Application Note

Booting Options for DA14531

AN-B-072

Abstract

This document describes the hardware and software setup using the DA145xx-pro 376-18-B development kit for available booting options using serial interfaces such as I2C, UART, SPI and how to program a firmware into Flash, EEPROM or OTP memory.

The mechanisms explained in this document are also valid for a custom board.

Contents

Figures

 © 2020 Dialog Semiconductor

Tables

1 Terms and Definitions

SoC	System on Chip
POR	Power on Reset
BLE	Bluetooth Low Energy
OTP	One-Time Programmable
HCI	Host Controller Interface
GTL	Generic Transport Layer

2 References

[1] DA14531 Datasheet
[2] SmartSnippets Toolbox User Manual
[3] AN-B-075 DA14531 Hardware Guidelines

 © 2020 Dialog Semiconductor

3 Introduction

The document gives an overview of the booting options available for the DA14531 device using the DA145xx DEVKT-P PRO-Motherboard. The booter will be executed when a POR or a HW Reset occurs or a RESET_ON_WAKEUP event happens which is a configurable feature. This document describes the available methods to connect, boot and program the external Flash/EEPROM or the internal OTP.

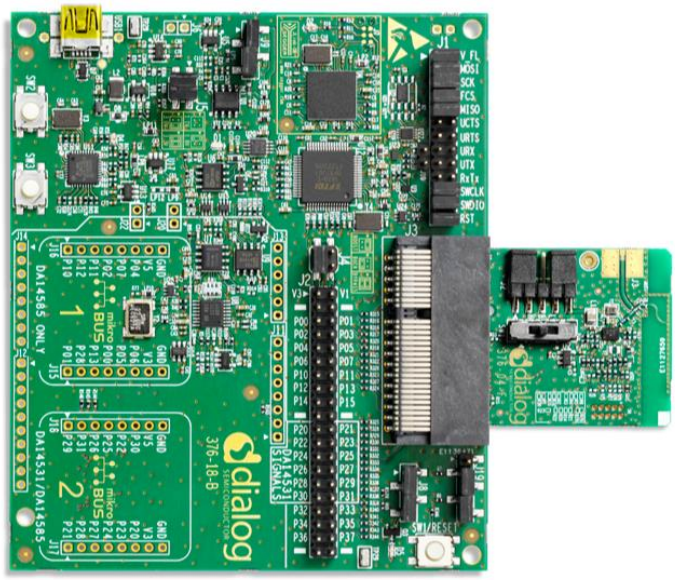


Figure 1: DA145xx DEVKT-P PRO-Motherboard + DA14531 daughterboard

4 Booting Sequence and booting pins

The booting sequence is shown in **Table 1** along with the corresponding booting pins that are used. Please refer to **Booting from serial interfaces** chapter to know more on the booting sequence and booting protocols.

Table 1: Booting sequence and booting pins for silicon DA14531AE(WLCSP17) & DA14531 AE(FCGQFN24) SoC

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
	Boot from ext SPI Master	**Boot from 1 wire UART (1st option)**	**Boot from 1 wire UART (2nd option)**	**Boot from 2 wire UART**	**Boot from ext SPI Slave**	**Boot from I2C**
P0_0/RST	MISO			Tx	MOSI	
P0_1	MOSI			Rx	SCS	
P0_2						
P0_3	SCS		RxTx		MISO	SDA
P0_4	SCK				SCK	SCL
P0_5		RxTx				

Note: The booter will try to boot from the above serial interfaces in the order from Step 1 to step 6.

© 2020 Dialog Semiconductor

4.1 Timing diagram of the booting sequence

The BootROM code execution time for booting from an external serial device is **17.3ms**. At the end of the scanning sequence the device will halt with the JTAG enabled (if there is no disabling from the configuration script). Figure 2 displays the power-up sequence of the DA14531 and pins P0_0 to P0_5, which are involved in the 6 steps as explained in booting sequence Table 1.

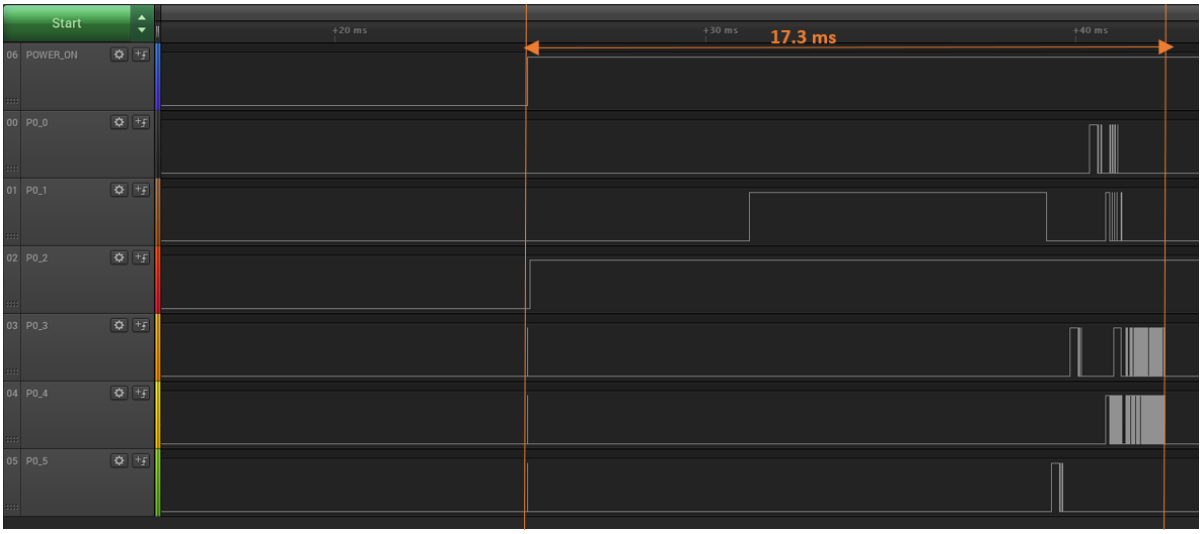


Figure 2: Scan timing for booting from external serial devices

Figure 3 gives the overview of the booting steps from step 1 to step 6.

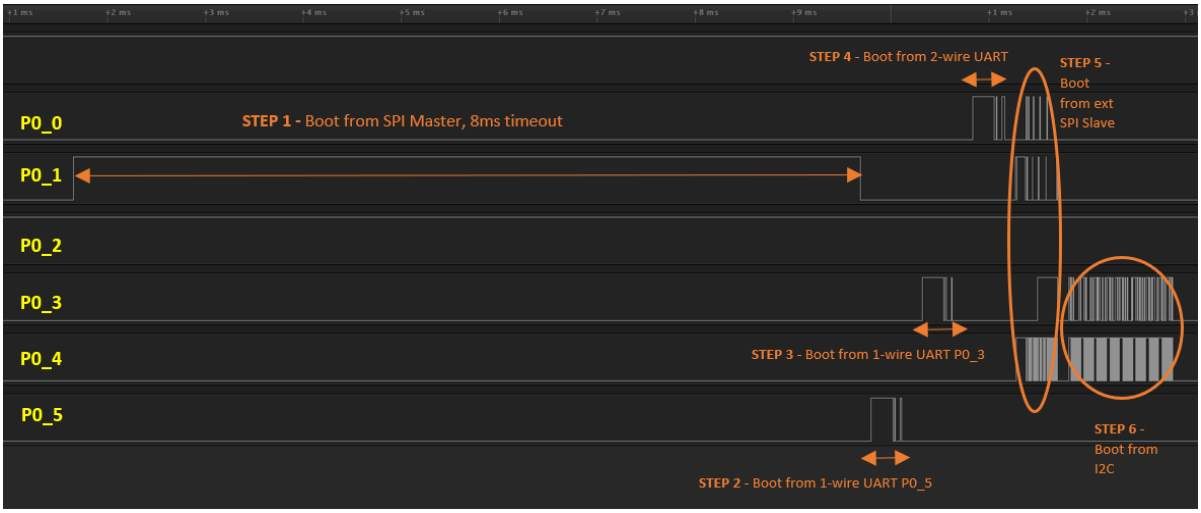


Figure 3: Step 1 to step 6 of the booting sequences zoomed in picture

Figure 4 shows the scan timing from the external devices when there are no external devices available on any of the possible interfaces to boot from. The approximate time to boot from each peripheral is also noted.

© 2020 Dialog Semiconductor

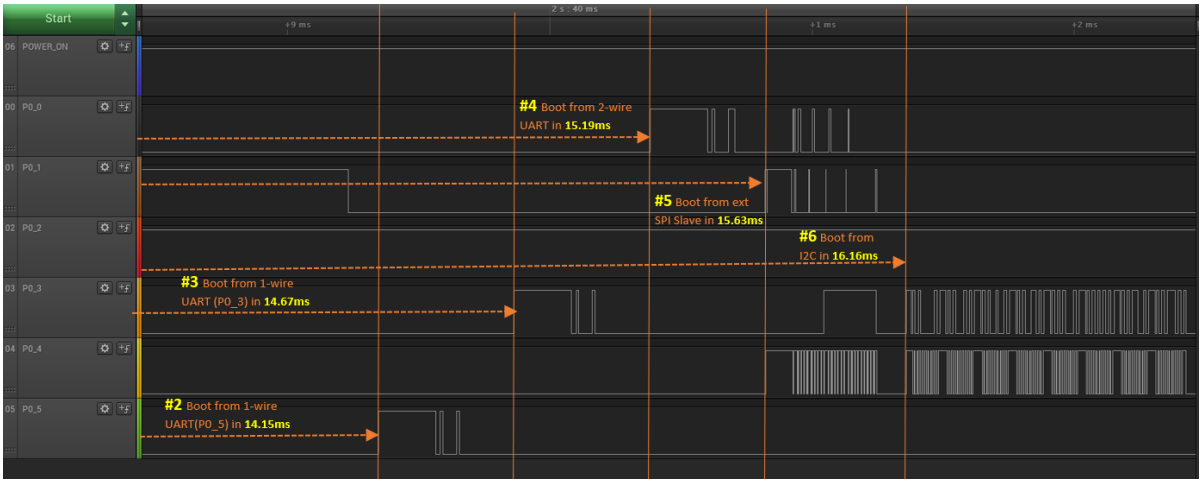


Figure 4: Timing diagram

As seen in the above Timing diagram, step 1 could not be captured in the zoomed in figure. Please refer to Figure 3 for the capture. The booting of DA14531 is incredibly fast and the above timing diagram can help the developer to select the booting most suitable for their application.

4.2 Reset functionality in DA14531

The reset functionality POR (Power-On Reset) or HW reset (hardware) on P0_0 is multiplexed with SPI Slave MOSI and UART Tx. It is disabled when the P0_0 is used as a peripheral pin for 2-wire UART or SPI. Please refer to Figure 2 to see the behavior P0_0 during booting.

After the booting, the reset functionality on the P0_0 is restored.

To configure the functionality of triggering a POR by a GPIO pin, follow the steps below:

1. Select a GPIO to be set as the POR source by programming POR_PIN_REG[POR_PIN_SELECT].
2. Set up the input polarity of the GPIO that causes POR by programming POR_PIN_REG[POR_PIN_POLARITY].
3. Configure the time for the POR to happen by programming POR_TIMER_REG[POR_TIME]. The default time is around three seconds.

To be able to enable POR on a GPIO from the application software, the SDK has a function that can be used for this.

```
GPIO_EnablePorPin(GPIO_PORT port, GPIO_PIN pin, GPIO_POR_PIN_POLARITY polarity,
uint8_t por_time);
```

port - GPIO port

pin - GPIO pin

polarity - GPIO port pin polarity. Active low = 0, Active high = 1.

por_time - Time for the Power-On Reset to happen. The time is calculated based on the following equation: Time = por_time x 4096 x RC32 clock period

Example: To configure GPIO PORT 0 and PIN 7 (P0_7) as POR with a default POR time as 3 seconds and setting the polarity as active high,

```
GPIO_EnablePorPin(GPIO_PORT_0, GPIO_PIN_7, 1 , 0x18);
```


© 2020 Dialog Semiconductor

Calling this function, say after the `system_init()` will enable P0_7 to be a POR GPIO.

```
int main(void)
{
    sleep_mode_t sleep_mode;
    // initialize retention mode
    init_retention_mode();
    //global initialise
    system_init();
    GPIO_EnablePorPin(GPIO_PORT_0, GPIO_PIN_7, 1 , 0x18);
    .
    .
}
```

To trigger a POR via P0_7 at application code run time, connect a with a fly wire from P0_7 to J2[V3] of the motherboard for more than 3seconds. Then the device will reset with POR event.

Please be aware if a GPIO is used as a POR source, the dynamic current of the system increases due to the dynamic current consumed by the RC32k oscillator. This increase is estimated to be from 100 nA to 120 nA and it is also present during sleep time period. POR from the RST pad does not add this dynamic current consumption.

Please note the POR functionality is not supported in the hibernation mode.

For more details on the reset pin mapping on DA14531 refer the Chapter 5 of the datasheet.

5 Bypass mode configuration

When the DA14531 is configured to run in Buck or Boost mode, the SDK auto-configures the mode depending on the power supply scheme (jumper settings on J4 on the motherboard). However, to run in Bypass mode, the appropriate flag needs to be set in the SDK in the da1458x_config_basic.h file of the project. This will configure the device to Bypass mode.

```
#define CFG_POWER_MODE_BYPASS
```

The jumper settings on J4 jumper for Buck or Boost mode is described in Chapter 6 *How to Configure Power of DA14531 DK PRO* of UM-B-114

The jumper settings that goes in the motherboard on J4 for the Bypass mode is as shown in Figure 5

© 2020 Dialog Semiconductor

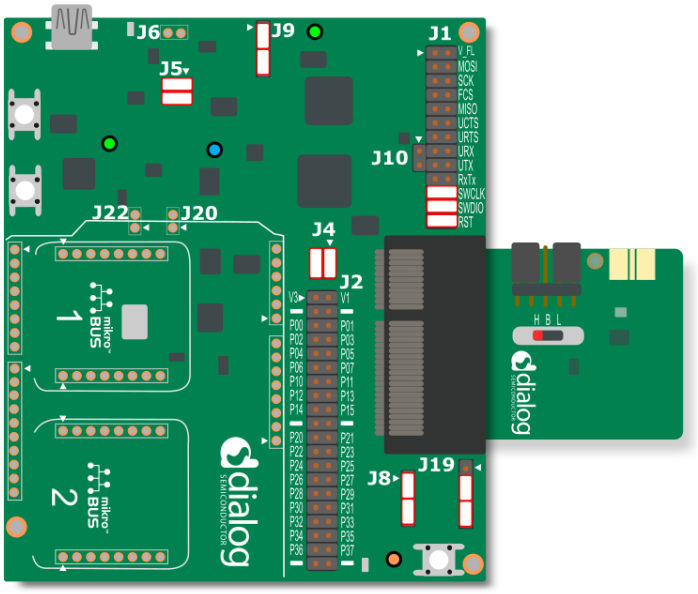


Figure 5: Jumper settings for Bypass mode on J4

6 Booting from 1-wire UART

The bootloader of the DA14531 supports 1-wire UART interface on pin P0_5 or P0_3 which is step 2 or step 3 of the booting sequence. To boot from P0_3, connect the FTDI RX and TX via a 1k resistor and connect the RX side of FTDI to the P0_3 pin.

The jumper settings to use the 1-wire UART are as shown in the Figure 6. They are marked as white box with red outline. The jumpers are set to enable the 1-wire UART functionality along with SWD but not the Flash. Using flash and 1-wire UART is described in the coming sections.

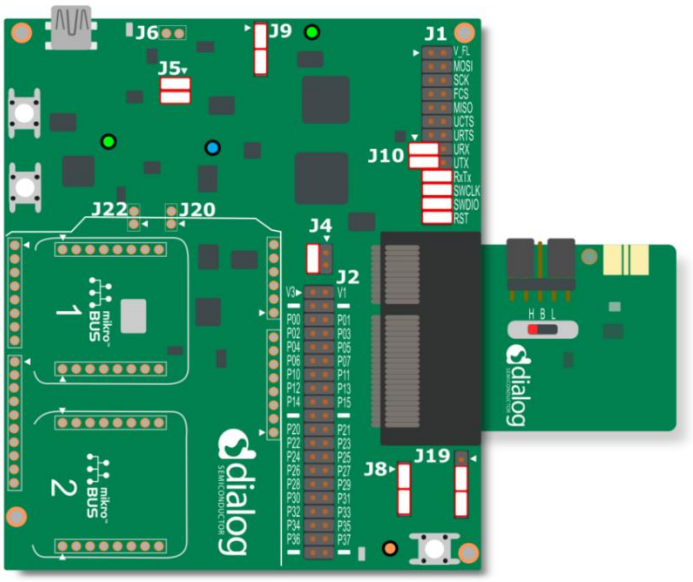


Figure 6: Jumper settings for 1-wire UART

© 2020 Dialog Semiconductor

Booting on the 1 wire UART using the SmartSnippets Toolbox requires following settings,

- Select the following SmartSnippet settings as shown in Figure 7 to boot from UART

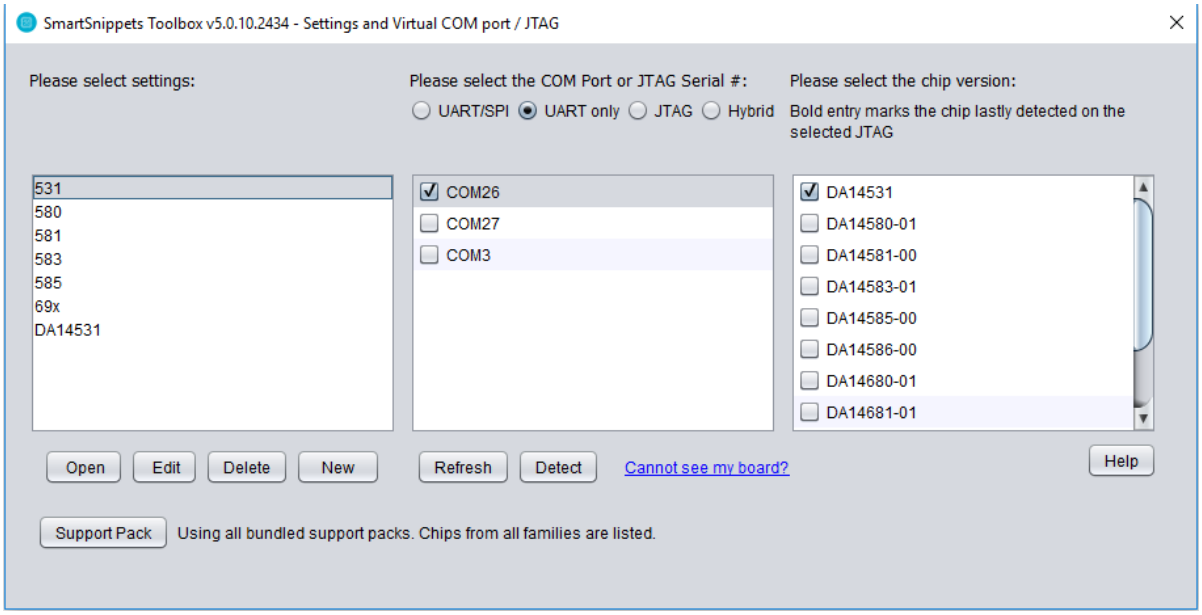


Figure 7: SmartSnippet Toolbox settings

- Select P0_5, P0_5, 115200 Bd in the Board Setup and in the Booter browse through the binary you want to download to SysRAM as shown in the figure below,

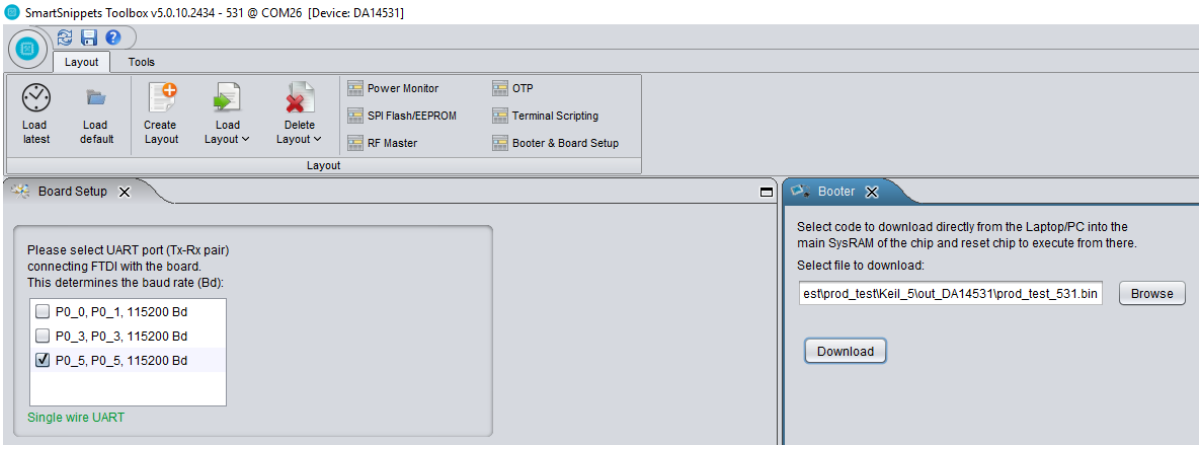


Figure 8: Board Setup and Booter

- Check the logs to confirm the firmware download as shown in Figure 9 and now the device will boot from 1-wire UART.

© 2020 Dialog Semiconductor

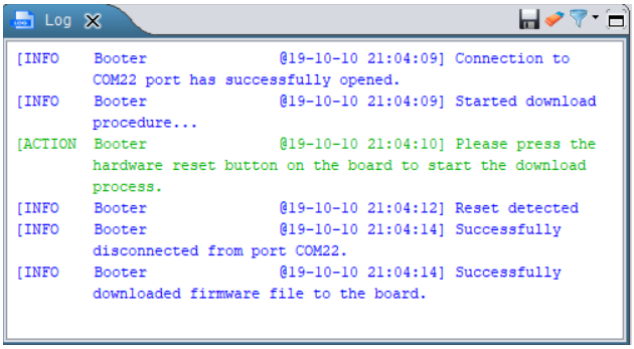


Figure 9: 1-wire UART booter logs

7 Boot from 2-wire UART

The bootloader of the DA14531 supports 2-Wire UART interface on pin P0_0 and P0_1. The hardware and software setup are like 1-wire UART except for the jumper settings and board setup as shown in Figure 10 and Figure 11.

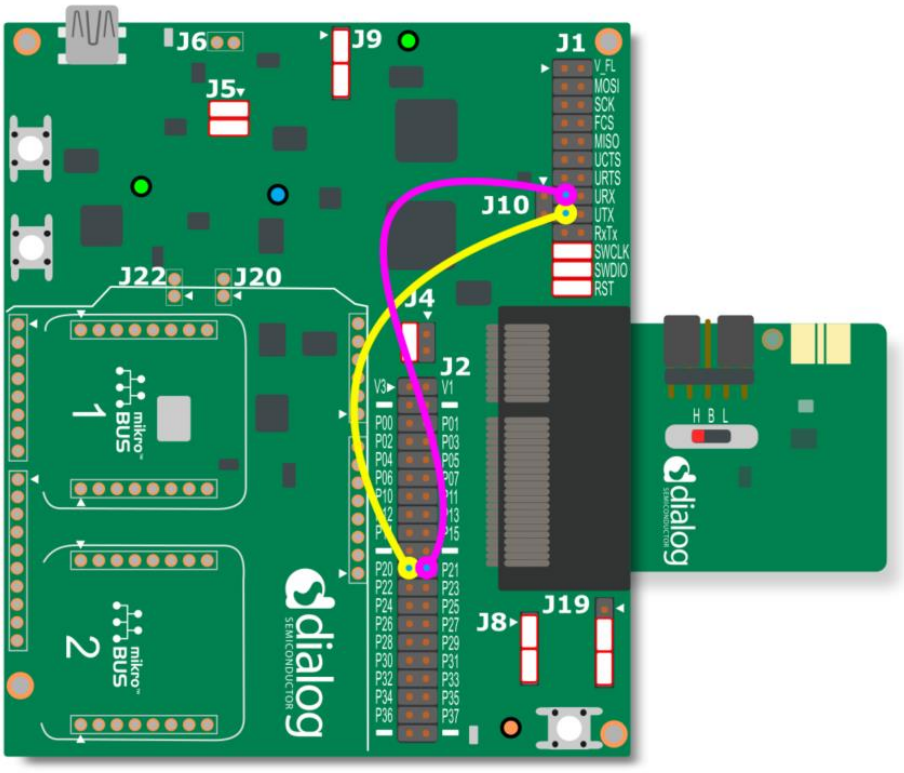


Figure 10: Jumper settings for 1-wire UART (P0_0 and P0_1)

For the 2-wire UART, the board setup is as shown below,

© 2020 Dialog Semiconductor

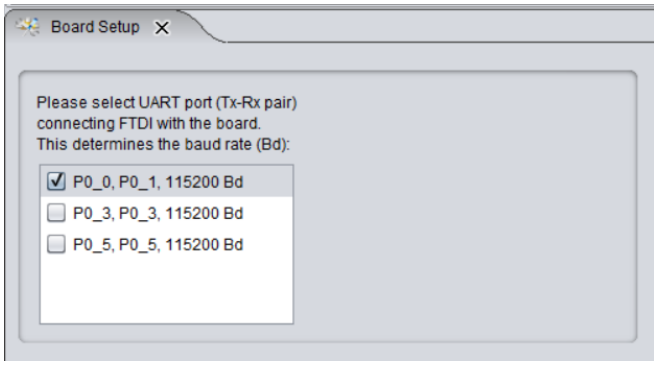


Figure 11: Board setup for 2-wire UART

The jumper settings on the motherboard from J1 to J2 header is as shown in Figure 8. The UTX (of the FTDI chip) is connected using a jumper cable to J2-P20 (P0_0/UART-Tx of the DA14531) and URX (of the FTDI chip) is connected to J2-P21 (P0_1/UART-Rx of the DA14531).

Using the Booter from the SmartSnippets Toolbox, download the firmware and the DA14531 will boot from the 2-wire UART as per the step 4 of the boot sequence.

8 Boot from SPI Slave

Booting from external SPI Slave is the 5th step in the booting sequence for the DA14531 and the sequence of events that takes place after the power up is described in the table below following with the power profile.

The following table is with respect to the BUCK mode wherein 13.1kB (BLE Barebone) application is copied from the external Slave SPI with speed of 2MHz speed.

Table 2: Boot energy - external SPI Slave

Interval	Time (ms)	Average current (mA)	Charge (uC)
Power Up and HW FSM	1.05	6.94	7.5
OTP Enabled & XTAL Settling	4.53	1.84	17.1
Booter steps 1-4	9.63	1.13	10.9
Booting from the SPI slave	57.8	3.22	189.0
Total	73.01	3.06	255.2

© 2020 Dialog Semiconductor

8.1 Programming Flash using SWD port

Use the jumper settings on the motherboard as shown in the picture below. The Flash memory connected over SPI to the DA14531 can be accessed over the SWD port. The description below will explain how to program the firmware into the Flash memory. The necessary motherboard jumper settings are shown in Figure 12.

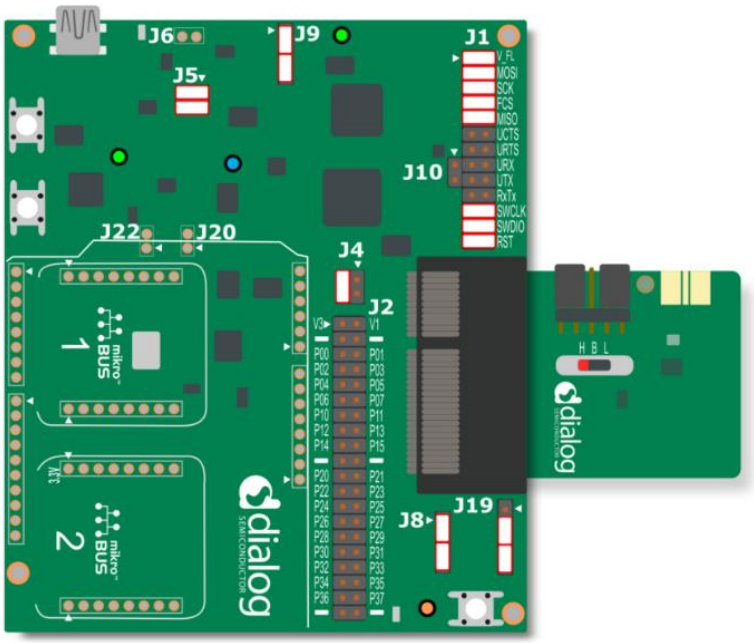


Figure 12: Jumper settings for SPI Flash

In the SmartSnippet toolbox, please select the following as shown in Figure 13 for JTAG interface,

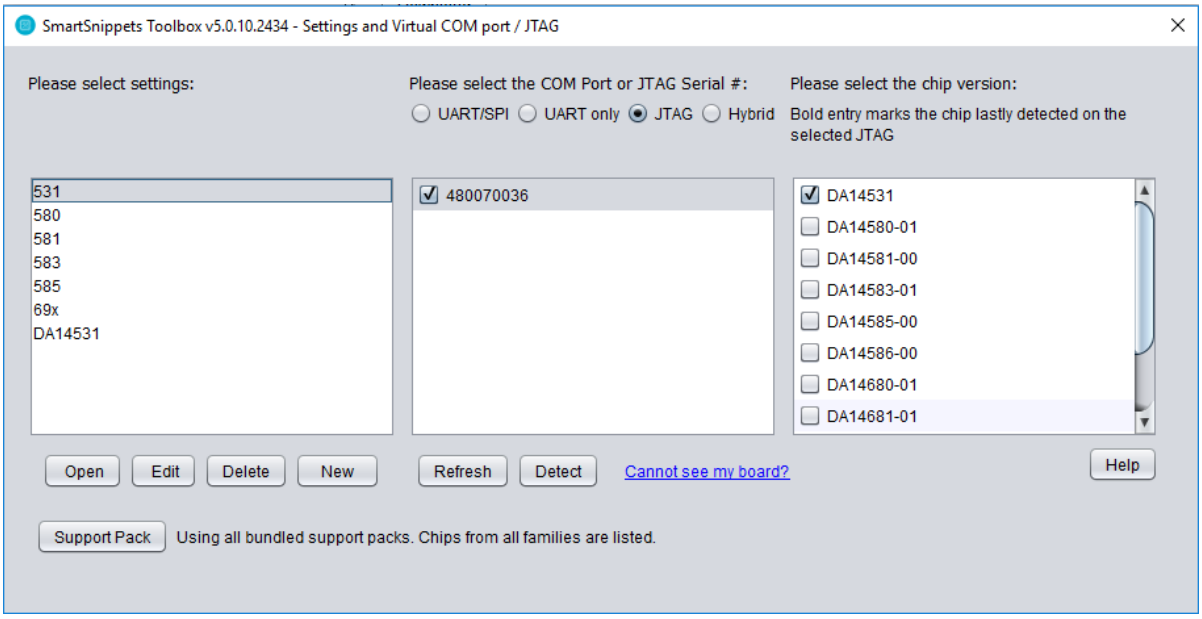


Figure 13: JTAG selection for SPI Flash

© 2020 Dialog Semiconductor

After you click on Open, do the SPI Pin configuration as shown in the Figure 14 below,

MOSI (SPI_DO) - P0_0
CS (SPI_EN) - P0_1
MISO (SPI_DI) - P0_3
CLK (SPI_CLK)- P0_4

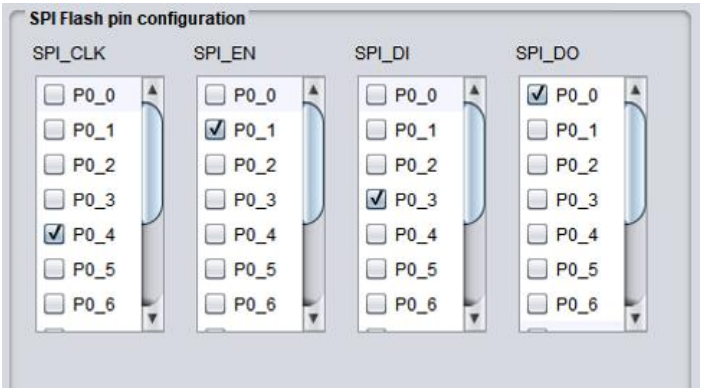


Figure 14: SPI Flash pin configuration

In the Layout tab, select SPI Flash/ EEPROM and you will SPI Flash Programmer tab opens as shown in Figure 15 below.

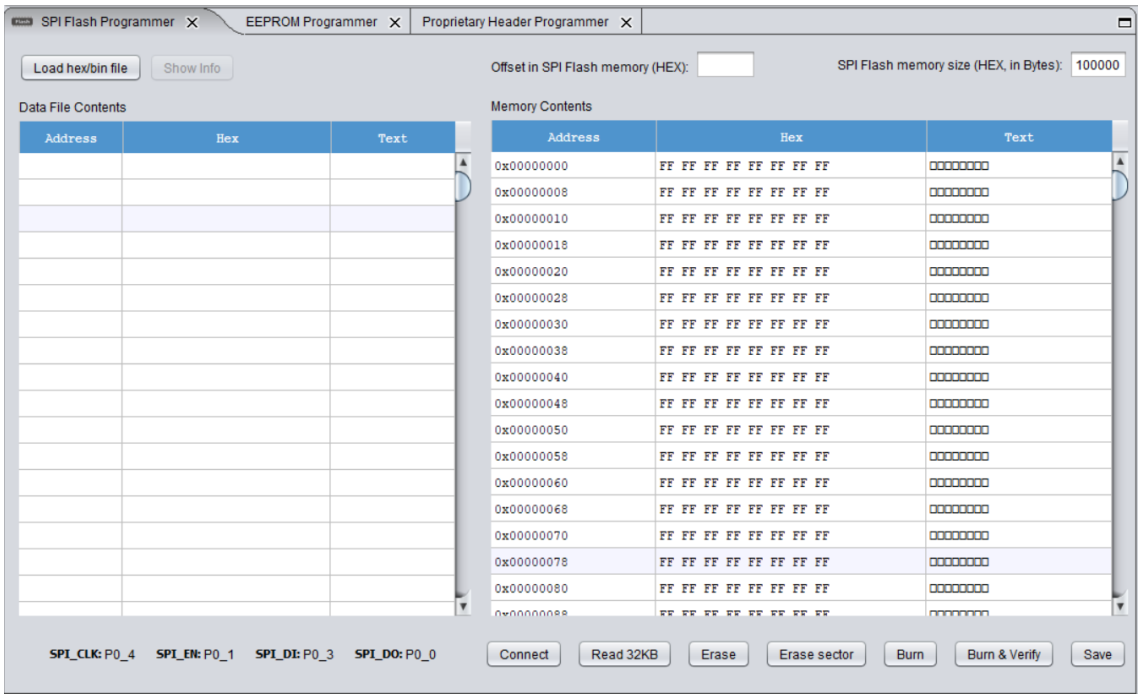


Figure 15: SPI Flash programmer to connect, read and burn the hex file

© 2020 Dialog Semiconductor

Using the **Load hex/bin file** button, browse through the location for the hex or binary file that you want to download to the device as shown below. Select **Bootable** as shown, to make the device bootable. Click on **Next** and then **Finish**.

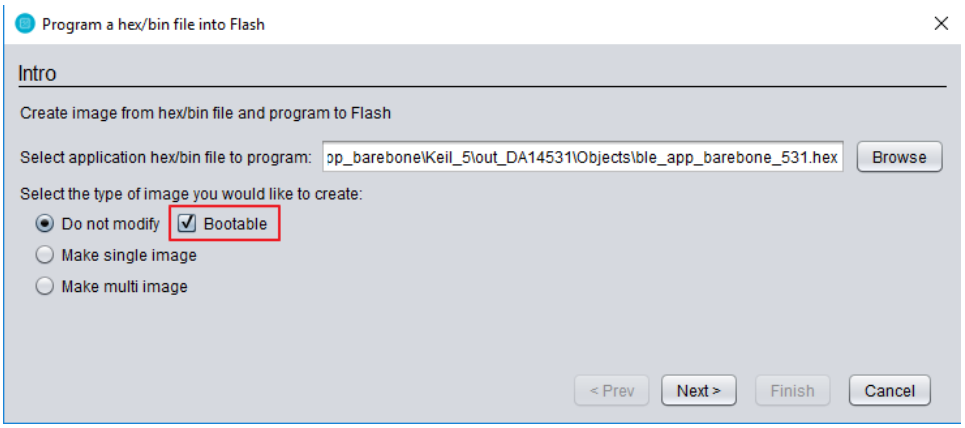


Figure 16: Program a hex/bin file into Flash

Click on **Connect** and if it prompts to press the hardware Reset button, press the reset button on the motherboard. Please look at the SmartSnippets Toolbox log to make sure the default jtag_programmer is loaded in the SysRAM.

Once connected, you can erase/read or Flash the program.

Click **Burn** to program the flash using the SPI slave interface. Check the SmartSnippets Toolbox log file to make sure the firmware is successfully programmed in the flash memory. Press the hardware RESET button on the motherboard and now the device will boot from the SPI flash.

8.2 Programming Flash using 1-wire UART

The Flash memory connected over SPI to the DA14531 can be accessed over the 1-wire UART. The description below will explain how to program the firmware into the Flash memory. To boot from SPI Slave using the 1-wire UART, the jumper settings and the board setup are shown in the Figure 17.

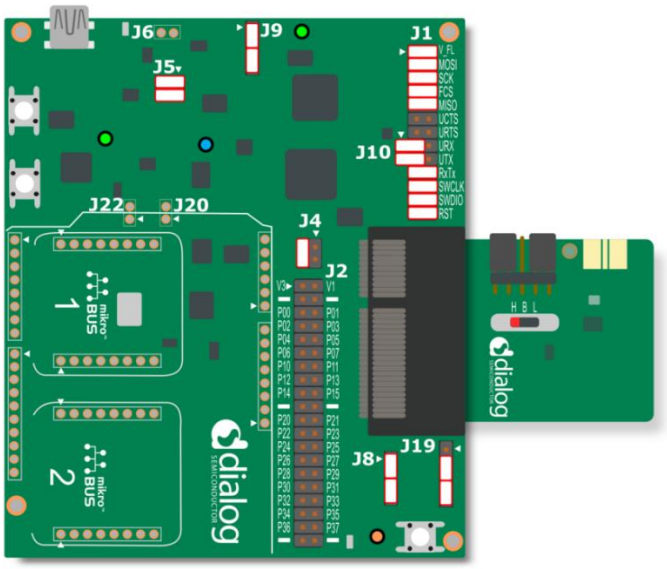


Figure 17: Jumper settings 1-wire UART with SPI Flash

© 2020 Dialog Semiconductor

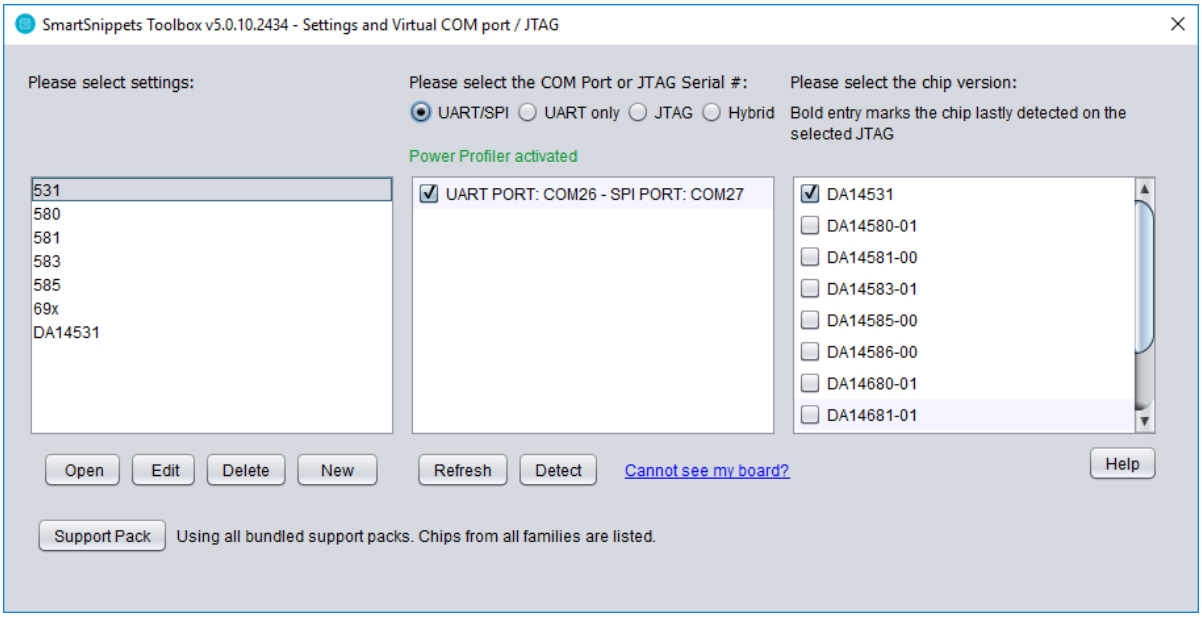


Figure 18: UART/SPI selection

Once you have these configurations, the following steps to connect/ read/ write / erase are like in Chapter 8.1

8.3 Programming Flash using 2-wire UART

The Flash memory connected over SPI to the DA14531 can be accessed over the 2-wire UART. The description below will explain how to program the firmware into the Flash memory. To boot from SPI Slave using the 2-wire UART, the jumper settings and the board setup are shown in the and below.

To boot from SPI Slave using the 2-wire UART needs a modification in the pin configuration due to the reason that the P0_0 is multiplexed between DA14531-UART Tx and SPI slave MOSI. Due to this reason we modify the pin configuration in the following way as shown below.

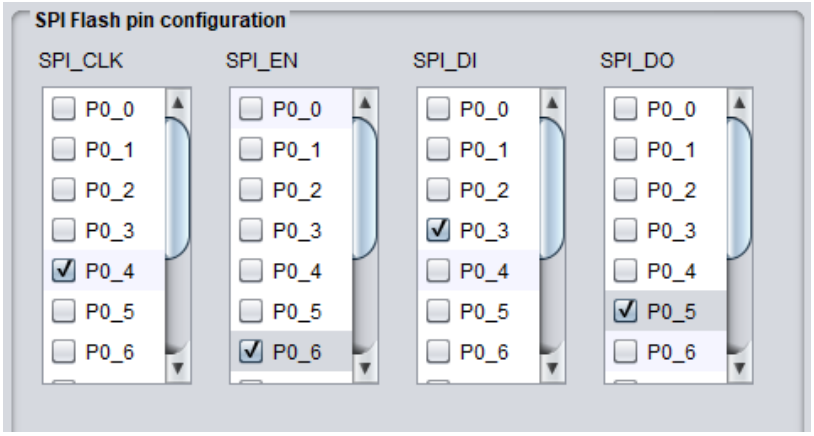


Figure 19: SPI Flash pin configuration

© 2020 Dialog Semiconductor

MOSI (SPI_DO) - P0_5
CS (SPI_EN) - P0_6
MISO (SPI_DI) - P0_3
CLK (SPI_CLK)- P0_4

The jumper setting would be as shown in Figure 20.

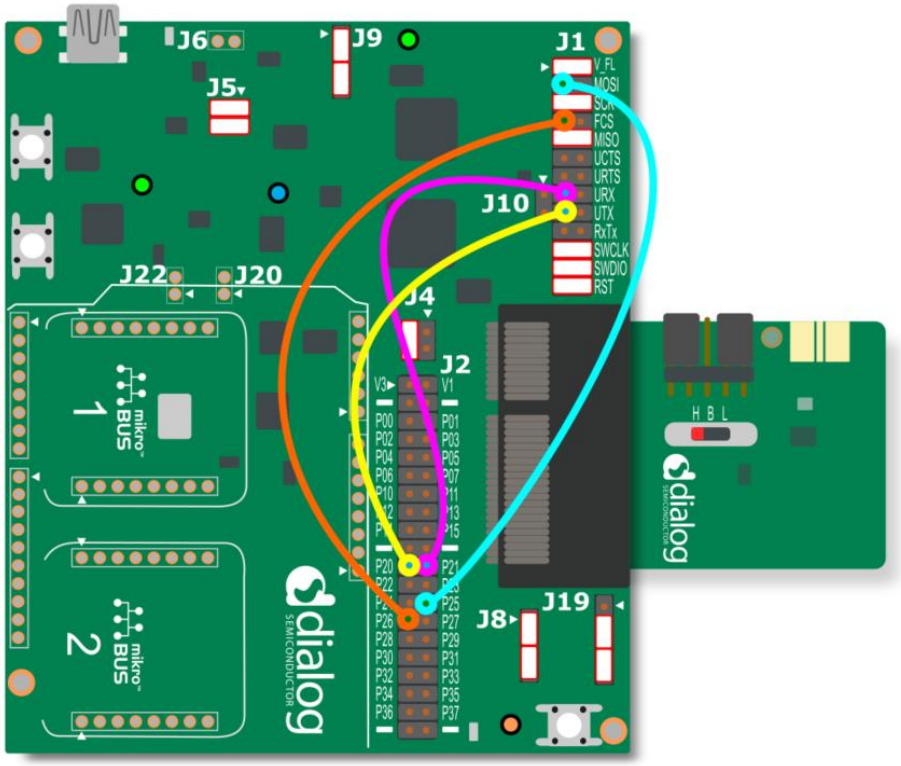


Figure 20: Jumper settings 2-wire UART with SPI Flash

Once you have these configurations, the following steps to connect/ read/ write / erase are like in **Chapter 8 above.**

9 Boot from I2C

Using the EEPROM Programmer in SmartSnippets™ Toolbox you can download the image to the EEPROM memory and boot from the EEPROM. The SDA and SCL lines to be selected in the board setup are P0_3 and P0_4 respectively.

© 2020 Dialog Semiconductor

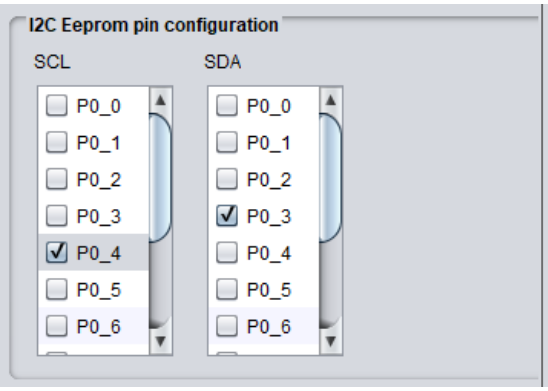


Figure 21: Toolbox I2C pin configuration

The breakout for the 531 on the Motherboard would be P2_3 and P2_4, so connect the SDA and SCL to P2_3 and P2_4 using jumper cables.
Now click on “Connect” and once connected select the firmware to download to the EEPROM memory.

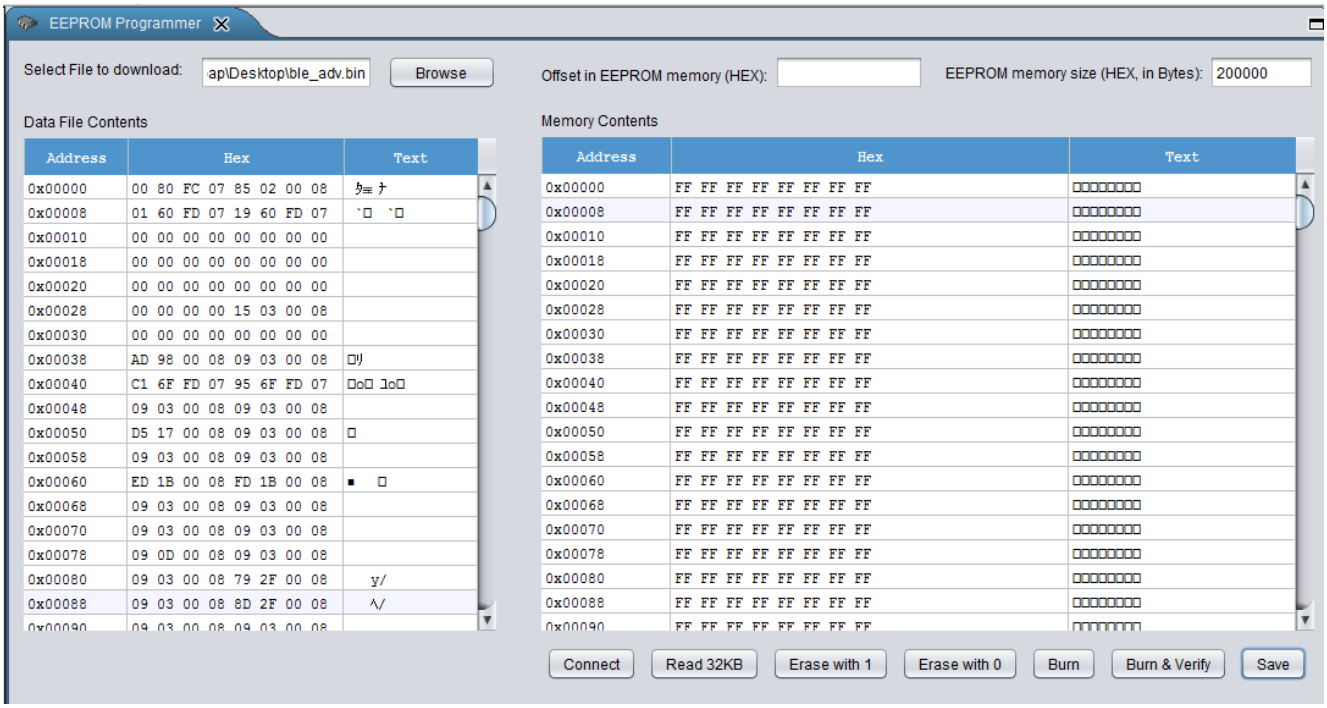


Figure 22: Toolbox EEPROM Programmer

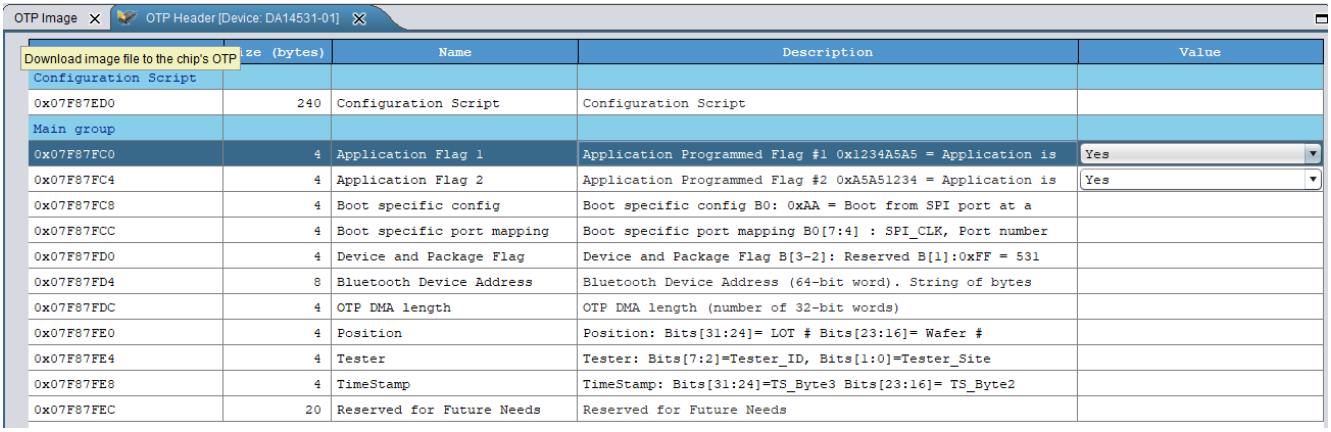
10 Boot from OTP

The OTP (One-Time Programmable) Programmer tool in the SmartSnippet Toolbox is used for burning the OTP Memory and OTP Header. Once this is programmed, it cannot be erased.

Set the right application flags to be able to boot from the OTP. Figure 23 shows the necessary header values. Once the values of the header are set, connect and burn the OTP header.

© 2020 Dialog Semiconductor

Note: The application flag 1 and application flag 2 should be set to value "Yes" as shown in Figure 23.



	ze (bytes)	Name	Description	Value
Configuration Script				
0x07F87ED0	240	Configuration Script	Configuration Script	
Main group				
0x07F87FC0	4	Application Flag 1	Application Programmed Flag #1 0x1234A5A5 = Application is	Yes
0x07F87FC4	4	Application Flag 2	Application Programmed Flag #2 0xA5A51234 = Application is	Yes
0x07F87FC8	4	Boot specific config	Boot specific config B0: 0xAA = Boot from SPI port at a	
0x07F87FCC	4	Boot specific port mapping	Boot specific port mapping B0[7:4] : SPI_CLK, Port number	
0x07F87FD0	4	Device and Package Flag	Device and Package Flag B[3-2]: Reserved B[1]:0xFF = 531	
0x07F87FD4	8	Bluetooth Device Address	Bluetooth Device Address (64-bit word). String of bytes	
0x07F87FDC	4	OTP DMA length	OTP DMA length (number of 32-bit words)	
0x07F87FE0	4	Position	Position: Bits[31:24]= LOT # Bits[23:16]= Wafer #	
0x07F87FE4	4	Tester	Tester: Bits[7:2]=Tester_ID, Bits[1:0]=Tester_Site	
0x07F87FE8	4	TimeStamp	TimeStamp: Bits[31:24]=TS_Byte3 Bits[23:16]= TS_Byte2	
0x07F87FEC	20	Reserved for Future Needs	Reserved for Future Needs	

Figure 23: Toolbox OTP Programmer

On the OTP Image tab, select the file to download and burn to OTP. After burning the header and the image, the device will be able to boot from OTP and bypass the ROM boot sequence as described in Chapter **4**.

The boot energy analysis in BUCK configuration can be found in Table 3 and the power profile from power-up until the first BLE advertisement is shown in Figure 24.

Table 3: Boot Energy OTP

Interval	Identification	Time, Charge	Description
From Power up to OTP mirroring	Hardware FSM + BootROM	1.39ms,5.95uC	The HW FSM which powers up LDOs and the DC-DC converter. The copy of the calibration values from the OTP into the registers
OTP Mirroring	OTP Mirroring	1.25ms, 2.44uC	OTP contents are copied into System RAM
From OTP Mirroring to RF calibration	Reset handler	2.6ms, 1.5uC	SW executes the reset handler
RF calibration	RF calibration	7.5ms, 7.9uC	Radio is trimmed by SW
From RF calibration to Advertising	Database initialization	11.5ms,4.2uC	SW initializes the descriptors and the database of the system.

© 2020 Dialog Semiconductor

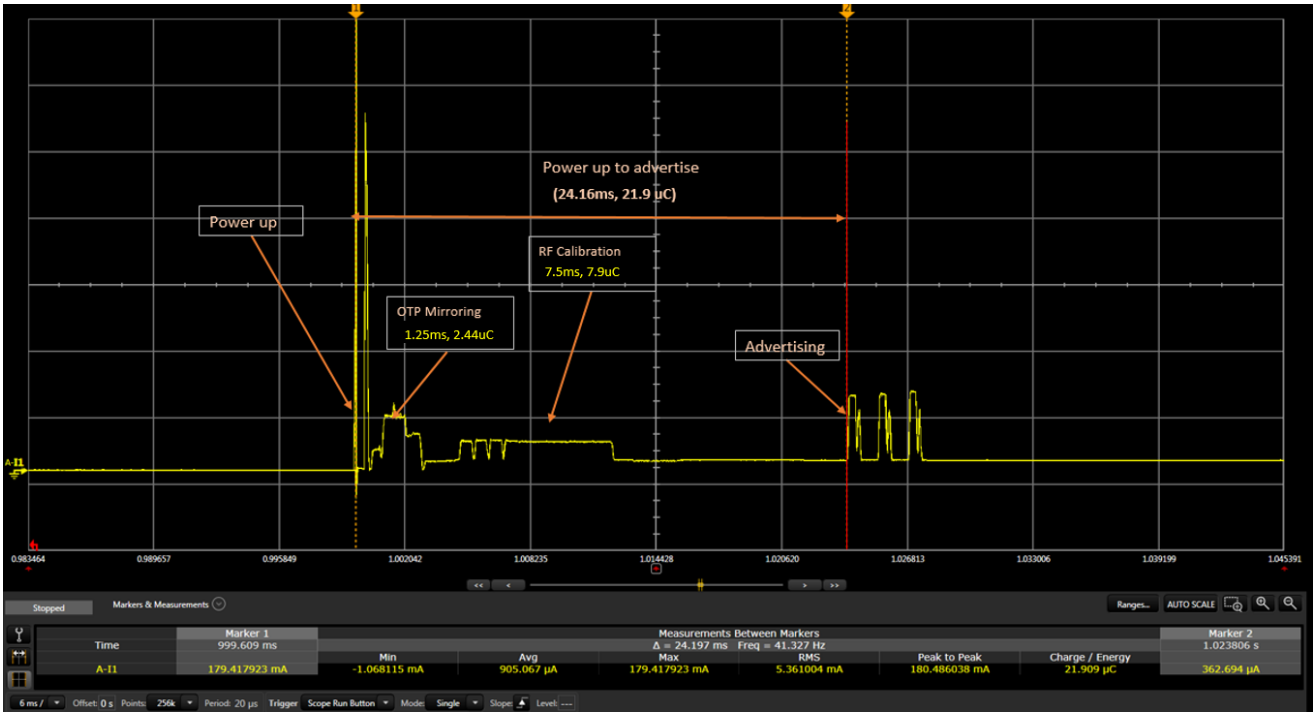


Figure 24: OTP Boot Energy Power Profile

In future revisions of the document we will add how to program the Configuration Script and use a secondary boot loader to customize the boot sequence. Please check the SmartSnippetsToolbox User Manual for more updates.

11 Booting from a custom PCB

When using a custom PCB to boot from 2-wire UART, it is advised to take care of the RST/P0_0 pin such that it doesn't continuously reset as the default setting after the boot sequence is finished is RST. The explanation related to this is given below.

Since the idle state of the UART lines are default high, FTDI devices and microcontrollers tend to have a pull up resistor on their RX line in order to avoid continuous break. Since the P0_0 is used as a TX while booting and also as a HW reset, connecting an external device might continuously reset the DA14531 due to the external pull up.

As you can see in Table 1, when booting from 2-wire UART (Step4), the Tx is mapped to P0_0 which is also used as Reset pin. The device starting its boot sequence will detect UART on P0_0/P0_1. After the boot sequence has finished the P0_0 will automatically be restored to reset mode. If a connected external microcontroller or FTDI USB to UART bridge pulls this port high, it will trigger a continuous reset.

A valid solution would be to use a stronger R pull-down resistor connected to P0_0 overriding the effect of the receiver's pull up, as shown in Figure 25. The value of R depends on the pull up on the receiver's side. This solution is acceptable if there is no constant voltage from an external device on P0_0 to leak through the strong pull down.

In case that the application requires constant connection on the P0_0/P0_1 due to external interface the user must make sure that during the booting process of the DA14531 the external controller will not have its RX pin in high state since this would cause the DA14531 to reset. . If the external processor

 © 2020 Dialog Semiconductor

can handle the fact that its RX line will be low thus getting continuous break conditions then the external pull down can be omitted.

The above configuration won't allow to reset the board from the external device (since using the P0_0 as a TX one should have the reset disabled), in that case either enable the reset again via a command from the external processor or set a GPIO pin to act as a POR source. An alternative solution would be to burn a secondary booter in the OTP and boot from different pins and keep the P0_0 as a reset.

On Dialog Pro-DK Motherboard this is solved by populating the resistor R30 resistor (1k) in the daughterboard as shown in Figure 26 and connecting the reset circuit of the motherboard as shown in Figure 27 with the R347 resistor (10k). This creates a strong pull-down. The figures below indicate part of the reset schematics.

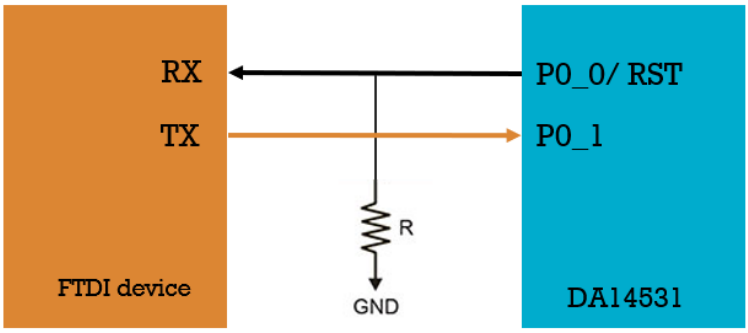


Figure 25: FTDI and DA14531 connection circuit for Reset

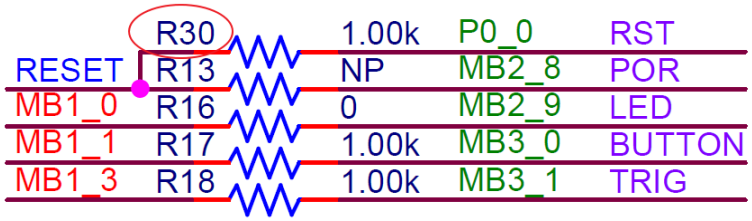


Figure 26: Reset line on the daughterboard with R30 resistor

© 2020 Dialog Semiconductor

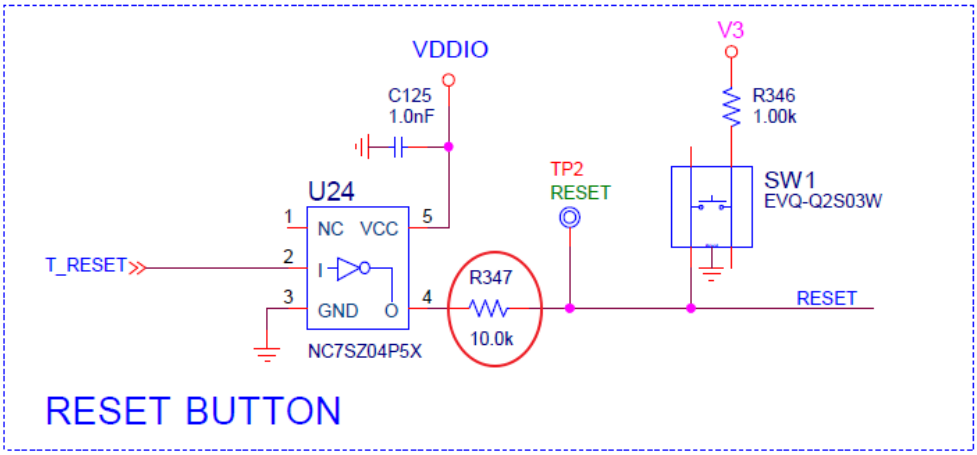


Figure 27: RESET circuit on the Pro-DK Motherboard with R347 resistor

12 List of supported FLASH/EEPROM memories

The memories listed below are currently supported by DIALOG. These memories support the standard Serial Peripheral Interface (SPI) and I2C and operate in the 3.3 V power supply.

Table 4: List of supported FLASH/EEPROM memories

Memory Type	Name	Memory Size	Page Size	Manufacturer Name
SPI	AT25XE011	128 kB	256 kB	Adesto
SPI	AT25XE021A	256 kB	256 kB	Adesto
SPI	AT25XE021A	512 kB	256 kB	Adesto
SPI	AT25XE512C	64 kB	256 kB	Adesto
SPI	AT25XE041D	512 kB	256 kB	Adesto
SPI	AT25DN011	128 kB	256 kB	Adesto
SPI	AT25DF512	64 kB	256 kB	Adesto
SPI	AT25XE512	64 kB	256 kB	Adesto
SPI	AT25DF011	128 kB	256 kB	Adesto

© 2020 Dialog Semiconductor

SPI	AT25DS011	128 kB	256 kB	Adesto
SPI	AT45DB041E	512 kB	256 kB	Adesto
SPI	P25X10U	128 kB	256 kB	Puya
SPI	P25Q10U	128 kB	256 kB	Puya
SPI	MX25R512F	64 kB	256 kB	Macronix
SPI	MX25R2035	256 kB	256 kB	Macronix
SPI	GD25VQ20C	256 kB	256 kB	GigaDevice
SPI	GD25WD10CTIG	128 kB	256 kB	GigaDevice
SPI	GD25WD40CTIG	512 kB	256 kB	GigaDevice
SPI	W25X10CL	128 kB	256 kB	Windbond
SPI	W25X20CL	256 kB	256 kB	Windbond
SPI	GD25WD20CTIG	256 kB	256 kB	GigaDevice
I2C	M24M01-R	128 kB	256 kB	STMicroelectronics

Note: This list is not extensive and other devices compliant to the requirement spec will work as well.

© 2020 Dialog Semiconductor

13 Conclusions

This document described the booting options available for the DA14531. Using this information, the user can select the best booting options for the end application.

© 2020 Dialog Semiconductor

14 Revision History

Revision	Date	Description
1.0	04-11-2019	Initial version
1.1	11-02-2020	Added section 10 Booting from custom PCB
1.2	12-03-2020	Added section 5 and updated few sections
1.3	10-05-2020	Added section 12 list of supported flash/eeprom memories

© 2020 Dialog Semiconductor

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, available on the company website (www.dialog-semiconductor.com) unless otherwise stated.

Dialog and the Dialog logo are trademarks of Dialog Semiconductor plc or its subsidiaries. All other product or service names are the property of their respective owners.

© 2020 Dialog Semiconductor. All rights reserved.

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 5769 5100

Taiwan
Dialog Semiconductor Taiwan
Phone: +886 281 786 222

Web site:
www.dialog-semiconductor.com

Hong Kong
Dialog Semiconductor Hong Kong
Phone: +852 2607 4271

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