DA9212 Multi-phase buck converters and Power Commander software

Abstract

This application note is a user guide on DA9212 and is delivered along with the Dialog evaluation box. It provides the basic information on how to configure the PCB, how to install and use the Power Commander evaluation software.
The practical tutorial introduces the not experienced user to first basic measurements on DA9212.

Contents

Figures

1. Revision History

Version	Date	Description
1.0	August 2014	Initial revision

2. Terms and definitions

BGA	Ball Grid Array
CPU	Central Processing Unit
DDR	Double Data Rate SDRAM (Synchronous Dynamic Random Access Memory)
DUT	Device Under Test
GPU	Graphic Processing Unit
PCB	Printed Circuit Board
POL	Point Of Load
PSU	Power Supply Unit
USB	Universal Serial Bus

3. References

1. DA9211(2) Data sheet, Dialog Semiconductor

4. Introduction

DA9212 extends the Dialog PMIC family with multi-phase buck converters for the supply of CPUs, GPUs, DDR memory rails in smartphones, tablets and other handhelds applications.

The output voltage is configurable in the range 0.3 - 1.57 V. The input voltage range of 2.8 – 5.5 V makes DA9212 suited for a wide variety of low voltage systems, including all Li-Ion battery supplied applications.

DA9212 integrates two dual-phase buck converters, each capable of delivering up to 6 A continuous output current.

The evaluation kit includes:

1) a performance board, which has been designed to allow the measurement, evaluation and programming of DA9212 (212-04-C).
2) a USB stick with the software to be installed on your computer. The Power Commander software is the same for both DA921 and DA9212. It uses a simple graphical interface, allowing the ICs to be controlled via USB port. The mini USB connection is visible on the bottom side of the motherboard. As long as the cable is connected to the USB port of the laptop, thus delivering 5 V supply to the motherboard, a green LED is on.
3) a USB cable

DA9212 performance board is based upon the PCB hardware numbered 212-04-C.

Board includes an USB-I2C bridge for communication with the device and a few external active components to reduce the requirement for external equipment.

The performance boards have some jumper links to provide access to different configuration and measurement test points. Altering the jumper positions should be done only after a complete understanding of the links description and may need a supplemental configuration of some registers of DA9212.

The performance boards have soldered devices (it’s not possible to exchange devices on a socket) and are intended to be supplied by a single 3.6 V supply (nominal) plugged into the 4 mm sockets VSYS and GND. A Lithium Ion battery may be used for the same purpose, or a “source meter” instrument, e.g. Keithley 2400, Hameg 8142 or similar.

The software allows the configuration of the device by write and read operations to all control registers, and provides monitoring of device status. Make sure you have an operating system Windows 2000/XP/Vista/Windows 7 with a USB1.1 or USB2.0 interface on your machine.

For questions and clarification please refer to your local Dialog support team.

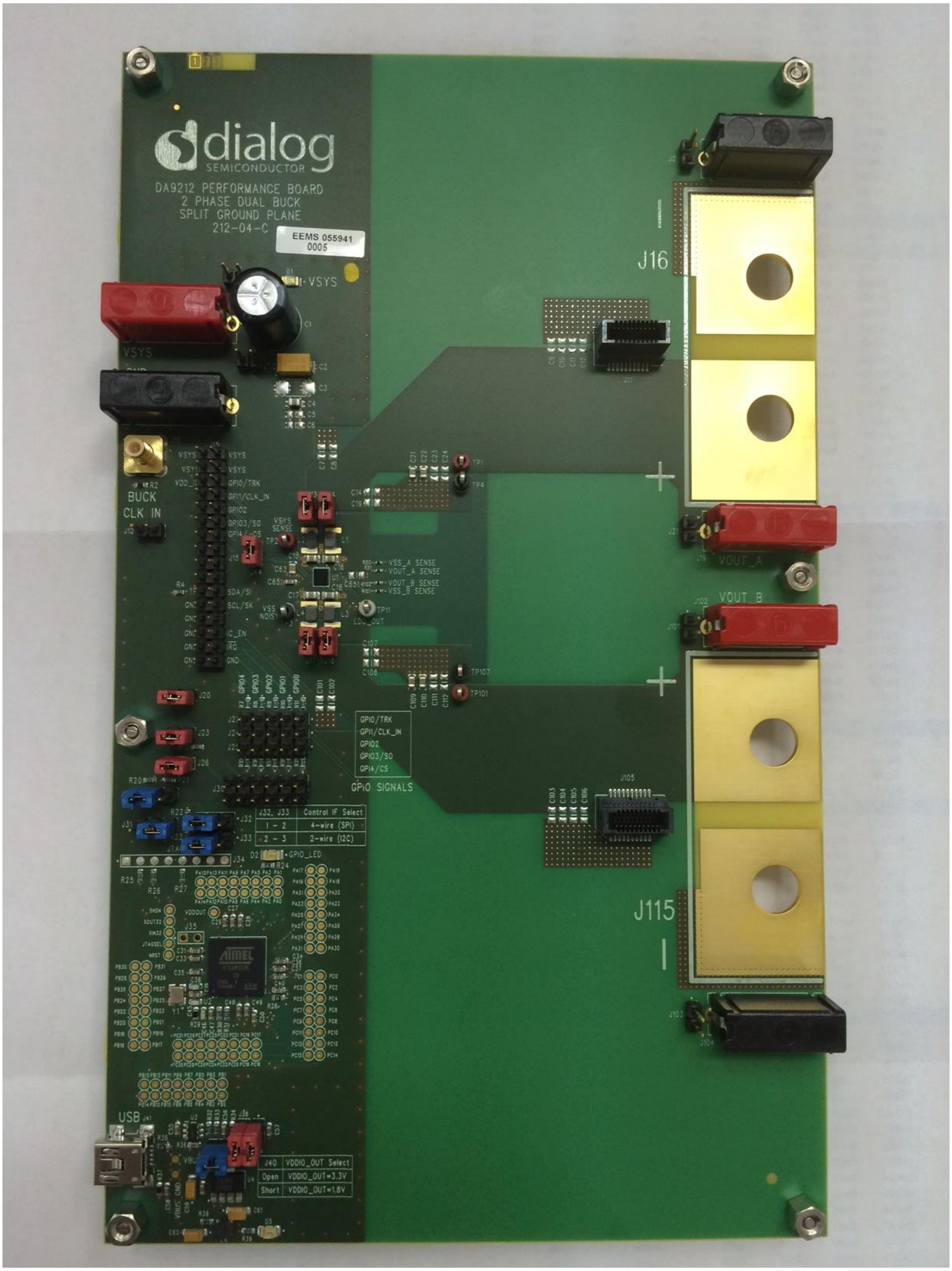


Figure 1: Performance board 212-04-C (for DA9212)

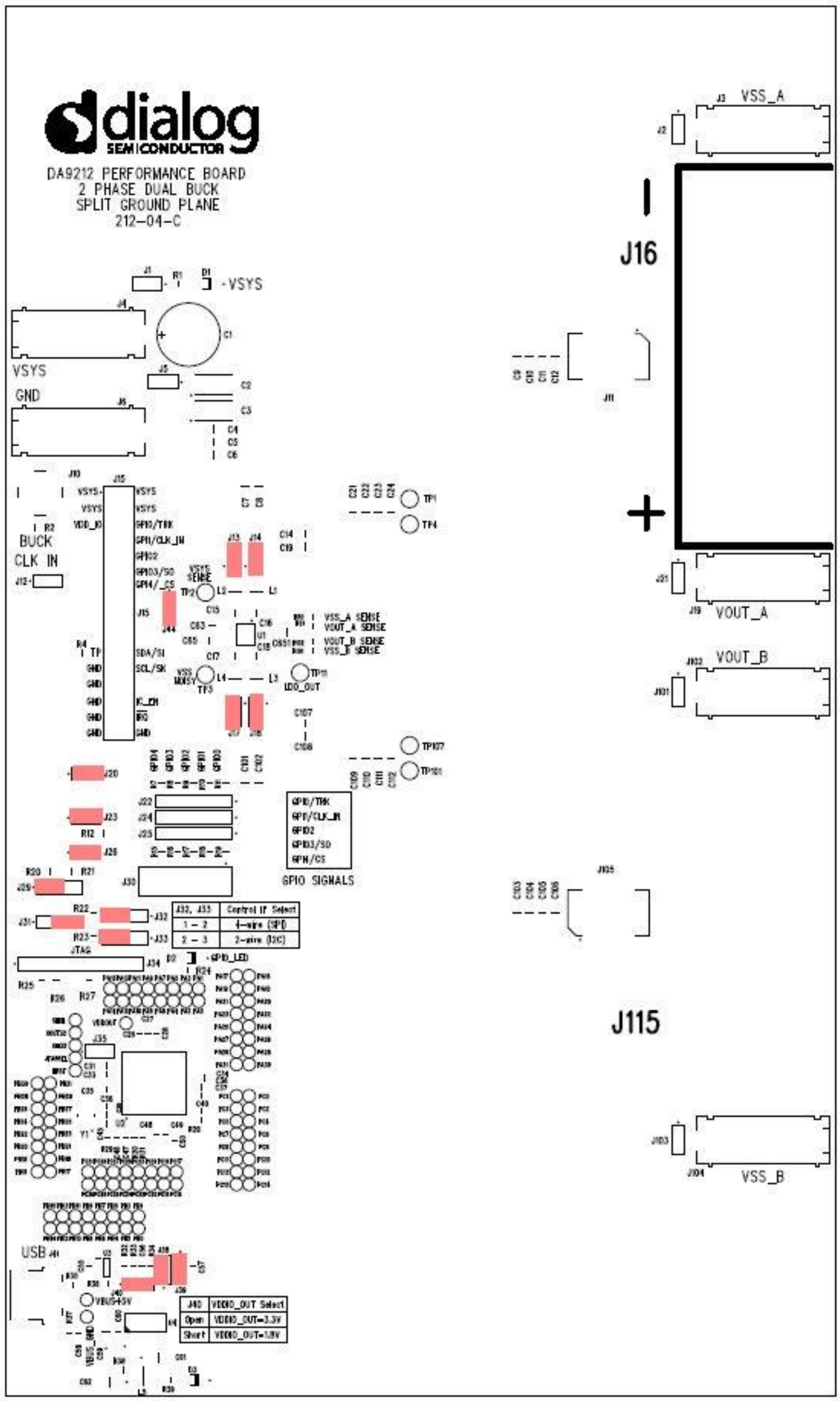
5. Links Description

Table 1: Performance Board 212-04-C. Links description

Link	Position 1 (pin 1-2)	Position 2 (pin 2-3)	Function
J1	to VSYS		enables the diode D1 when VSYS supply is connected
J2	Buck_A local GND		additional probe of Buck_A VSS local for load jumps measurements
J5	to GND		short VSYS to GND
J10	to clock input		connects GPIO1 to plug for external clock input signal
J13	to VOUT		connects the LX_A2 to the output of the buck converter (needed for Buck_A phase2)
J14	to VOUT		connects the LX_A1 to the output of the buck converter (needed forBuck_A phase1)
J15			probe I/Os and main signals
J17	to VOUT		connects the LX_B2 to the output of the buck converter (needed for Buck_B phase2)
J18	to VOUT		connects the LX_B1 to the output of the buck converter (needed for Buck_B phase1)
J20	to VDDIO		connections for GPIOs (via 100 kΩ pull up resistors)
J21	Buck_A local VOUT		additional probe of Buck_A VOUT local for load jumps measurements
J22			high side connection of GPIOs
J23	to VDDIO		connections for nIRQ (via 100 kΩ pull up resistors)
J24			sense connection of GPIOs
J25			low side connection of GPIOs
J26	to VDDIO		connection for I2C interface pull up (via 2.2kΩ resistors)
J29	to VDDIO (via 100kΩ resistor)	to GND (via 1kΩ resistor)	connection for IC_EN
J30			connection of GPIOs to ATSAM3U
J31	to ATSAM3U port	to J29	connection for IC_EN
J32	to SPI_SCLK	to SCL_0	selection of SPI or I2C clock
J33	to SPI_MOSI	to SDA_0	selection of SPI or I2C data
J38	to VDDIO_OUT		connect to VDDIO to the VDDIO_OUT rail generated from USB
J39	to VDDIO_OUT		connection of VDDIO_USB used by ATSAM3U

Link	Position 1 (pin 1-2)	Position 2 (pin 2-3)	Function
J40	low voltage VDDIO		selection of VDDIO_OUT voltage: open => 3.3 V short => 1.8 V
J44	to VSYS		connection of the clean analogue supply to the VSYS rail
J101	Buck_B local VOUT		additional probe of Buck_B VOUT local for load jumps measurements
J103	Buck_B local GND		additional probe of Buck_B VSS local for load jumps measurements

Figure 2: Performance board 212-04-C. Links location



6. Performance board features

6.1 USB-I2C bridge

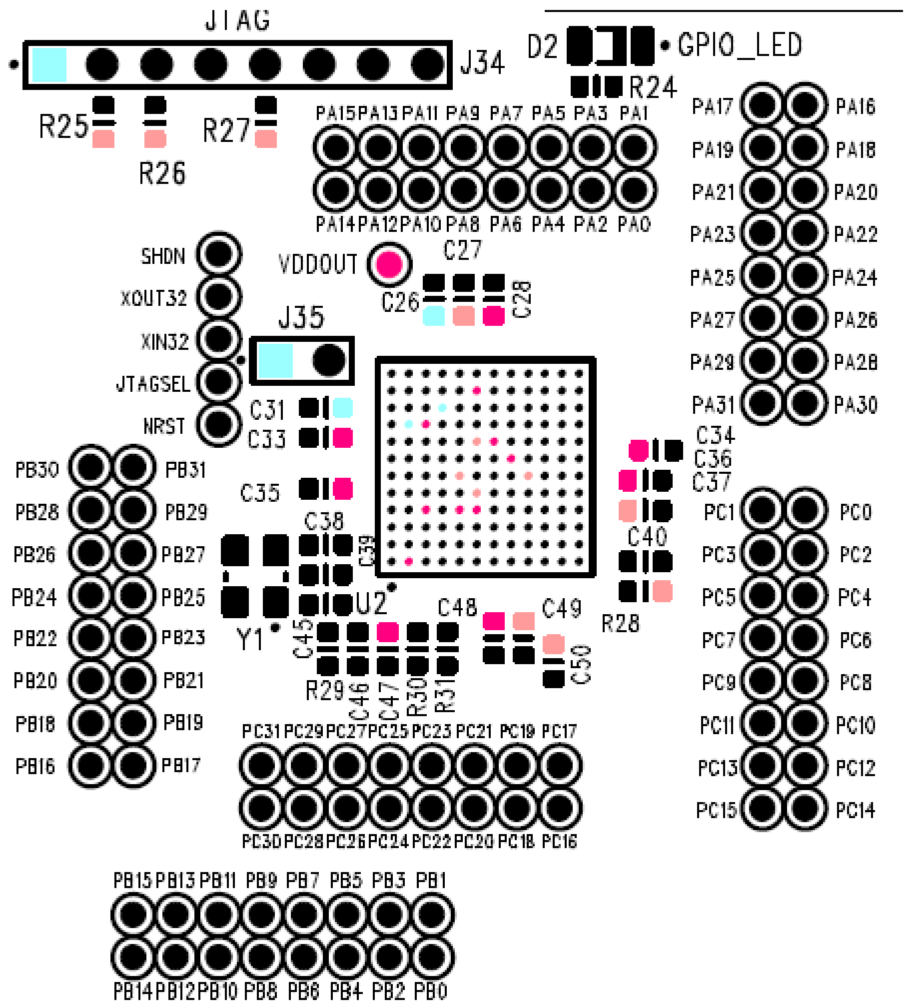


Figure 3: ATSAM3U µ-Controller for USB-I2C bridge

The USB-I2C bridge is used here mainly for two purposes:

1) as a source of I2C and SPI control signals

2) to provide the discrete signals for the GPIOs

The USB-I2C bridge is powered through the externally plugged USB bus cable via a fixed 3.3 V regulator.

The jumper links J32 / J33 on the performance boards are used to select between I2C and SPI communication to DA9212. This has to be configured together with the IF_TYPE register field.

6.2 I/O and supply regulators

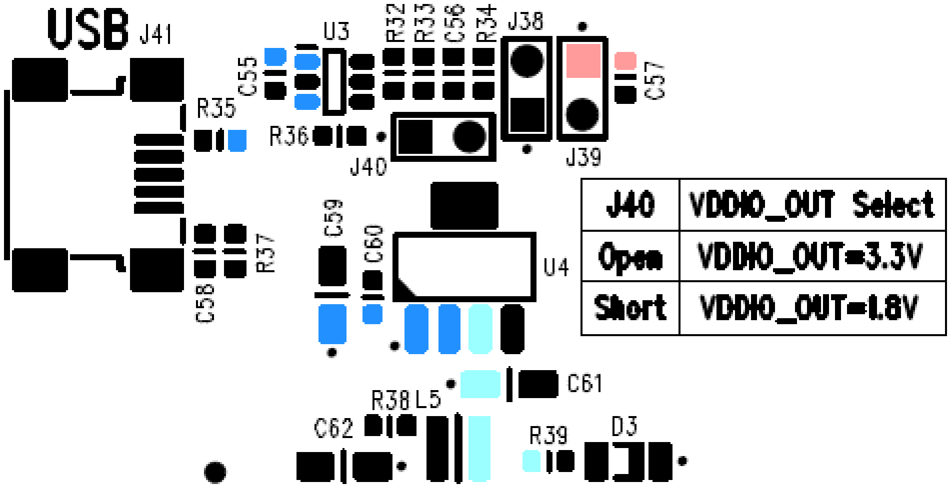


Figure 4: Mini USB input and internal rails generation

The U3 and U4 LDO regulators included in the performance boards generate the internal supply for ATSAM3U and the I/O voltage needed by DA92112.

A voltage of 3.3 V or 1.8 V can be selected as supply for the I/O (see jumper link J40).

6.3 Control and I/O signals

All of the Control and IO signals of DA9212 appear on the headers pins shown in Figure 5. These are the most useful monitor points for debug purposes.

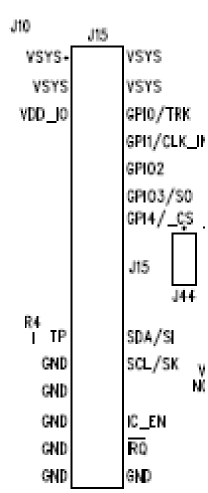


Figure 5: Main probe signals of DA9212

If required, a suitable connector can bring these signals to the system board for integrated development. 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 is shown in Figure 6. The jumpers may be inserted in the links J30 for USB control and monitor. The jumpers can also be placed in the headers J22, J24, J25 as follows.

1) Between J22 and J24, connecting to VDD_IO vis 100 kΩ pull up resistor,
2) Between J24 and J25, connecting down to ground via 100 kΩ pull down resistor,
3) Left open for external connections.

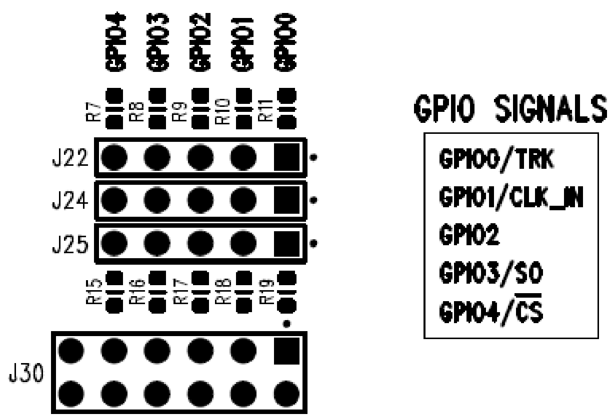


Figure 6: GPIOs control jumpers

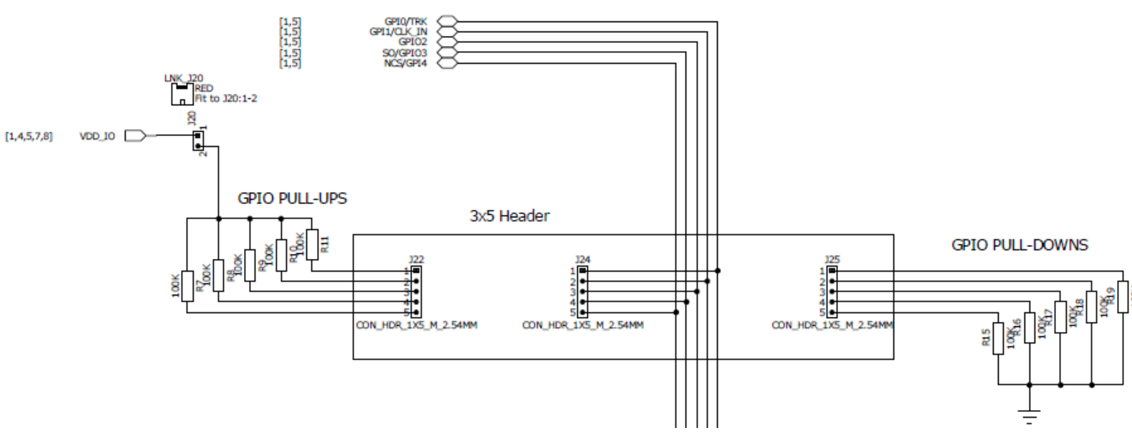


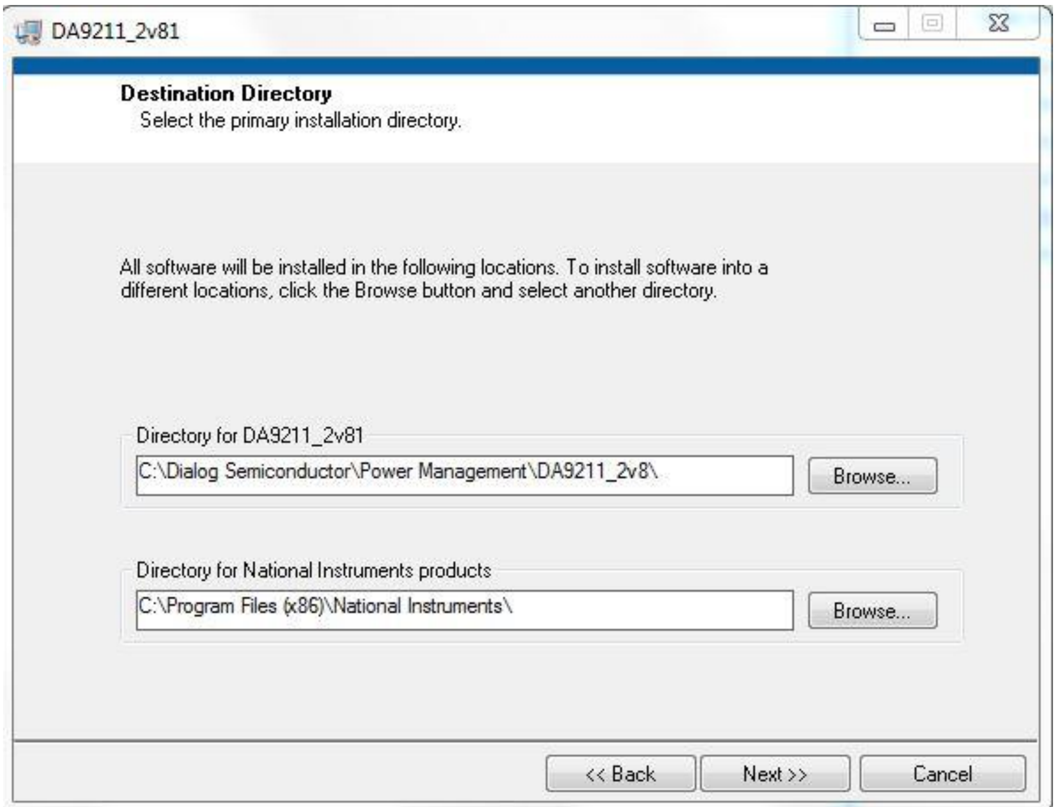
Figure 7: GPIOs control schematics

7. Software

7.1 Installation

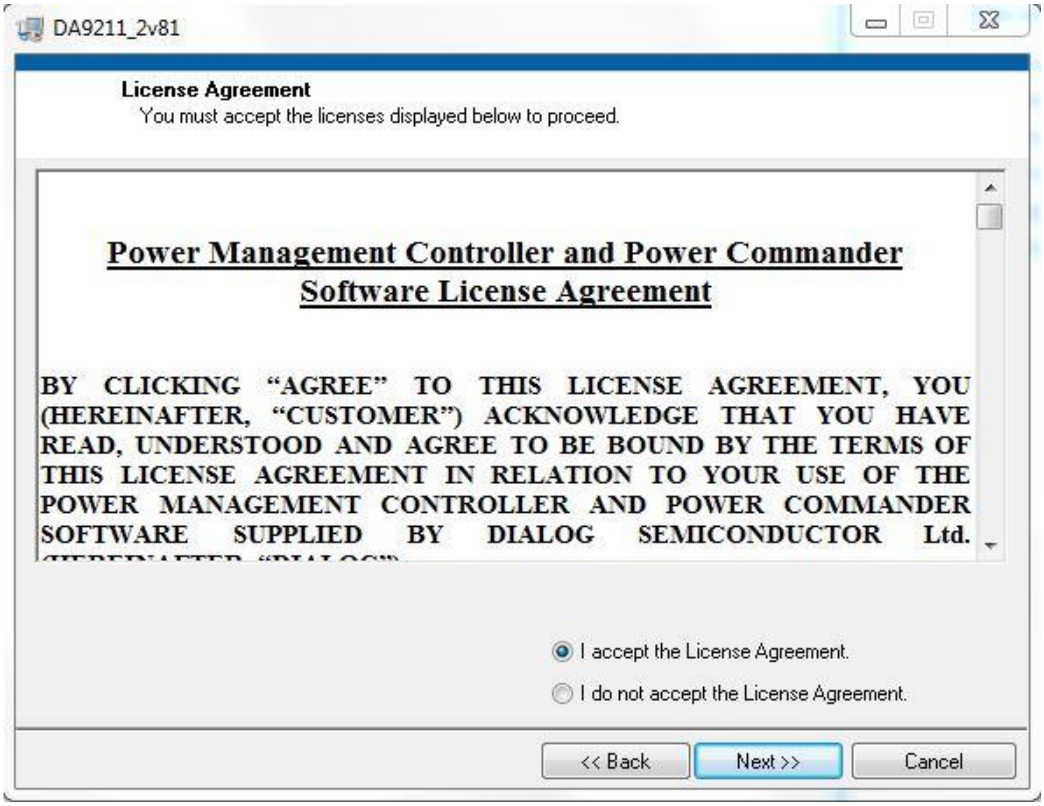
Plug the USB stick in one of the free ports of your computer and run the program 'setup.exe'. This will be found in the directory 'Software'. An automated script will start.

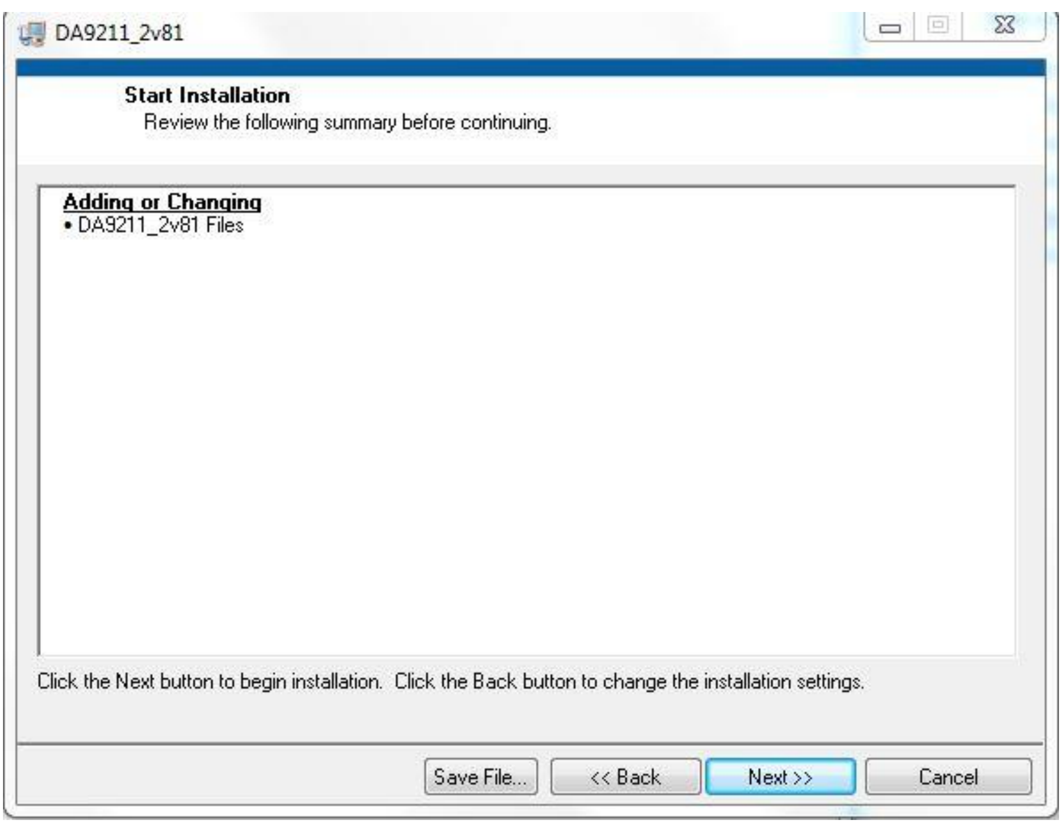
By default, the directory C:\Dialog Semiconductor\Power Management\DA9211_vxx will be used.

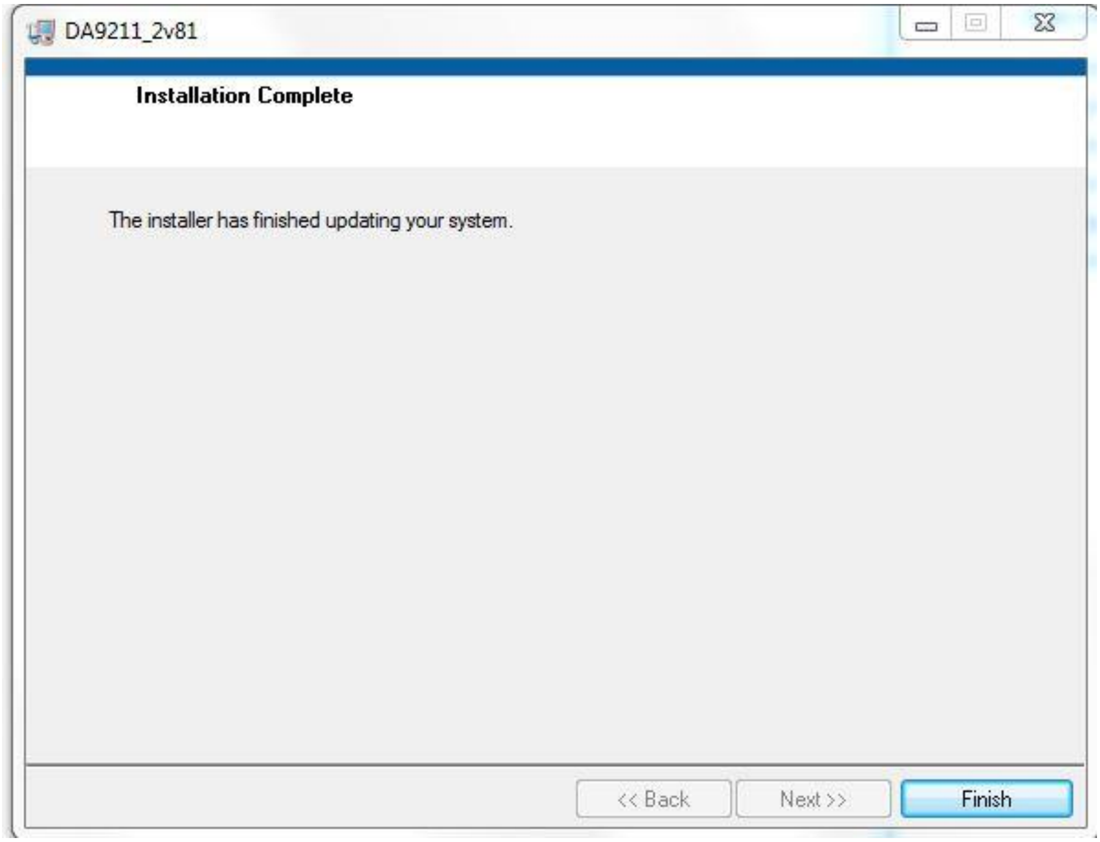


Press "Next" and accept the software agreements twice.

Then start the installation.

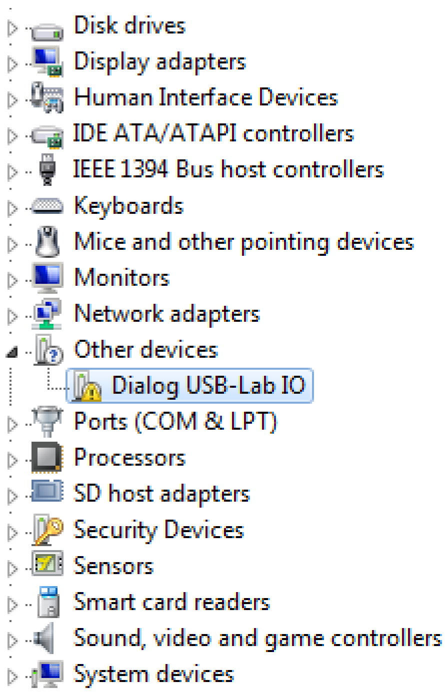

DA9211_2v81
License Agreement
You must accept the licenses displayed below to proceed.
Power Management Controller and Power Commander Software License Agreement
BY CLICKING "AGREE" TO THIS LICENSE AGREEMENT, YOU (HEREINAFTER, "CUSTOMER") ACKNOWLEDGE THAT YOU HAVE READ, UNDERSTOOD AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT IN RELATION TO YOUR USE OF THE POWER MANAGEMENT CONTROLLER AND POWER COMMANDER SOFTWARE SUPPLIED BY DIALOG SEMICONDUCTOR Ltd.
I accept the License Agreement.
I do not accept the License Agreement.
<< Back
Next >>
Cancel


DA9211_2v81
Start Installation
Review the following summary before continuing.
Adding or Changing
• DA9211_2v81 Files
Click the Next button to begin installation. Click the Back button to change the installation settings.
Save File...
<< Back
Next >>
Cancel

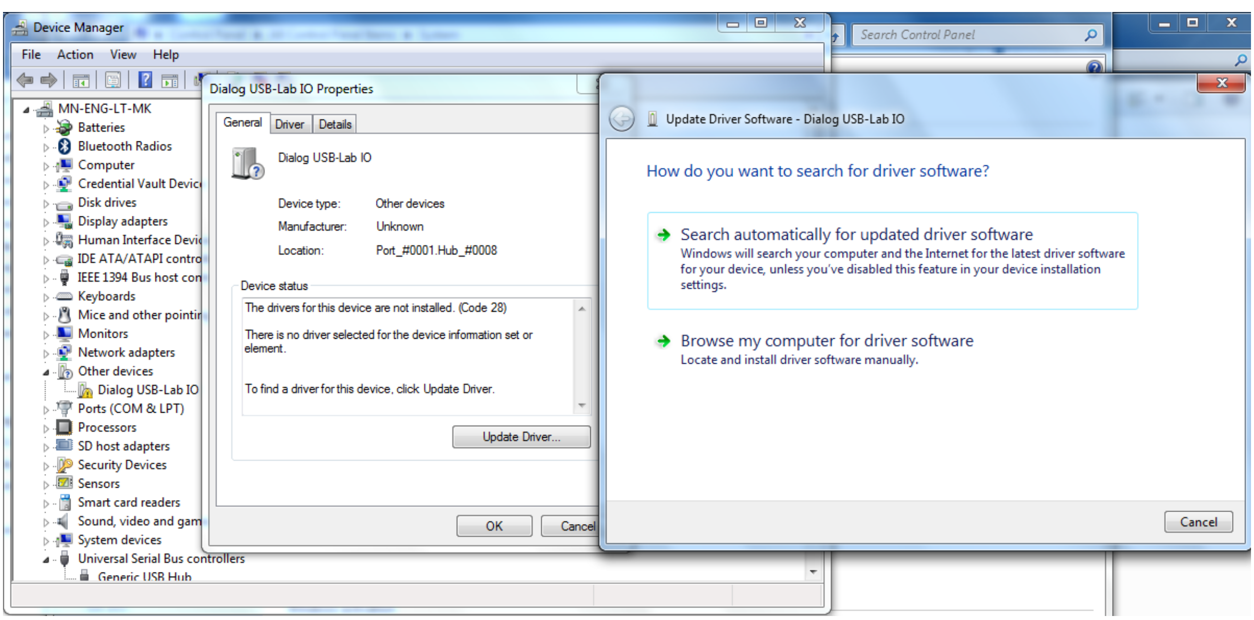


After the installation has been completed you need to restart your computer.

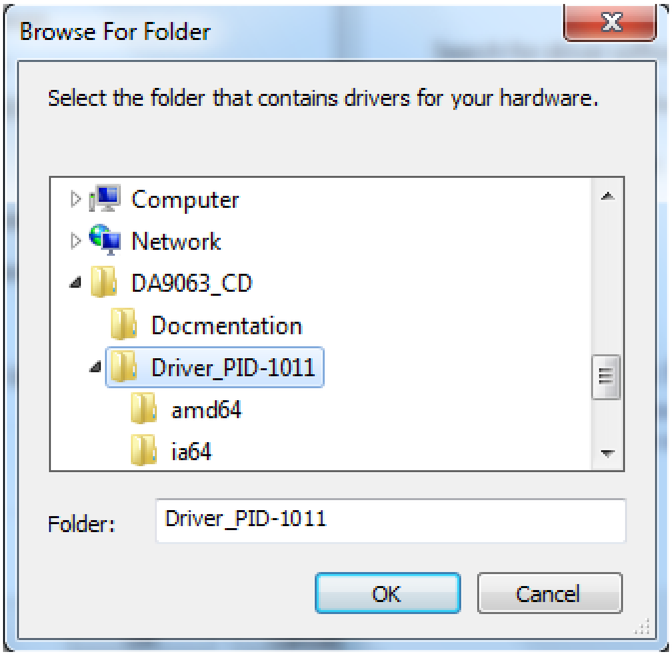
When you plug in the USB cable Windows will detect the USB device. It will prompt for the drivers, which should be automatically located in the “Driver_PID-1011” directory of the USB stick. If this does not happen automatically you should open the Device Manager and get a list similar to the following:

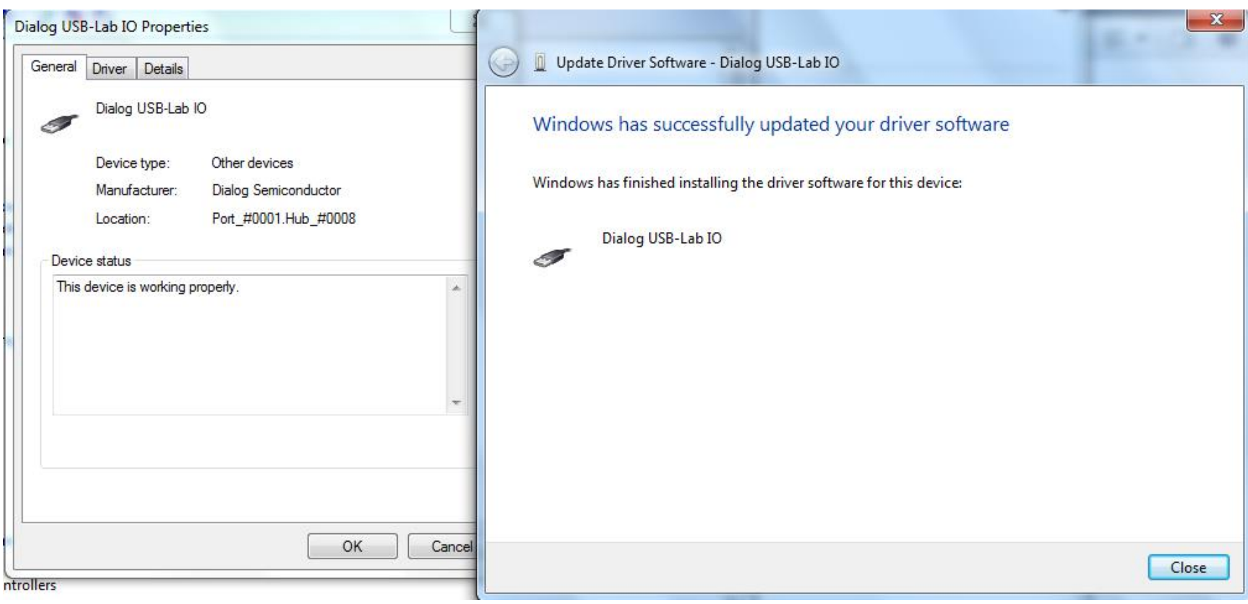


After double click on the unknown Dialog device, you should update the Driver like the following:



By selecting the right directory the Driver should be found.





If installed correctly, Dialog USB Driver should be listed on Device Manager.

To uninstall the software please use the Windows ‘Add/Remove Programs’ function that can be found under ‘Start’->’Settings’->’Control Panel’.

7.2 Configuration panel

Run the software by clicking the shortcut on the appropriate item in the Start menu.

The best setting for the PC display size is 1024x768 pixels or above. 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 panels appear.

A window will pop up for selecting a configuration file. If you don't have any to load, just click on "Cancel".

The following screen appears, with the "USB OK?" LED lit if the USB cable is correctly plugged and the interface is functioning.

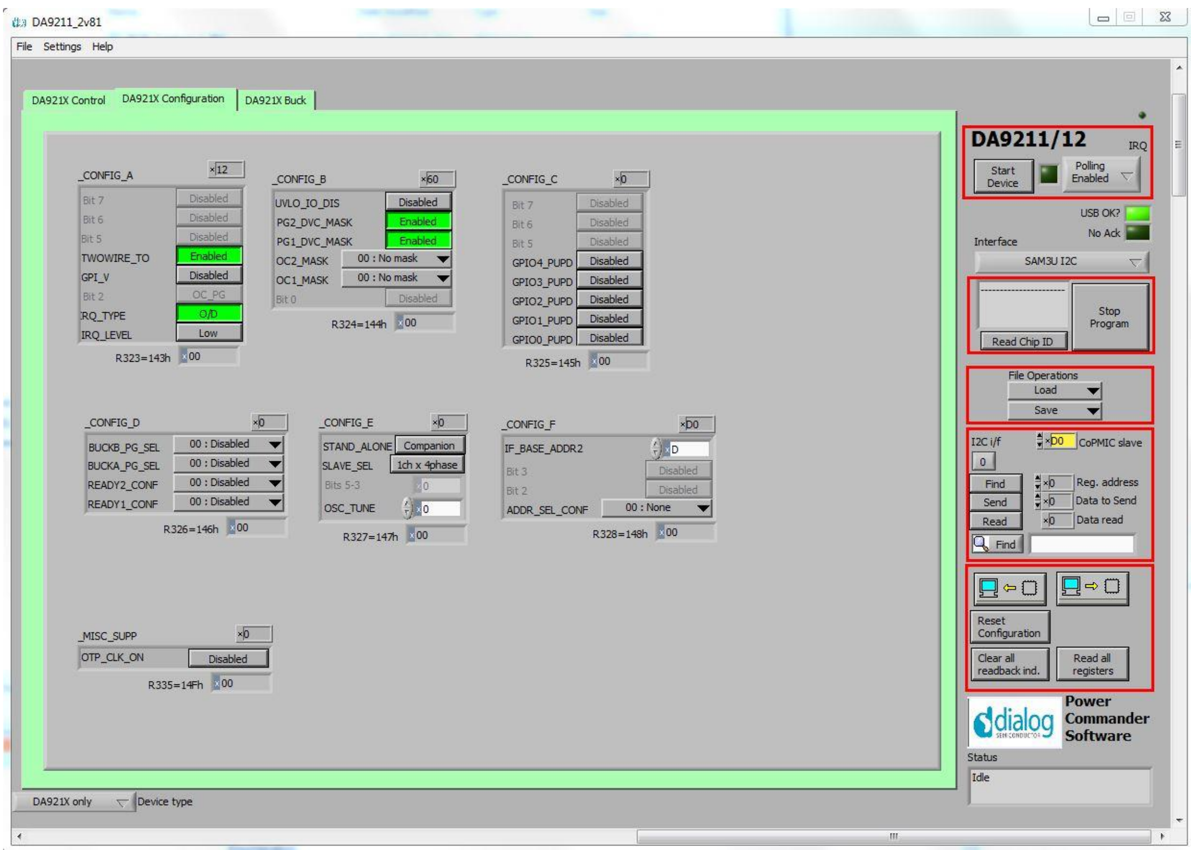


Figure 8: Initial screen

You'll find three tabs on the top:

1) a "Control" tab for the main settings of I/Os, the interface and the status/event registers
2) a "Configuration" tab for advanced DA9211(2) configuration
3) a "Buck" tab including all the buck settings like output voltages, operating modes, etc.

As the software has been developed to evaluate DA921X and DA9063 (main PMIC) separately or together on the same PCB, you can configure the specific DUT in the window at the bottom left. By default this is only DA921X.

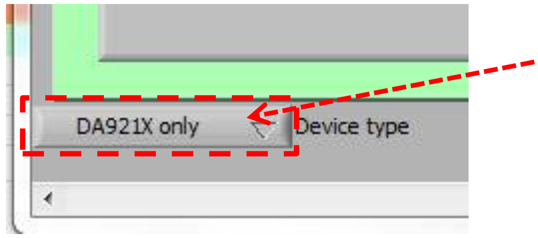
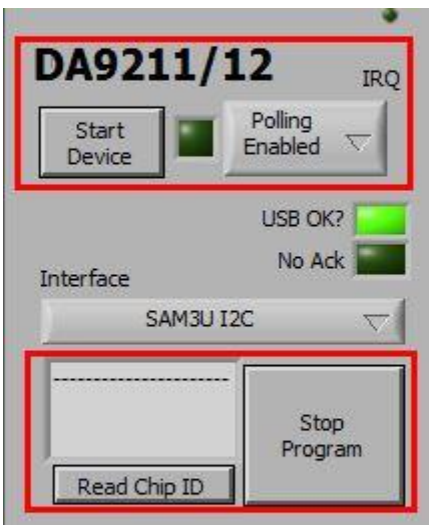


Figure 9: DUT selection

Some parts of the GUI software, like the Start Device button, the File and Settings options (see Configuration, Autostart options etc), do apply only to the main PMIC, so they can be safely ignored for DA9211(2).

The rest of the chapter will introduce you step by step to the relevant configuration parts of the GUI, which apply to DA9211 and DA9212.

7.2.1 Interface indicators



Start Device

When pressed it automatically configures the register R327 CONFIG_E for stand alone operation.

Polling Enabled

If disabled, main read backs from DA9211(2) 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

If device is active this is green, else red.

Interface

Select between USB I2C control, offline mode and SPI. Switching to offline, then back to USB re-initialises the USB interface.

USB OK?

The light indicates that the USB is fine and communicating.

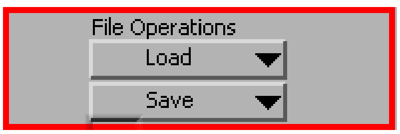
Read Chip ID

These fields indicate the device 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.

7.2.2 File load / save



Load

Load previously saved text files.

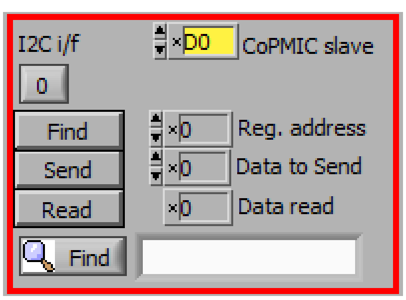
Save

Save current panel state to a 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.

7.2.3 Register Access

This is probably the most important section of the Configuration Panel.



CoPMIC slave

Set the slave address of the device. This affects all I2C communications. See also register INTERFACE in DA9211(2) Datasheet.

Find

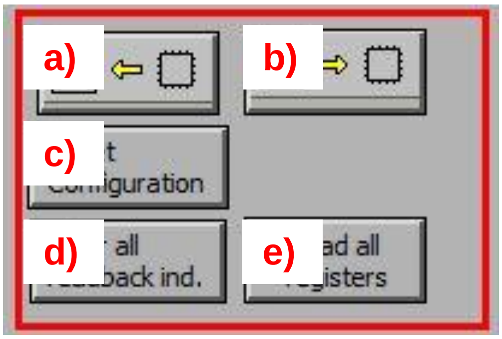
Finds a control matching a full or partial register name, a control bit name, a register number (e.g R208 or D0h). Pressing "Find" repetitively will step through all matching items.

Send

Sends a single byte data to I2C device using Slave Address, Register Address and Data to Send.

Read

Reads single byte data from I2C device using Slave Address and Register Address.



a) ***Synchronize Panel from Device***

Reads all the register contents of the device and updates the panel to match

b) ***Synchronize Device from Panel***

Writes all the device registers to match the panel (refresh operation)

c) ***Reset Configuration***

Resets the registers to values specified in configuration file.

d) ***Clear all read-back indication***

Sets all read-back indicators to 0

e) ***Read all read-back***

Reads all registers, comparing with the panel controls

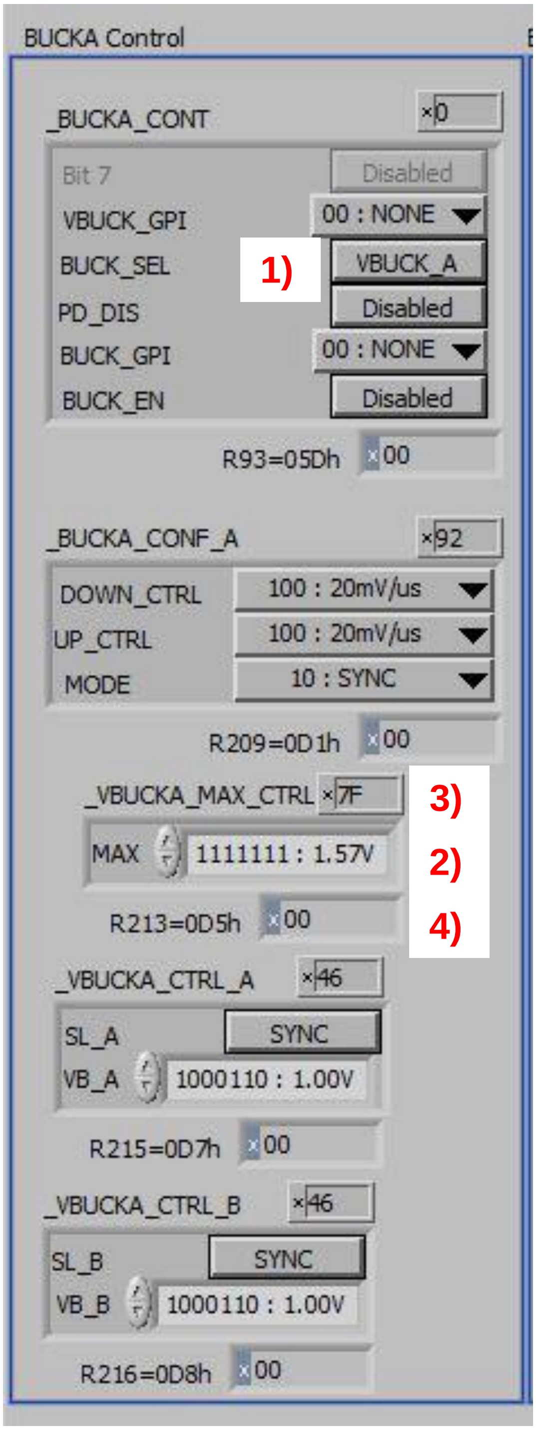
7.2.4 Control tabs

The tabs DA921X Control and DA921X Configuration and DA921X Buck all have the same format.

Each register cluster comprises a control with a mixture of

1) boolean toggle buttons
2) multi-value ring controls or slide controls
3) a hex indicator showing the total equivalent value
4) a read-back indicator showing the current contents of the register. The read-back indicator is labeled with the register number in both decimal and hex.

Read-back 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 Settings > Binary Indicators menu item.



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 Read-back indicator is red, it indicates that the current value does not match the panel.

8. Troubleshooting

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

8.1 Software Issues

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

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 to familiarise oneself with the software in a desk environment without the hardware attached. If the board is subsequently attached, move the Interface control to "SAM3U I2C" or "SAM3U SPI".

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 optimised 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.

Also switching the "Interface" to Offline, then back to the desired interface can reinitialise the USB interface. Last resort is to unplug the USB then reconnect. The software will detect this and reinitialise.

8.2 Hardware Issues

Most hardware problems can be traced to incorrect jumper positions.

Check carefully each jumper position, 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.

Especially watch for power supply configuration.

Link J40 can cause some confusion. The intention of these links is to control the 3.3 V / 1.8 V regulator generated by the USB.

Most times it is preferable if the USB is deactivated when 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.

9. A tutorial guide

This section will take you step by step through starting up the evaluation board and the software to get the device up and running. It will also tackle the first steps of making easy modifications on DA92112.

We refer here to the PCB hardware numbered 212-04-C with a DA9212 device mounted on it.

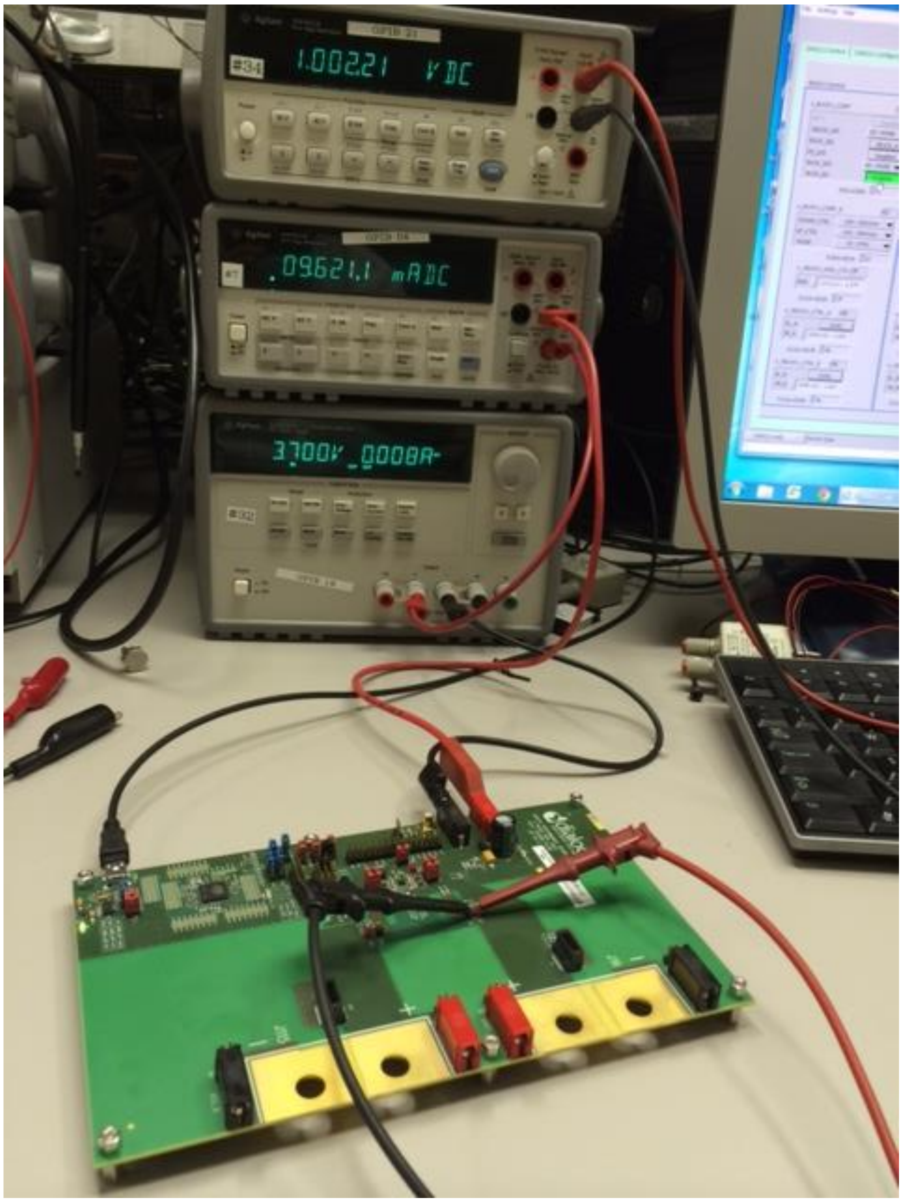
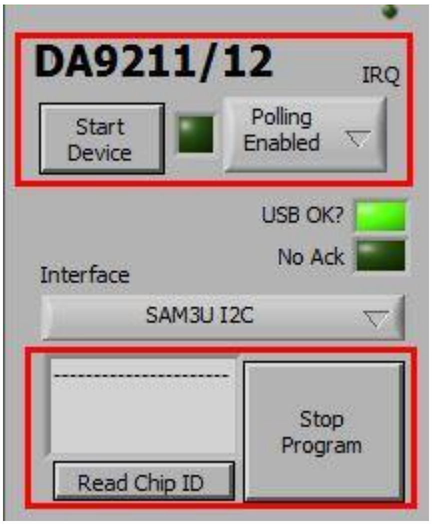


Figure 10: Measurement set-up

Supply DA9212 from the VSYS and GND connectors as shown in Figure 10. A voltage of 3.8 V (typical Li-Ion battery voltage) is applied and the input current is measured via multi-meter, see the blue cable from the PSU to the multi-meter input. The output voltage can be directly measured by connecting the yellow cable to a second multi-meter.

Start the GUI software installed on your computer. The “USB OK?” LED on the right hand of the screen should be off if the USB cable is not plugged. Once you plugged in the USB cable the LED should turn to a steady green. The Interface selector should show “SAM3U I2C”. If not, change it to off and then to “SAM3U I2C”.

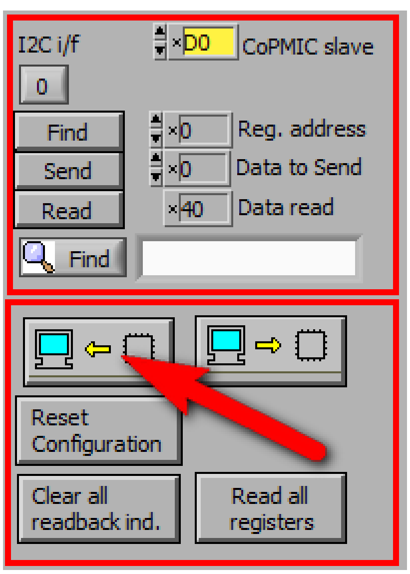
The following should appear on the top right end of your window.



Press the Start Device button to correctly configure the software for stand alone operation. We are now able to control DA9212 in stand alone mode. In this mode DA9212 acknowledges all I2C commands, which does not always happen if DA9212 is in companion mode.

For details please refer to the STAND_ALONE bit in the Datasheet.

If you now synchronize the panel from the device, by pressing the dedicated button (see red arrow below), you get the target voltages and all DA9212 parameters updated to the content of the IC registers.



Note that the quiescent current shown on the multi-meter may still be higher than expected. This is due to the missing configuration and connection of floating I/O ports and will be reduced as soon as they are correctly set.

You can enable the buck1 converter by pressing on the BUCK_EN button in BUCK1 Control, which is equivalent to write 0x01 into the related bit (see Figure 11). You'll notice an increase in the current drawn from the external supply and measured via the multi-meter of approximately 10 mA.

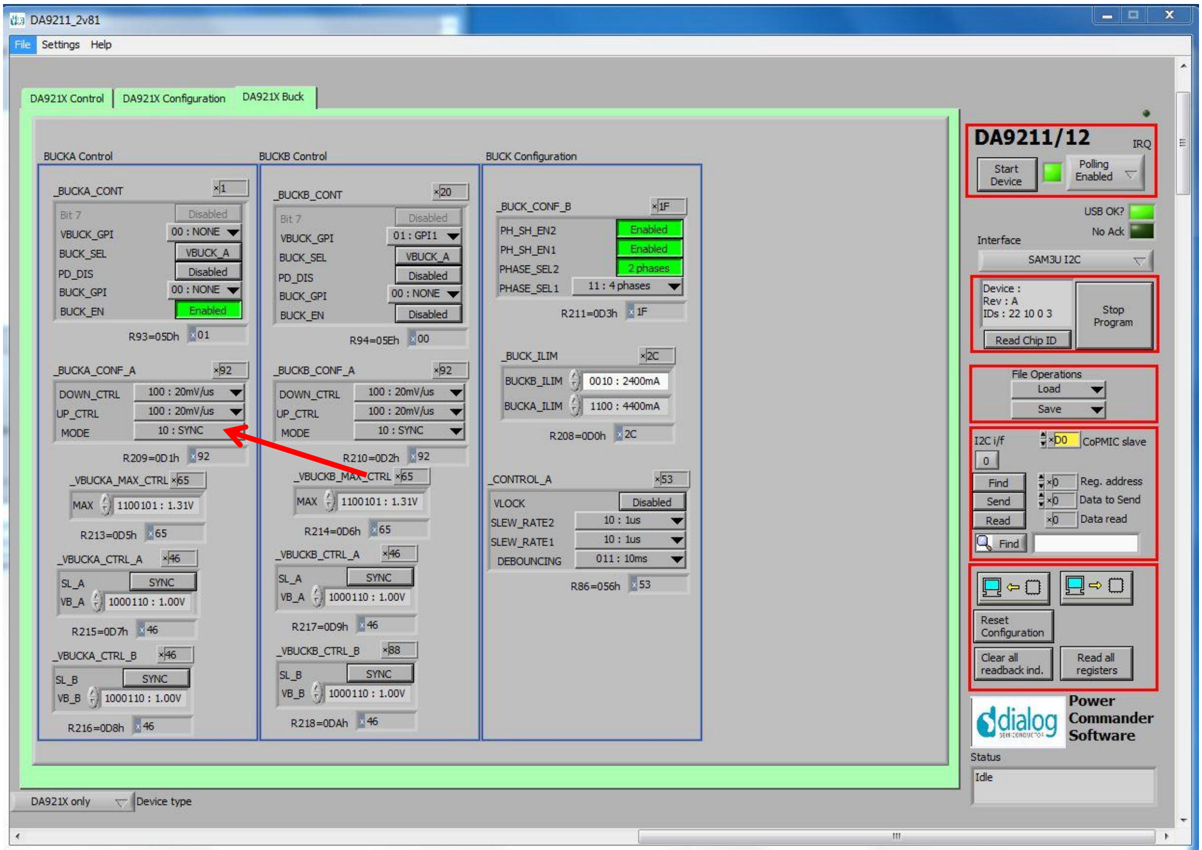
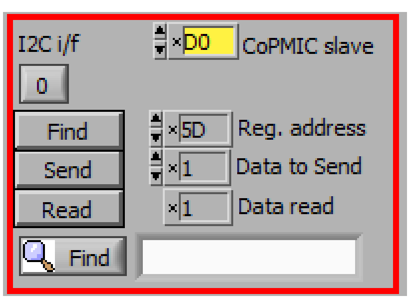


Figure 11: Enabling the buck 1

The same operation can be done also via I2C interface panel by selecting the register address, the data to send and pressing the Send button. A read back can be performed by pressing the Read button (see below).



After enabling the buck1 converter you can sense the output voltage on the PCB via external multi-meter connected to the VOUT_A pin, as shown in Figure 10. You should read back a value close to 1.0 V. If the part is not trimmed the value can slightly differ from the target.

You can directly sense on the PCB some relevant signals, e.g. the switching node of each phase. In the picture below you'll see how to find the phase 1 (red arrow) and the phase 2 (blue arrow) switching node.

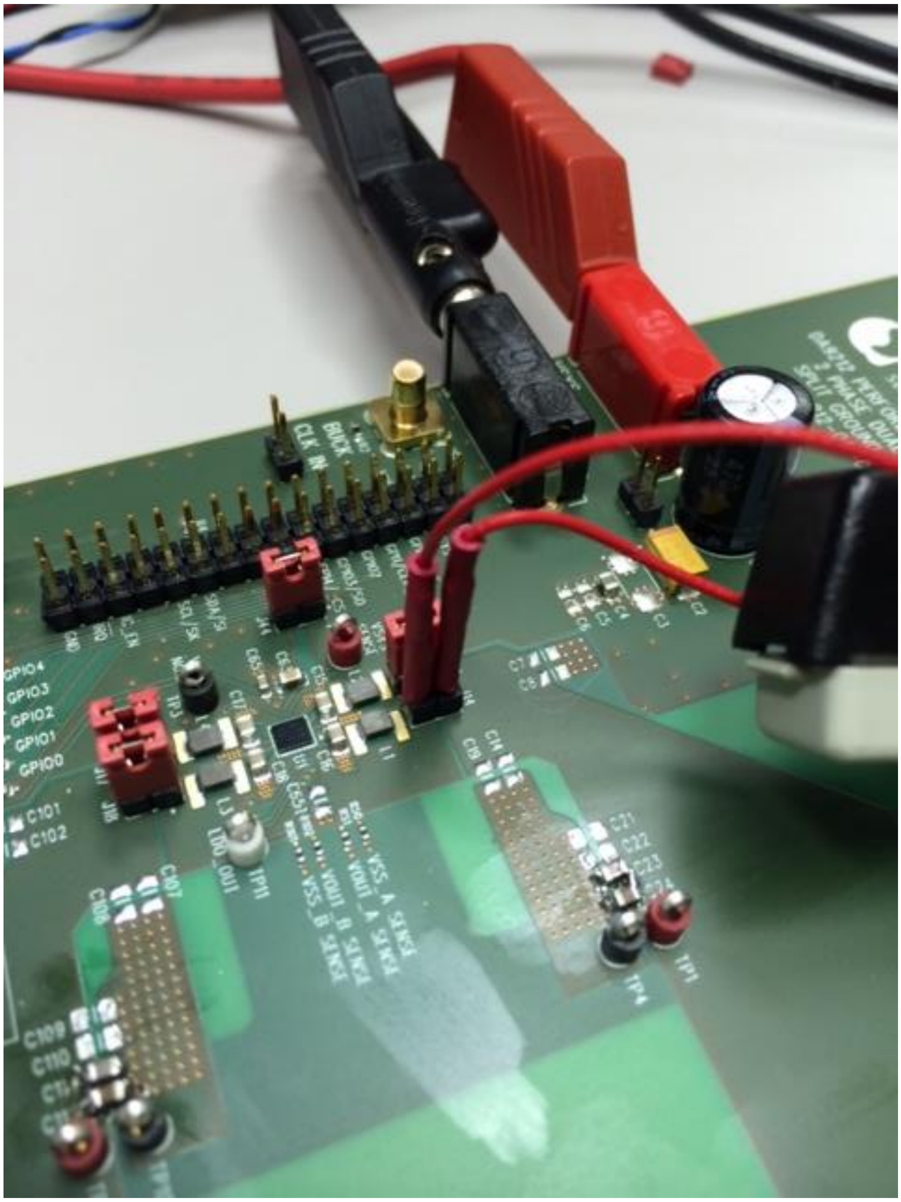


Figure 12: Sensing the LX signals of buck A

When you enable the BUCK_A, you will notice that only the phase 1 (LX_A1) shows a switching waveform, whilst the phase 2 (LX_A2) does not. This is the case if you have PH_SH_EN1 asserted. As you don't apply any load at the output of the buck converter, DA9212 automatically optimizes the number of active phases and turns off phase 2 to save power.

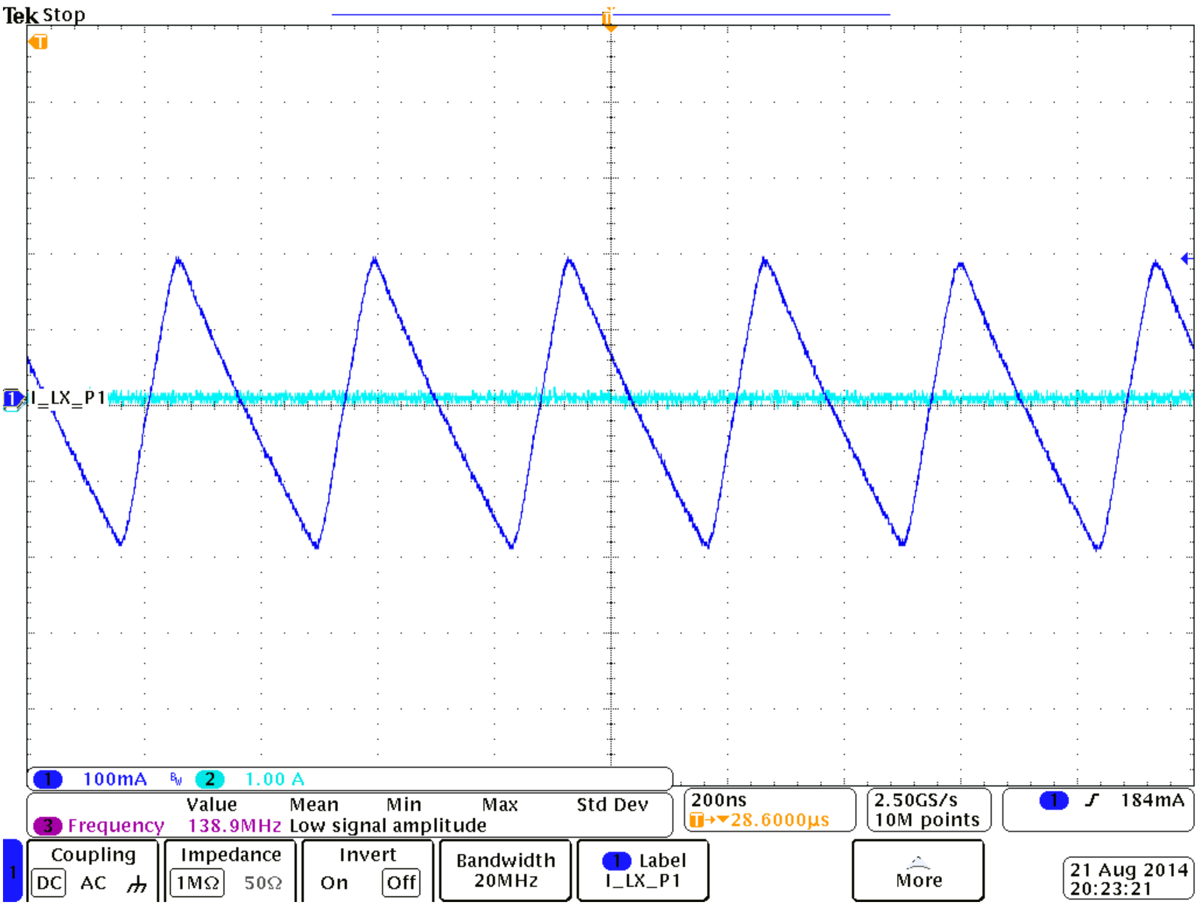
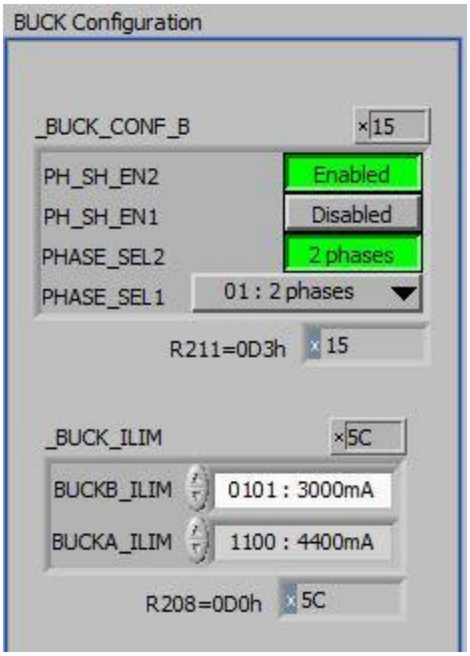


Figure 13: Inductor Current of Phase A, Phase Shedding Enabled

Let's disable the phase shedding by setting the PH_SH_EN1 bit at zero, thereby forcing both phases to be enabled. The quiescent current you measure on the multi-meter is now increased to a value close to 40 mA and the coil currents of phase 1 and 2 are now shown in Figure 14.

When the buck 1 operates with two active phases, the probe at the LX nodes shows a similar behaviour as in Figure 14, typical of a multi-phase converter.



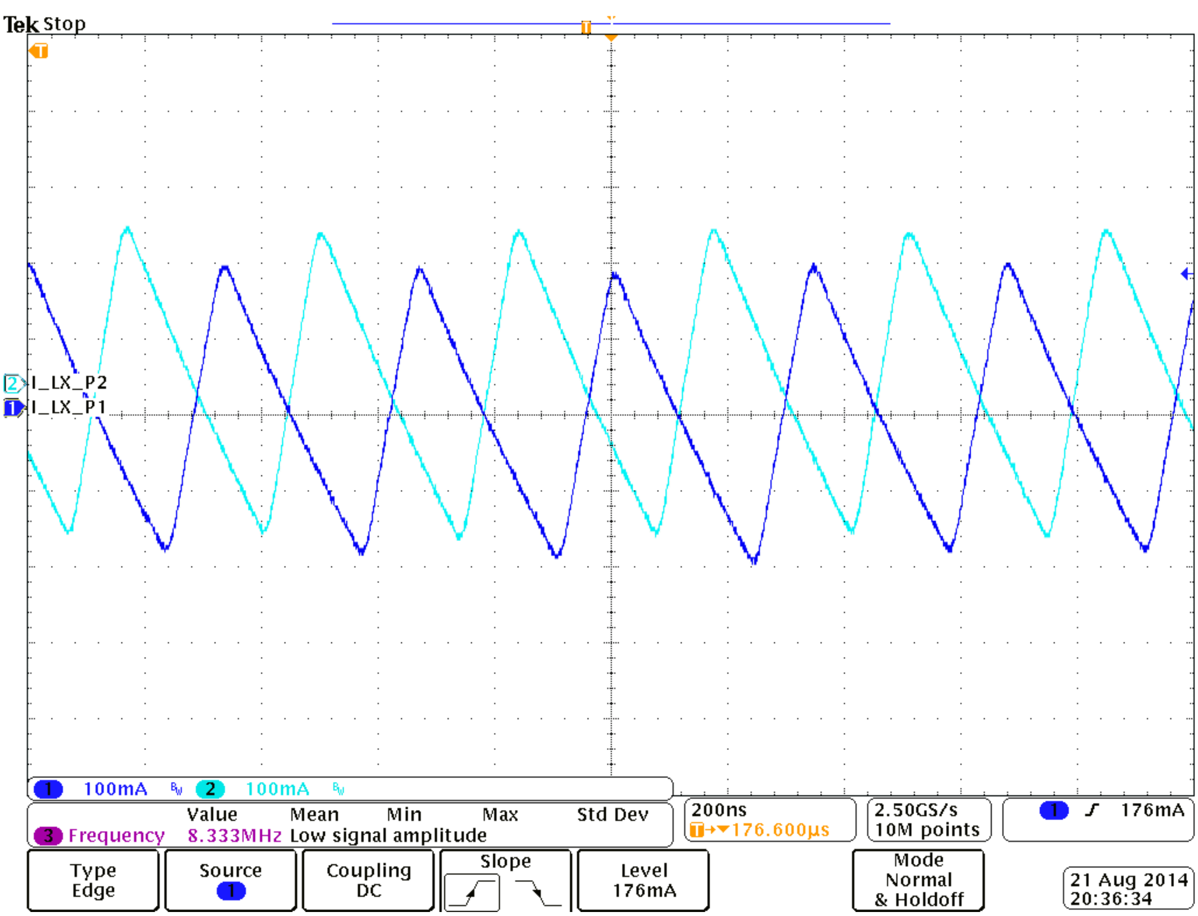


Figure 14: Inductor Current of Phase 1 and Phase2, Phase Shedding Disabled

There is an easy way you can modify the output voltage of buck 1. This is by using the VBUCK1_CTRL_A and VBUCK1_CTRL_B registers (see red box in Figure 15).

If on the BUCK_SEL field you've got VBUCK_A selected (see red arrow in Figure 15), then you'll be able to change the output voltage by changing the value in VBUCK1_CTRL_A. This initiates a DVC transition towards the new selected voltage.

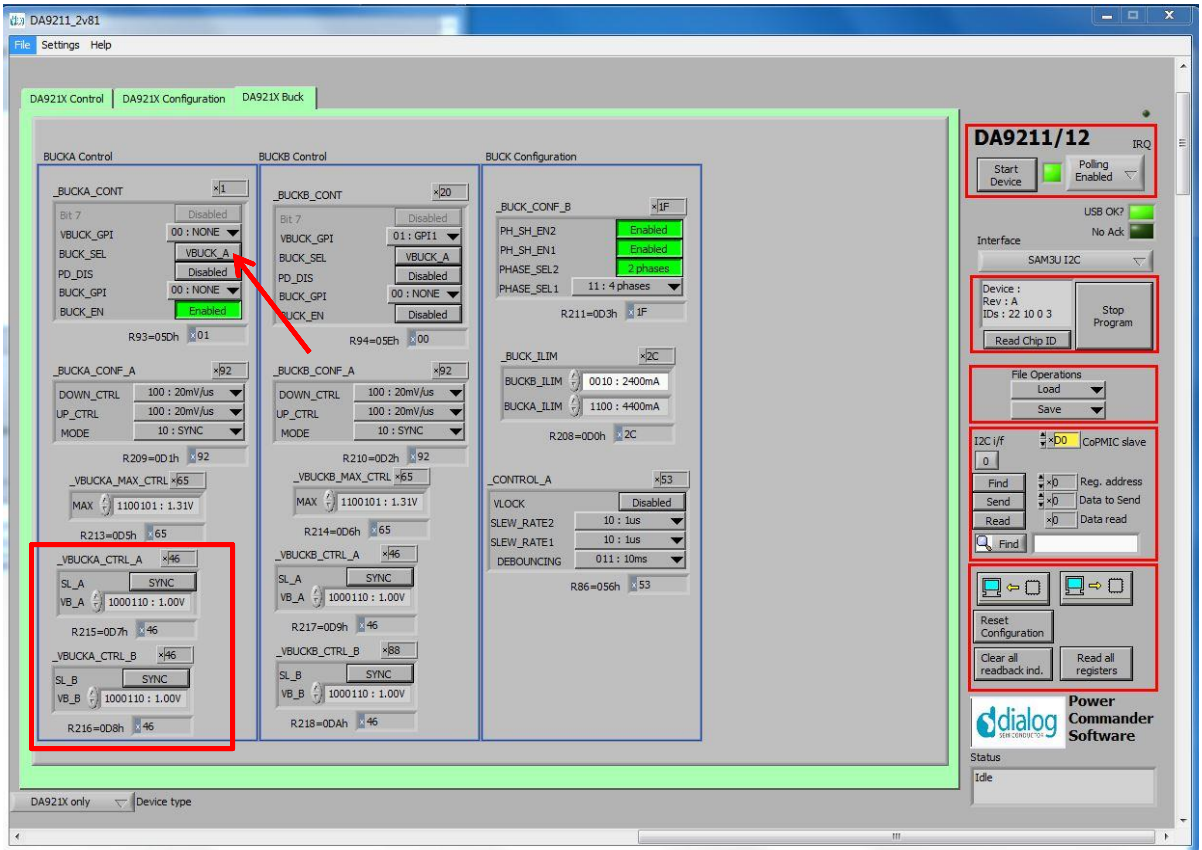


Figure 15: Output voltage control of the buck

By toggling the value on BUCK_SEL you can also change the output voltage between the value contained in VBUCK1_CTRL_A and the value contained in VBUCK1_CTRL_B.

Let's configure for instance 1.2 V in VBUCK1_CTRL_A and 0.8 V in VBUCK1_CTRL_B. After asserting the BUCK_SEL bit you'll see the following transition in the output voltage.

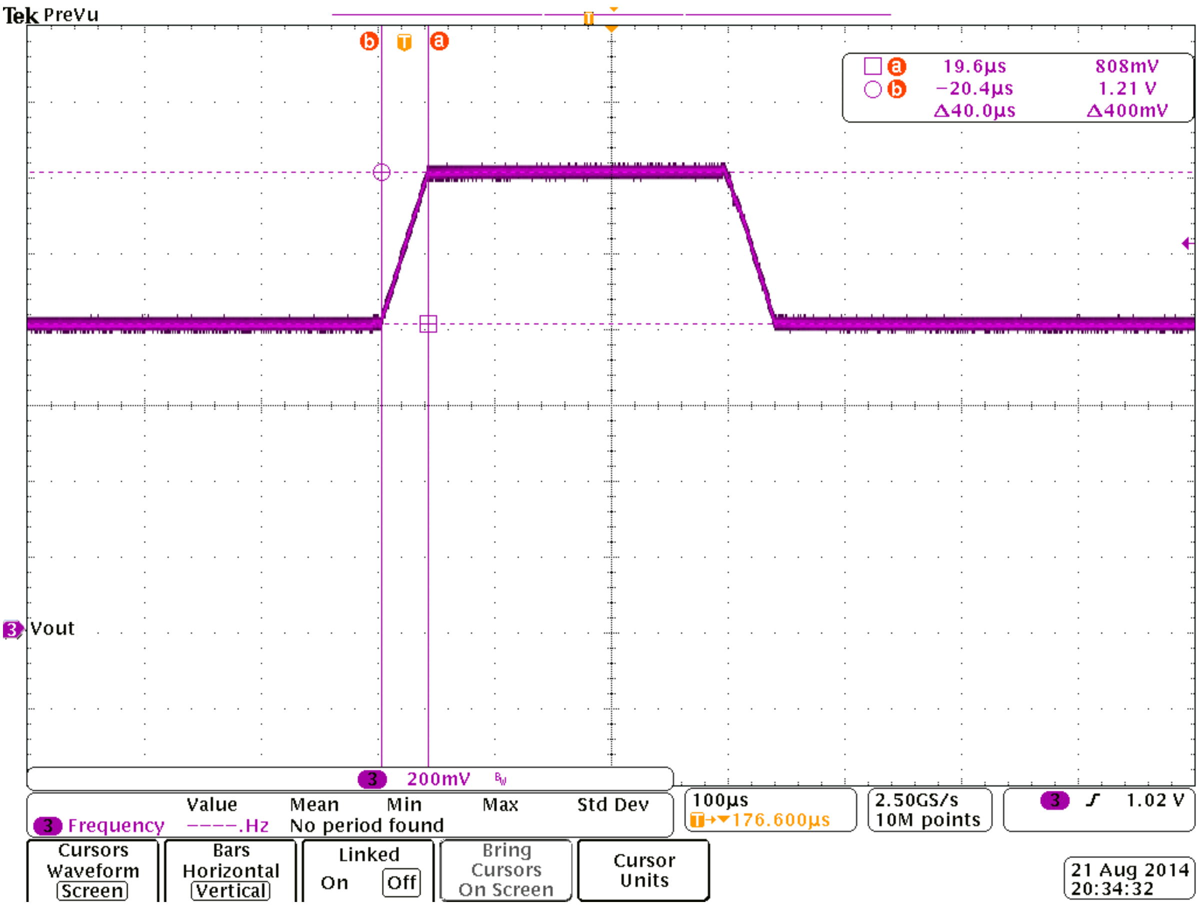
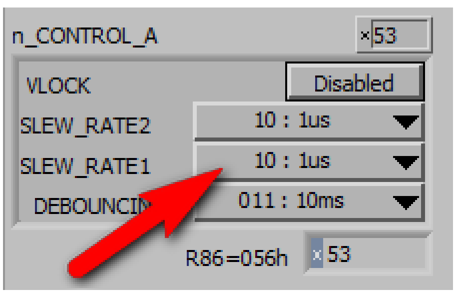


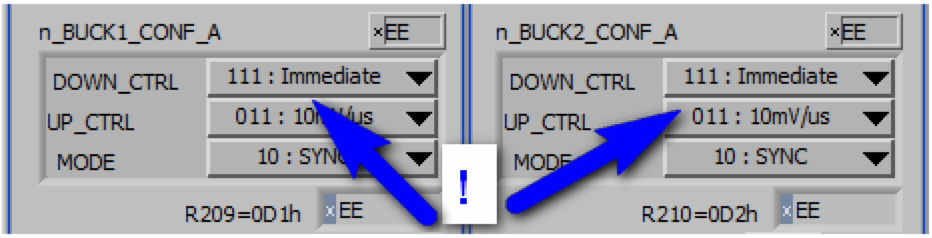
Figure 16: DVC transition of the buck

As the voltage is decreased by 400 mV in approximately 40 µs, you can clearly identify the slope of the DVC transition being 10 mV/µs.

The value can be set on the SLEW_RATE1 field of CONTROL_A register. You can find it in the "Buck" tab of the GUI software, as shown by the red arrow below.

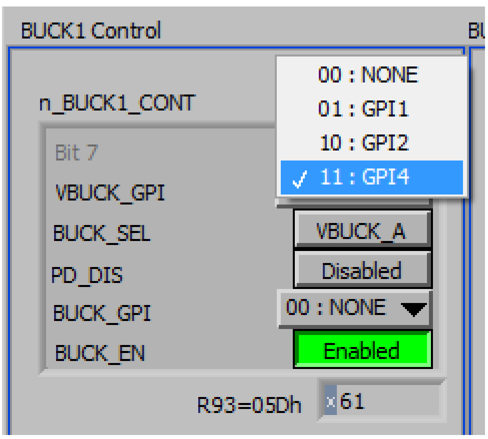
Do not confuse the slew rate setting of DVC voltage changes with the UP_CTRL, DOWN_CTRL of each buck, as they do only refer to the power up and power down slope.

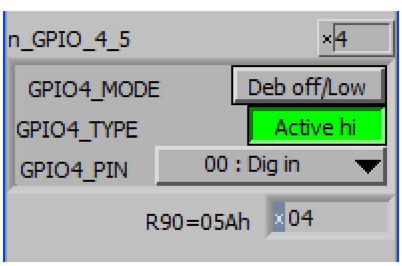




There's another possibility of controlling the output voltage of the buck converter and it is using one of the GPIO ports available on DA9212.

Let's select GPI4 on VBUCK_GPI field of BUCK1_CONT register. The GPIO4 is already selected as input port active high in the related register of DA9212 Control tab (see below).





You can simply change the connection of the green cable in Figure 17 and this will be reflected into a change of the output voltage of buck 1 between VBUCK1_CTRL_A and VBUCK1_CTRL_B. So the BUCK_SEL bit functionality has been replaced by the GPI4 input port.

The same procedure can be applied to the BUCK_GPI field, which selects a GPIO to implement the buck enable/disable functionality. By changing the connection of the selected GPIO you'll see the buck converter being enabled and disabled.

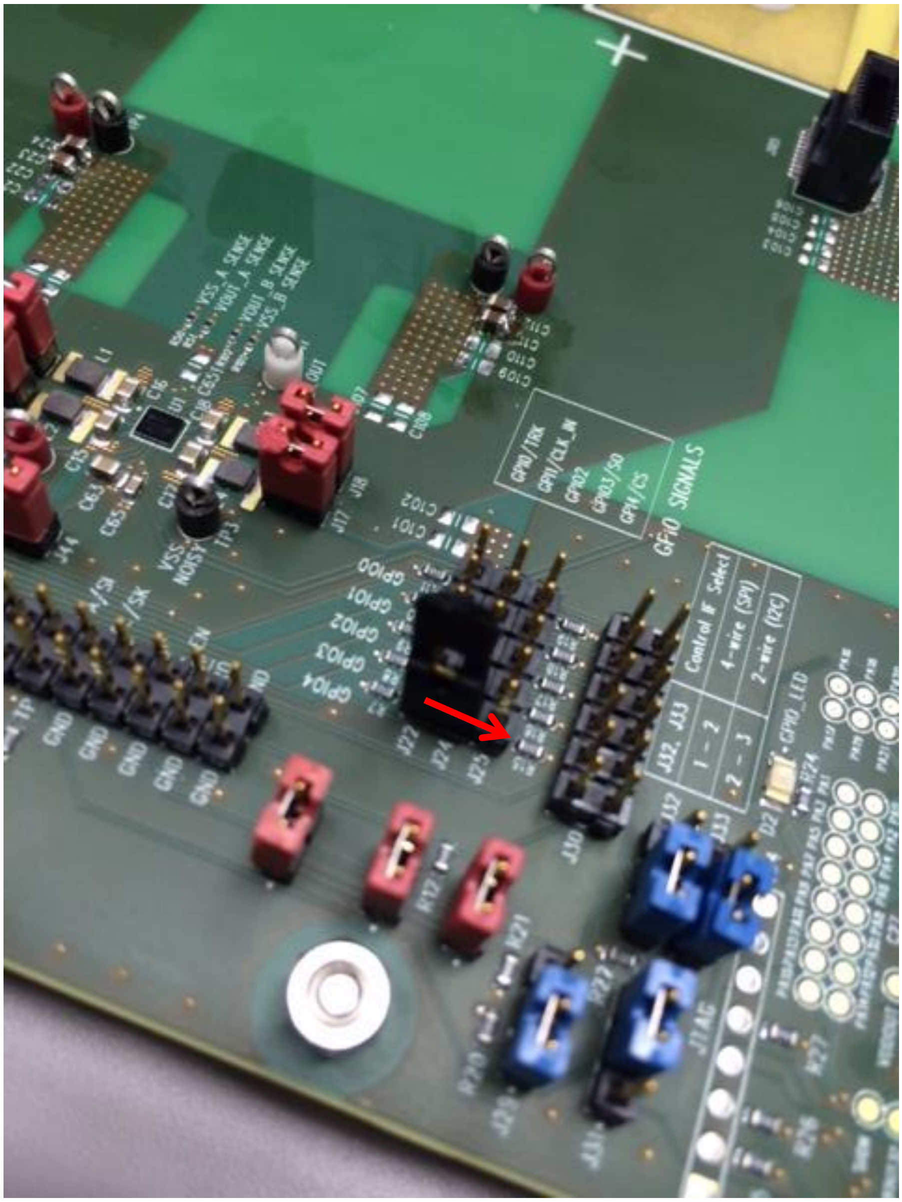


Figure 17: Port control of the buck output voltage

You may now want to turn off the whole IC and not only the buck converter. This is achieved via the IC_EN port (see also jumper link J29). If you toggle the position of the jumper to the alternative position, you switch off completely the whole IC (not only the buck converter) and the current measured from the supply is close to zero.

That's probably enough for now.

DA9212 is a very powerful and flexible device, so it is possible that issues will arise through continued and personalised usage. For questions and clarification please refer to your local Dialog support team.

10. Appendix: scripting

The software includes the ability to save and load a text file containing hex codes representing the register addresses and data. This allows a basic scripting functionality on DA9212 for advanced users.

If you wish to directly use register names (easier to read), instead of register addresses, go to the menu and select “Reg names in file” under the option “Settings”. In this case the slave address is replaced by the word “WRITE2”. This is generally preferable and more readable. The register names are stored in an external file. Hex codes for slave address and register address are still accepted on reading in the file.

Note that the suffix “2” is used to identify coPMIC devices like DA9212 and distinguish them from main PMIC devices like DA9063. Although they can share the I2C address if STAND_ALONE is set to zero (see Datasheet for details on STAND_ALONE bit operation), the GUI offers the possibility to assign different I2C addresses to main and coPMIC devices. Thus “WRITE” automatically uses the main PMIC I2C address and “WRITE2” uses the coPMIC I2C address assigned (see section 7.2.3).

10.1 Text File Format

- 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 8 bit format.
- Alternatively the first parameter can be a token:
  - “WRITE2” will write to the device at the currently selected slave address (I2C mode only)
  - “READ2” will read from the 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 be repeated the specified number of times.

- If you wish to use the GPIO-related commands, you need to jumper the respective jumpers of J30, to allow the ATSAM3U µC device to really operate on the specific port. The levels are not forced by DA92112 but instead this is done externally by the ATSAM3U.
- The second parameter is the register address as a name or hex value.
- The third parameter is the data.
- Comments (i.e. lines beginning with ‘//’) are permitted in the file.
- Inline comments (i.e. //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 I2C bus to be controlled.
- The token PORT2 will allow control over the GPIOs which are configured as inputs. The second parameter is the port name. The third parameter is 0 or 1.
- For read operations, the result of the read is passed to the history log window

10.2 Examples

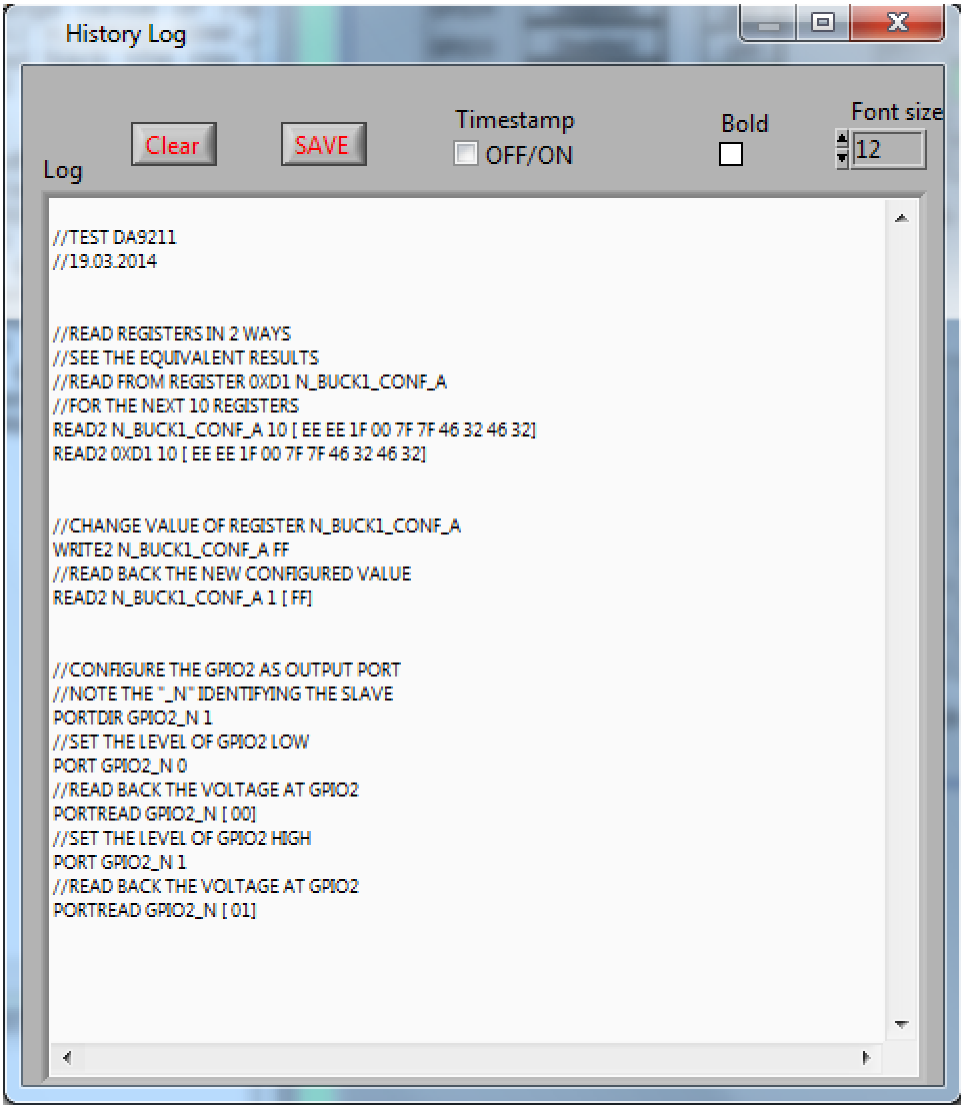
```
//Test DA9212
//19.03.2014


//Read registers in 2 ways
//See the equivalent results
//Read from register 0xD1 n_BUCK1_CONF_A
//for the next 10 registers
READ2 n_BUCK1_CONF_A 10
READ2 0xD1 10


//Change value of register n_BUCK1_CONF_A
WRITE2 n_BUCK1_CONF_A FF
//Read back the new configured value
READ2 n_BUCK1_CONF_A 1


//Configure the GPIO2 as output port
//Note the "_N" identifying the slave
PORTDIR GPIO2_N 1
//Set the level of GPIO2 low
PORT GPIO2_N 0
//Read back the voltage at GPIO2
PORTREAD GPIO2_N
//Set the level of GPIO2 high
PORT GPIO2_N 1
//Read back the voltage at GPIO2
PORTREAD GPIO2_N
```

The results in the history log can be seen by selecting Settings > History Log.


History Log
Clear
SAVE
Timestamp
OFF/ON
Bold
Font size
12
Log
//TEST DA9211
//19.03.2014
//READ REGISTERS IN 2 WAYS
//SEE THE EQUIVALENT RESULTS
//READ FROM REGISTER 0XD1 N_BUCK1_CONF_A
//FOR THE NEXT 10 REGISTERS
READ2 N_BUCK1_CONF_A 10 [EE EE 1F 00 7F 7F 46 32 46 32]
READ2 0XD1 10 [EE EE 1F 00 7F 7F 46 32 46 32]
//CHANGE VALUE OF REGISTER N_BUCK1_CONF_A
WRITE2 N_BUCK1_CONF_A FF
//READ BACK THE NEW CONFIGURED VALUE
READ2 N_BUCK1_CONF_A 1 [FF]
//CONFIGURE THE GPIO2 AS OUTPUT PORT
//NOTE THE "_N" IDENTIFYING THE SLAVE
PORTDIR GPIO2_N 1
//SET THE LEVEL OF GPIO2 LOW
PORT GPIO2_N 0
//READ BACK THE VOLTAGE AT GPIO2
PORTREAD GPIO2_N [00]
//SET THE LEVEL OF GPIO2 HIGH
PORT GPIO2_N 1
//READ BACK THE VOLTAGE AT GPIO2
PORTREAD GPIO2_N [01]

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

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including without limitation the specification and the design of the related semiconductor products, software and applications.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Customer notes that nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document are subject to Dialog Semiconductor's Standard Terms and Conditions of Sale, unless otherwise stated.

RoHS Compliance

Dialog Semiconductor complies to Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003 concerning Restriction of Hazardous Substances (RoHS). Dialog Semiconductor's statement on RoHS can be found on the customer portal https://support.diasemi.com/. RoHS certificates from our suppliers are available on request.

Contacting Dialog Semiconductor

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

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

The Netherlands
Dialog Semiconductor B.V.
Phone: +31 73 640 88 22

Email:
enquiry@diasemi.com

North America
Dialog Semiconductor Inc.
Phone: +1 408 727 3200

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

Taiwan
Dialog Semiconductor Taiwan
Phone: +886 226 580 388

Web site:
www.dialog-semiconductor.com

Singapore
Dialog Semiconductor Singapore
Phone: +65 64845419

China
Dialog Semiconductor China
Phone: +852 2607 4271

Korea
Dialog Semiconductor Korea
Phone: +82 2 569 2301

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/.