Clear Inputs** instead of **Set All OTP**. This will set all registers to zero. The device may power down, but it will maintain communication and can be programmed.

Type the hex value for the register to be changed into the appropriate line on the table. If the OTP cannot be programmed to this value, due to existing content, the **Readback** box will turn red and indicate the value that will be achieved.

NOTE
A value of **1** cannot be reprogrammed to a **0**.

Press **OTP_Program** and OTP_Read will be performed automatically.

You will see the latest content of the OTP, which should include the modified value.

7 Troubleshooting

This section is an aid to resolving problems occurring in the previous sections.

7.1.1 Software Issues

The USB device should install without difficulty automatically. Make sure that the installation finds and uses the driver contained on the CD.

If the program is started before the USB Interface board is plugged in, the program will default to the offline mode. This can be useful for learning about the software in a desk environment without the hardware attached. If the board is subsequently attached, move the Interface control to **USB**. Make sure the USB is connected and then restart the program.

The software can have unpredictable effects when used in conjunction with a USB hub. It is recommended that a direct connection is made to the USB Interface board.

The software is optimized for a display screen size of 1024 by 768 pixels or greater, with fonts set to normal (96dpi).

There have been reported issues of unpredictable display effects when large fonts (120dpi) are used. This can be changed by right-clicking on the desktop, select **Properties**. Select the **Settings** tab, select **Advanced**, then **Normal size** from the drop-down box.

If communications are apparently lost, first click the **Start Device** button. This attempts to make the device go active. Also switching the **Interface mode** to **Offline**, then back to **USB** can reinitialize the USB interface. Alternatively, unplug the USB then reconnect so that the software detects it and reinitializes.

7.1.2 Hardware Issues

Care should be taken to ensure that the charger voltage is within limits to control power dissipation in the device whilst investigating charger performance.

The board should not be powered from VBAT while the USB cable is disconnected. This can cause unexpected currents to flow in protection diodes of the USB device, possibly causing damage.

Most hardware problems can be traced to incorrect jumper positions. Carefully check jumper positions, compare them with the default positions on Figure 2. Use the jumper table details and the board schematic as a guide to the jumper functions and locations.

Many problems can be traced to the Power Commander switch position. This should be in the left hand position for **Power Commander mode**. Also check for position of jumper J41 (nONKEY jumper).

Power supply configuration can cause issues if incorrect:

- If the link J37 is set to the right (VBAT>3.3 V) **and** VBAT is applied also, one of the supplies will win, although which one cannot be predicted.
- Link J58 can cause some confusion. The intention of this link is to control the 3.3 V regulator used by the USB.

In most instances it is preferable if the USB is deactivated if the device supply is missing. This will mimic the target system operation in which the host processor has no power if the PMIC has no supply. It also avoids unintended current flow between a USB port output and an unpowered GPIO input. However, if the device is to be USB powered, and VBAT is connected to 3V3, J58 will prevent the 3.3 V regulator starting up.

Two solutions are offered for this:

1. Connect 5 V to VUSB (jumper 21 instead). This will lead the 5 V on the USB lead to be treated as a charger with no battery present.
2. Set jumper J58 to permanent position.

⚠	**CAUTION**
Do not apply a charger (either external or via jumper J21) while VBAT is supplied from 3V3 through J37 (in its rightmost position).	

Appendix A Register Text Files

The software includes the ability to save and load a text file containing hex codes representing the register addresses and data. This file is principally used to verify or test a customer's own driver details.

Note that at the end of the startup process, initiated by pressing **Start device**, a file **Host_configuration.txt** is loaded and run to emulate the host processor writing immediately to the device.

If the option **View**, **Reg names in file** is set from the menu, register names, rather than numbers are used in the file, and the slave address is replaced by the word "WRITE". This is generally preferable and more readable. Names are defined in the file Registers_DA9053.h.

Hex codes for slave address and register address are still accepted on reading in the file.

The use of the **File Out** facility is normally to permit register contents to be transferred to the user's own software. Pressing the **File Out** button brings up Figure 17, showing the location and name of the saved file. This is a Register dump of the entire device.

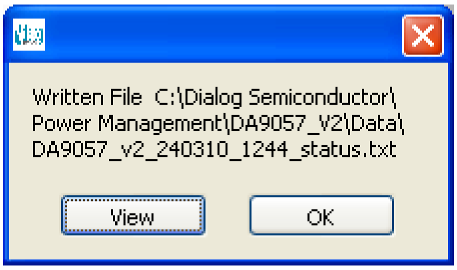


Figure 17: Example File Out Popup

A.1 I^{2}C Register Text File Format

The following format is used for both read and write in the text file:

- Numbers apart from time delays are expressed in hex, separated by tabs. The use of "0x" in front of the hex value is optional.
- The first parameter is the device slave address in 8bit format. *OR*
- The first parameter is a token:
  - WRITE or PMIC will write to the PMIC device at the currently selected slave address (I^{2}C mode only)
  - READ will read from the PMIC device a the values of a number of registers
  - PORT will set the selected digital control line to the specified value (1 or 0)
  - PORTREAD will read the value of the specified digital control line
  - PORTDIR will set the direction of the digital control line to an input if the value is 0, or an output if set to 1.
  - DELAY or WAIT will implement a time delay specified up to 65535 milliseconds. The delay time is specified in decimal, or hex if preceded by 0x.
  - ITERATE will cause the whole script to repeat by the specified number of times.
- The second parameter is the register address as a name or hex value.
- The third parameter is the data.
- Comments (lines beginning with '//') are permitted in the file.
- Inline comments (//comment) are permitted.
- The data will be processed in the order written, and written directly to the specified device. The screen controls will be updated once command in the file have finished.
- The use of the slave address in the file allows any device attached to the I^{2}C bus to be controlled.

- The token PORT will allow control over the GPIOs which are configured as inputs. The second parameter is the port name, either GPIOx or SYSEN etc. Names are defined in the file Hardware_DA9053.h. The third parameter is 0 or 1.
- For read operations, the result of the read is passed to the history log window.

A.1.1 Example File Contents

```
//Write to PMIC in 3 ways
PMIC   LDO10  0x61
WRITE LDO9 0x62
0x92 0x39  0x61

//Read PMIC starting at LDO1
// for next 10 registers
//Result is visible in Datalog window

READ   LDO1 10

//Set port GPIO8 or SYS_EN to 1
PORT GPIO8 1
//Set port GPIO9 or PWR_EN to 1
PORT GPIO9 0

//Wait  255ms
DELAY 255
WAIT 0xFF

//Read ports GPIO8 and GPIO9
//Result is visible in Datalog window

PORTREAD GPIO8
PORTREAD GPIO9
//finish
```

Alternative form

```
//Register Dump
//Slave      Register    Data
0x92   0x01   0x00
0x92   0x02   0x01
0x92   0x03   0x02
0x92   0x04   0x02
0x92   0x05   0x04
0x92   0x06   0x05
0x92   0x07   0x06
0x92   0x08   0x07
0x92   0x09   0x08
```

The results in the history log file are shown in Figure 18. The Read operations are in brackets following the command.

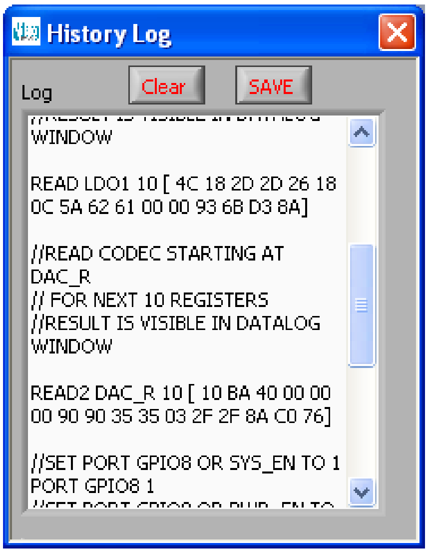


Figure 18: Example History Log Results

Appendix B Tutorial Guide

This section provides a step-by-step guide for starting up the Evaluation Board and software in order to get the device up and running. It also details the first steps of using a project template and making modifications.

B.1 Powering the Board

The board can be configured to be self-powered from the USB 3V3 via the VBAT connection. Move jumper 37 to the right-hand position giving the board a battery supply. It will also be necessary to switch J58 to its uppermost position (permanent). However, please see the cautionary note in Section 4.1.

The positions are indicated by the green arrows in Figure 19.

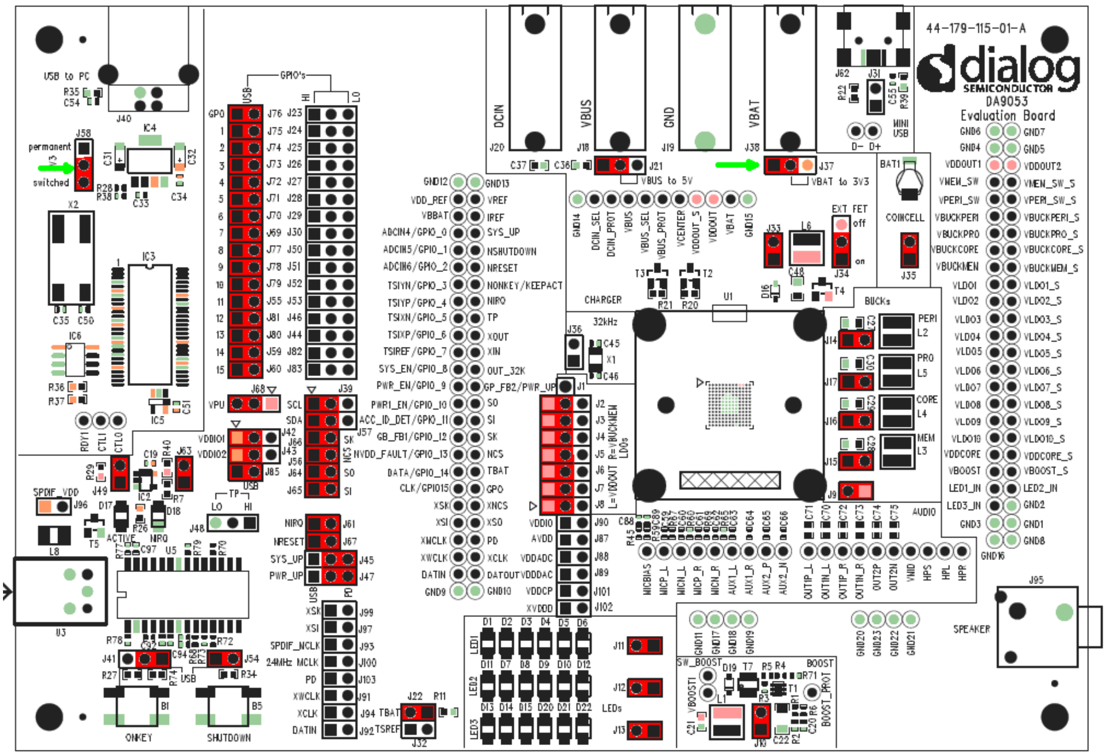


Figure 19: Self-Powered Board Configuration

Ensure that the toggle switch J48 is in the left-hand position. This is opposite to the marking on the board. TP will be held HI.

B.2 Using a Project Template

In this section all references to LED are on the screen unless otherwise stated.

To begin, start the software and perform some routine checks:

- **USB OK?** LED on the right-hand side of the screen is a steady green.
- **Interface** selector shows **USB I2C**.
- **Status** indicator at the top right of the screen has the **Reset** LED lit.

From the **Device** pulldown control in the middle of the screen, select **DA9053_MX53_3B_0R2_BDCB**. It will take a few seconds to update all the controls on the screen.

Ensure that the **Power Commander mode** LED in the middle of the screen is lit, and the **Download** LED is red.

The chip is now in its Reset state. When in its current mode (Power Commander mode) it is waiting for the initial register state to be transmitted over the interface. In normal (not Power Commander) mode the device loads all its defaults from OTP and progress to the Power Down phase. There it either waits for a Wakeup event, or progresses straight on to Active mode if **Autoboot** has been set.

In this example the device receives data by pressing **Start Device** and progress to the Active state whereby the **Download** LED goes out.

Start Device performs some other useful functions:

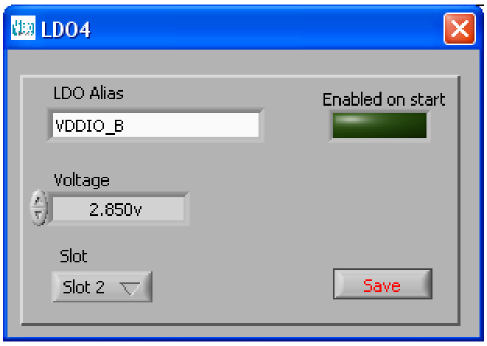
- sets the Date and Time for the RTC and Alarm to the current time,
- begins sending Watchdog register writes in case this option is set,
- starts monitoring the Interrupt flag,
- continues to check that the device is communicating and that the USB is still connected.

The device is now active and ready for some initial checks to be performed:

- Monitor the output of LDO4. This should be on, and reading approximately 2.85 V.
- Click the **Regulators** tab and locate register LDO_4. The control LDO4_EN should be green and Enabled.
- Change the voltage to 2.600 V by clicking on the LDO4 control and observe the result on the voltmeter.

To make changes to the configuration:

1. To open the LDO4 window, click the **Configuration** tab, and then click on **LDO4**.



2. For a list of available voltages click on the **Voltage** field, then select **2.600v**.
3. Click **Save**.

The LED next to the device selection box is now orange. This is a flag to indicate that something has changed from the original configuration. The original configuration has not been changed, a temporary copy of the configuration has been created.

To make the changes permanent they can be saved as a new project.

1. From the **File** menu, choose **Save Project**.
2. Type the new project name, for example, **Project_XYZ**. The .ini extension is automatic.

The yellow LED goes out and the name of the project is now **Project_XYZ**. Therefore, the association with the original file **DA9063_MX53_3B_0R2_BDCB**.ini has been removed.

Further changes can be made in the same way. Project _XYZ can be overwritten with other changes, or you can generate incremental versions for debugging purposes. When exiting the software with unsaved modifications (the yellow LED is lit), there will be a prompt to save them.

At this point the changes have not been loaded into the device, only stored in the configuration file. To see the effect of the change, the device can be shut down by pressing the **Shutdown** button, and the new configuration reloaded by pressing **Start Device**.

B.3 Using the Power Sequencer

The Power Sequencer is launched by pressing the **Power Sequencer** button on the **Configuration Panel**.

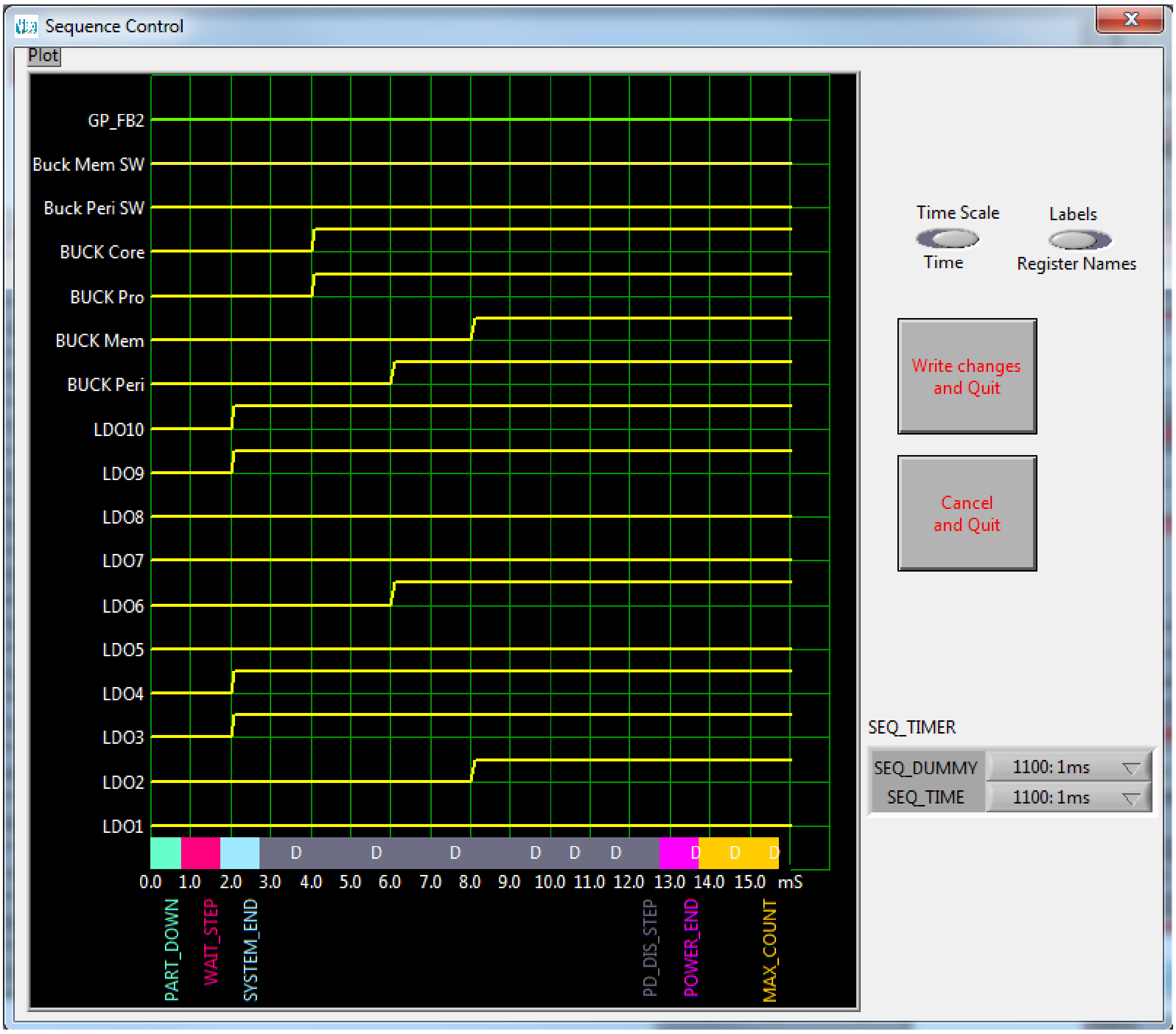


Figure 20: Power Sequence Control

NOTE
The waveform displayed has limitations. There are other control registers which affect the way that regulators and signals appear from the device, and the effect of these are not all reflected in this view. The intention of this screen is to allow interactive scheduling of regulators in a way that is more familiar and understandable than a simple list of register values.

As an example, move LDO4 (or VDDIO_B) to be enabled after LDO5, and at the same time as LDO2. The toggle switch **Labels** changes between the Alias names and the LDO numbers.

1. Click on the **LDO4** transition in slot 2 and drag it to slot 4.
2. Click **Write Changes and Quit**.

The yellow LED is lit, indicating a change, and the project file can be updated using the **File**, **Save Project** menu.

The project file Project_XYZ.ini is stored in the directory:

C:\Dialog Semiconductor\Power Management\Power_Commander_vxx\data

It is easily readable and can be modified in this form if desired. However, only limited error checking of the file can take place, so transcription errors can be a bit difficult to spot (for example, non-existent voltage settings).

B.4 Setting User-Wakeup Mode

To allow control over the download and wakeup events, rather than these being controlled automatically change the device from **Autoboot** mode (requiring only power application to wake up), to a **User-wakeup** mode:

1. From the **Settings** menu, deselect **Autostart**.
2. From the **Configuration Panel** click **Change Configuration Register** then select register **CONTROL_B**.
3. Change the AUTOBOOT bit from **Enabled** to **Disabled**.
4. Click **Save**.
5. Save the project using the **File**, **Save Project** menu.
6. From the **Controls** tab, switch off the device by pressing **SHUTDOWN** on register **CONTROL_B**.
7. Change to the **Configuration** panel.

The device is now back to the beginning, waiting for a download. The following observations can be made:

- Press **Download** and nothing happens - The device is now in **Powerdown** mode waiting for a wakeup event. Pressing the **Start Device** button will pulse nONKEY low and supply the wakeup. There are a number of wakeup events, as described in [1].
- As a result of the wakeup event the **Download** LED illuminates again. The device is waiting for a second download, which is provided by pressing the **Download** button. At this point the device goes active.

The device performs *two* reads of the OTP (or download from the PC). The first loads all the registers to take the device into **Powerdown** mode. The second sets only the LDO and buck voltages back to their default state. They may have been altered temporarily in the active state.

DA9053 is a very powerful and flexible device, but it will obviously bring with it complexity and some logic puzzles. This software will evolve to assist in these tasks.

Revision History

Revision	Date	Description
1.0	08-May-2013	Initial version.
1.1	12-May-2017	Update to latest template.

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.

Disclaimer

Contacting Dialog Semiconductor

United Kingdom (Headquarters)
Dialog Semiconductor (UK) LTD
Phone: +44 1793 757700

Germany
Dialog Semiconductor GmbH
Phone: +49 7021 805-0

The Netherlands
Dialog Semiconductor B.V.
Phone: +31 73 640 8822

Email:
enquiry@diasemi.com

North America
Dialog Semiconductor Inc.
Phone: +1 408 845 8500

Japan
Dialog Semiconductor K. K.
Phone: +81 3 5425 4567

Taiwan
Dialog Semiconductor Taiwan
Phone: +886 281 786 222

Web site:
www.dialog-semiconductor.com

Singapore
Dialog Semiconductor Singapore
Phone: +65 64 8499 29

Hong Kong
Dialog Semiconductor Hong Kong
Phone: +852 3769 5200

Korea
Dialog Semiconductor Korea
Phone: +82 2 3469 8200

China (Shenzhen)
Dialog Semiconductor China
Phone: +86 755 2981 3669

China (Shanghai)
Dialog Semiconductor China
Phone: +86 21 5424 9058

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/.