User Manual

DA14580 Bluetooth® Smart Development Kit – Basic

UM-B-025

Abstract

This document describes the Bluetooth Smart Development Kit - Basic based on DA14580-01.
It helps users to set up the hardware development environment, install required software and quickly start product development with help of example source code.

Contents

Contents **2**

Tables **3**

Figures **3**

1 Terms and definitions **4**

2 References **4**

3 Introduction **5**

3.1 Hardware content 5

3.2 Web content 6

3.2.1 Software Development Kit content 6

3.3 DA14580DEVK –Basic Kit 8

3.3.1 PCB design and functionalities 8

3.3.2 Electrical Schematics 9

3.3.3 Configuring the basic kit-board by jumpersettings 11

3.3.4 Schematics and layout 12

4 Installation of tools and drivers **13**

4.1 Keil 13

4.2 SEGGER Jlink driver 14

4.3 Tera Term 14

5 Memory and tools **15**

6 Using the demo kit **16**

6.1 Run an example on DA14580 16

6.2 Run ‘Blinky’ and using the UART. 20

7 Power Management: measuring current **22**

Appendix A Layout **23**

Appendix B Connections of J10 and J11 ..... **23**

Appendix C Using the SmartSnippets CLI **24**

8 Web-Links **25**

9 Revision history **25**

Tables

Table 1: Content of the DA14580DEVKT –Basic Kit 5
Table 2: SDK Examples 7
Table 3: Installation tools and drivers 13
Table 4: Run an example on DA14580 16
Table 5: Way of Working (WoW) loading ‘Blinky’-code 20

Figures

Figure 1: DA14580DEVKT –Basic Kit 5
Figure 2: DA14580 Development Kit -Basic 8
Figure 3: Topview of PCB with components and functionalities 8
Figure 4: Electrical schematic of DA14580DEVKT-B sheet 1 of 2 9
Figure 5: Electrical schematic of DA14580DEVKT-B sheet 2 of 2 10
Figure 6: (Fabrication default) UART boot settings (T_TxD = P0_5 and T_RxD = P0_4) 11
Figure 7: Boot from external SPI memory 11
Figure 8: Layout of headers J4 and J6 12
Figure 9: Block diagram with different memory locations 15
Figure 10: Setup of DA14580DEVKT –BASIC during power measurement 22
Figure 11: Open the CLI of SmartSnippets 24

1 Terms and definitions

BLE	Bluetooth Low Energy
CS	Chip Select
DK	Development Kit
EEPROM	Electrically Erasable Programmable Memory
FTDI	Brand name of USB – UART interface
GPIO	General Purpose Input Output
OTP	One Time Programmable
PCB	printed circuit board
QFN	Quad-Flat No-leads
SDK	Software Development Kit
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SWD	Serial Wire Debug
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver/Transceiver
WLCSP	Wafer Level Chip Scale Packaging
WoW	Way of Working

2 References

1. DA14580, Datasheet, Dialog Semiconductor
2. UM-B-015, DA14580 Software Architecture, Dialog Semiconductor
3. DA14580 CB PXI QFN40 layout, Dialog Semiconductor
4. DA14580_CB_PXI_QFNP40, Dialog Semiconductor
5. DA14580_CB_PXI_WLCSP, Dialog Semiconductor
6. DA14580_CB_PXI_WLCSP_layout, Dialog Semiconductor
7. DA14580_MB_VB_layout, Dialog Semiconductor
8. DA14580 CB PXI_QFN48, Dialog Semiconductor
9. UM-B-005, DA14580 Peripheral Examples, Dialog Semiconductor
10. UM-B-010, DA14580 Proximity application, Dialog Semiconductor
11. AN-B-015 DA14580 Supply current measurement, Dialog Semiconductor

3 Introduction

DA14580 is a Bluetooth Smart chip, working with extremely low power while providing world-class RF performance, a small footprint and flexible peripheral configurations for a wide range of applications.

DA14580 development kit includes a set of hardware (e.g. development board with on-board debugger), a Software Development Kit (SDK) (e.g. development tools, source code examples documents and so on) along with documentation.

This document, as a user guide, helps users to set up hardware/software development environment, install required software and quickly start product development with the help of example source code.

Web content can be downloaded at www.dialog-semiconductor.com/support.

Product information about DA14580 can be found at:
www.dialog-semiconductor.com/products/bluetooth-smart/smartbond-da14580.

3.1 Hardware content

In Figure 1 the kit components are shown and in Table 1 the parts are printed.

Figure 1: DA14580DEVKT –Basic Kit

Table 1: Content of the DA14580DEVKT –Basic Kit

DA14580DEVKT –BASIC
Battery CR2032
Mini USB Cable

3.2 Web content

3.2.1 Software Development Kit content

3.2.1.1 Tools

Web-link: www.dialog-semiconductor.com/support

SmartSnippets (a framework of PC based tools to control DA14580 development kit), consisting of

- OTP Programmer: Tool for OTP memory programming
- UART booter: Tool for downloading hex files to DA14580 SRAM over UART
- SPI & I2C memory programmer: Tool for SPI flash and I2C EEPROM programming

Connection Manager (a PC based software tool to control the link layer of the DA14580), with the following capabilities:

- Functional in Peripheral and Central role
- Set advertising parameters
- Set connection parameters
- Reading from Attribute database
- Perform production test commands

3.2.1.2 SDK documents

- UM-B-003, DA14580 Software development guide
- UM-B-004, DA14580 Peripheral drivers
- UM-B-005, DA14580 Peripheral examples
- UM-B-006, DA14580 Sleep mode configuration
- UM-B-007, DA14580 Software Patching over the Air (SPOTA)
- UM-B-008, DA14580 Production test tool
- UM-B-010, DA14580 Proximity application
- UM-B-011, DA14580 Memory map – scatter file
- UM-B-012, DA14580 Secondary bootloader
- UM-B-013, DA14580 External Processor Interface over SPI
- UM-B-014, DA14580 Development Kit
- UM-B-015, DA14580 Software architecture

3.2.1.3 SDK source code examples (created with Keil IDE)

Web-link:

- **dk_apps.** This folder holds all the necessary folders needed for DA14580 application development.
  - **dk_apps\keil_projects\proximity**

    The folder contains the following subfolders and in each one of them resides the respective project file:

Table 2: SDK Examples

Folder	Project File	Description
monitor_fe	fe_proxm_sdk.uvproj	Proximity Monitor (External processor)
reporter_fe	fe_proxr.uvproj	Proximity Reporter (External processor)
reporter_fh	fh_proxr_sdk.uvproj	Proximity Reporter (Integrated processor)

  - **dk_apps\keil_projects\prod_test:** This folder include the source code of the production test firmware. Refer to UM-B-008_DA14580_Production_test_tool.pdf for more information how to build and use it.

- **host_apps:** This folder holds the DA14580 PC applications:
  - ***host_apps\windows\proximity:*** This folder includes two Windows C applications, with each one acting as part of a proximity monitor and a proximity reporter application. They are placed in subfolders *monitor* and *reporter* respectively. For details, please read the *DA14580 Proximity Application* Guide.
  - **host_binaries\windows\proximity:** This folder includes two pre-compiled Windows executables which correspond to the C applications described right above and are included for user convenience.
  - **peripheral_examples:** This folder includes sample code of how to use the peripheral blocks of the DA14580 (e.g. UART, SPI, I2C etc.) bundled to a demo-kit. For details, please refer to [9].

- Tools
  - **tools\prod_test\prod_test_cmds:** This folder includes the source code of the production test tool. Refer to UM-B-008_DA14580_Production_test_tool.pdf for more information how to build and use it.

Remark: in release **DA14580_SDK_3.0.2.1** the ‘define’ below has to be set, in order to configure the UART GPIOs for the Basic DK.

File : da14580_config.h

```
/*
* HW configuration
*/
#define HW_CONFIG_BASIC_DK  // Basic DK (DA14580DEVKT-B)
```

3.3 DA14580DEVK –Basic Kit

3.3.1 PCB design and functionalities

The top-screen layer of the Basic Kit PCB is shown in Figure 2.

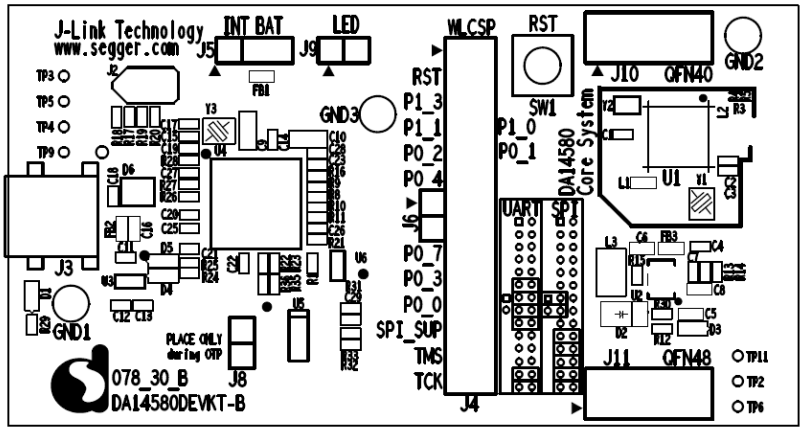


Figure 2: DA14580 Development Kit -Basic

In Figure 3. the different components and functionalities are shown.

A larger picture is displayed in Appendix A

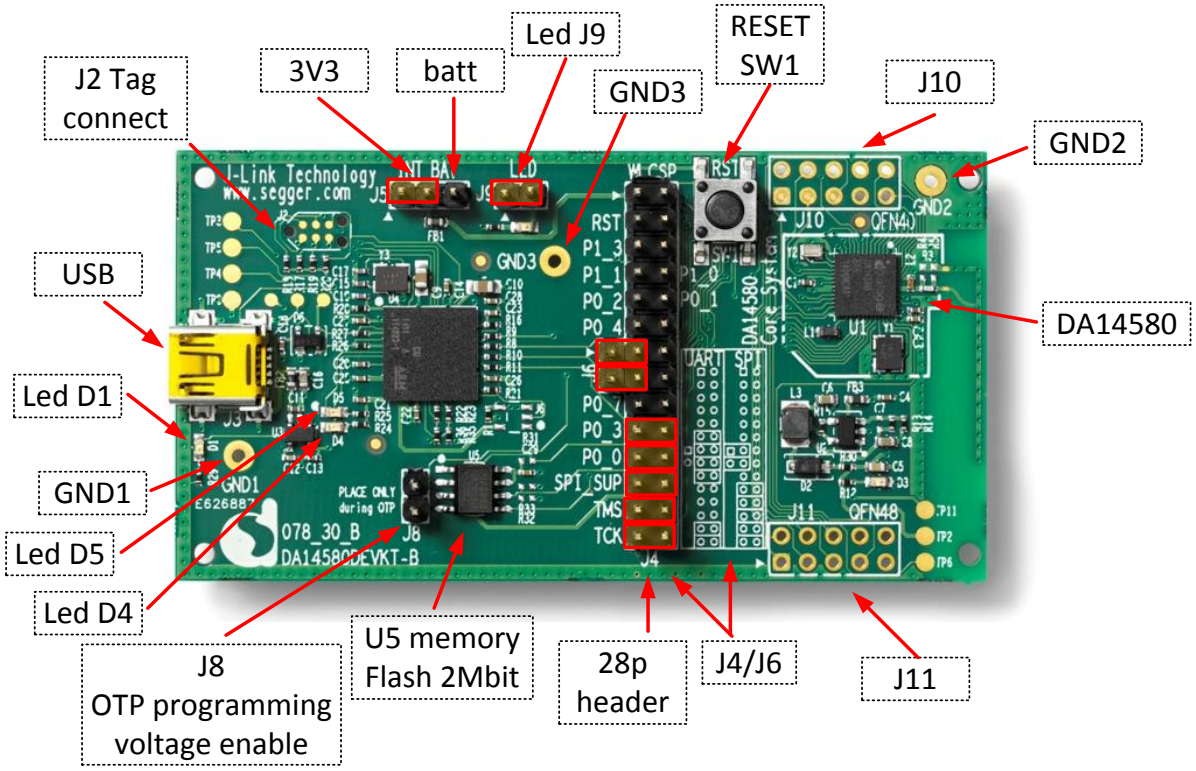


Figure 3: Topview of PCB with components and functionalities

3.3.2 Electrical Schematics

The schematics for the PCB are depicted in the following figures: Figure 4 and Figure 5.

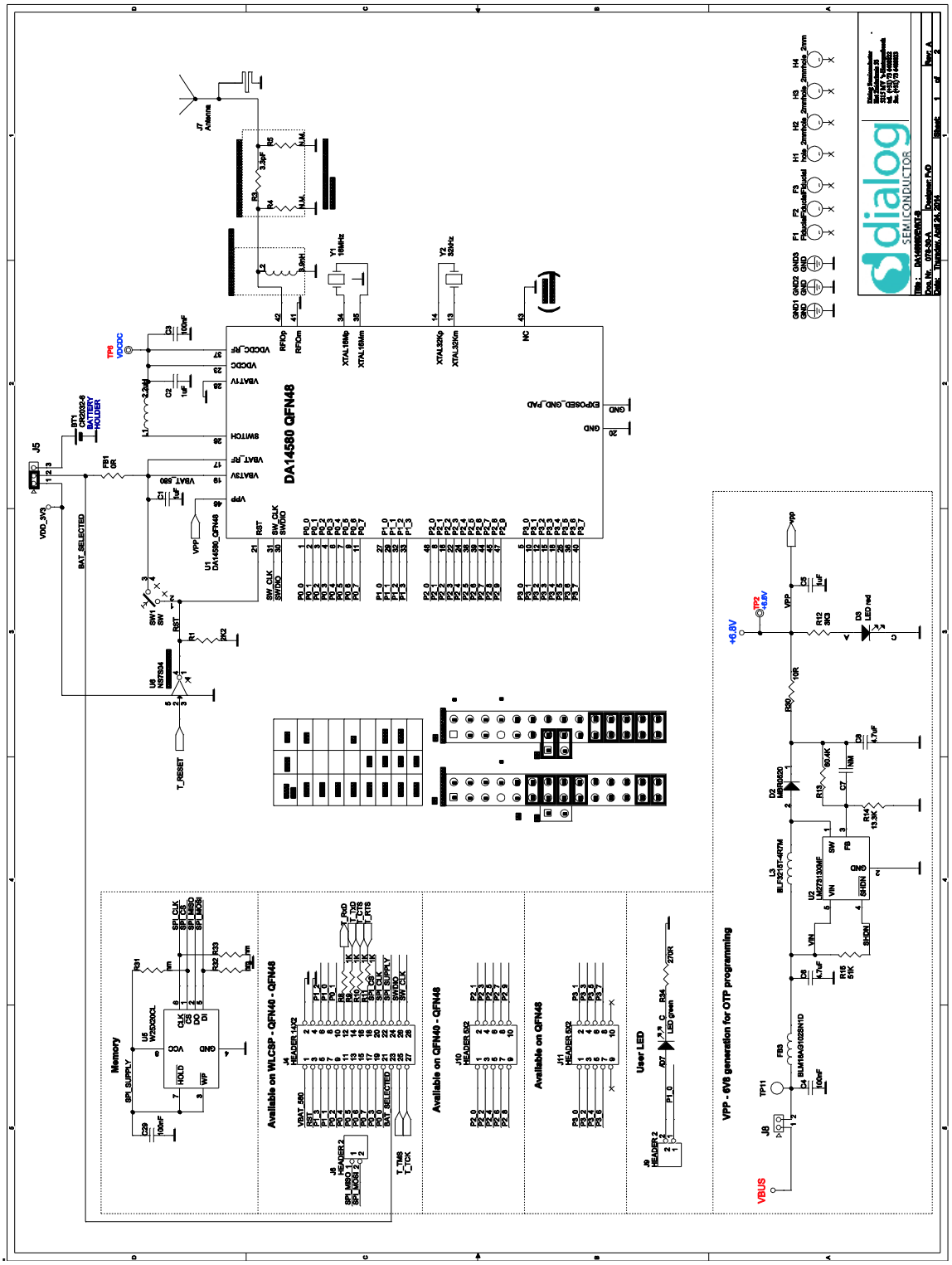


Figure 4: Electrical schematic of DA14580DEVKT-B sheet 1 of 2

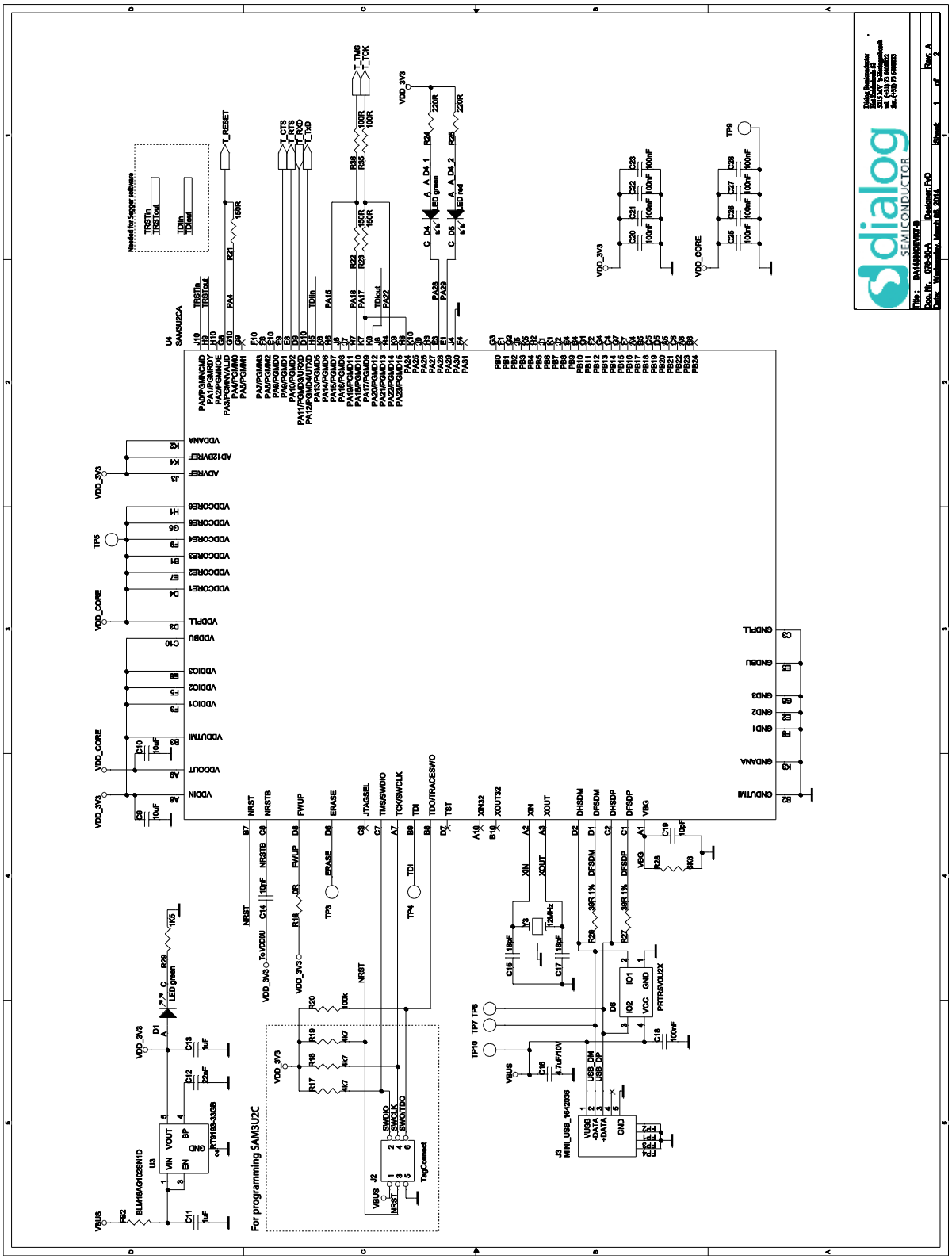


Figure 5: Electrical schematic of DA14580DEVKT-B sheet 2 of 2

3.3.3 Configuring the basic kit-board by jumpersettings

Different functionalities are shown in Appendix A

The jumper settings are displayed below.

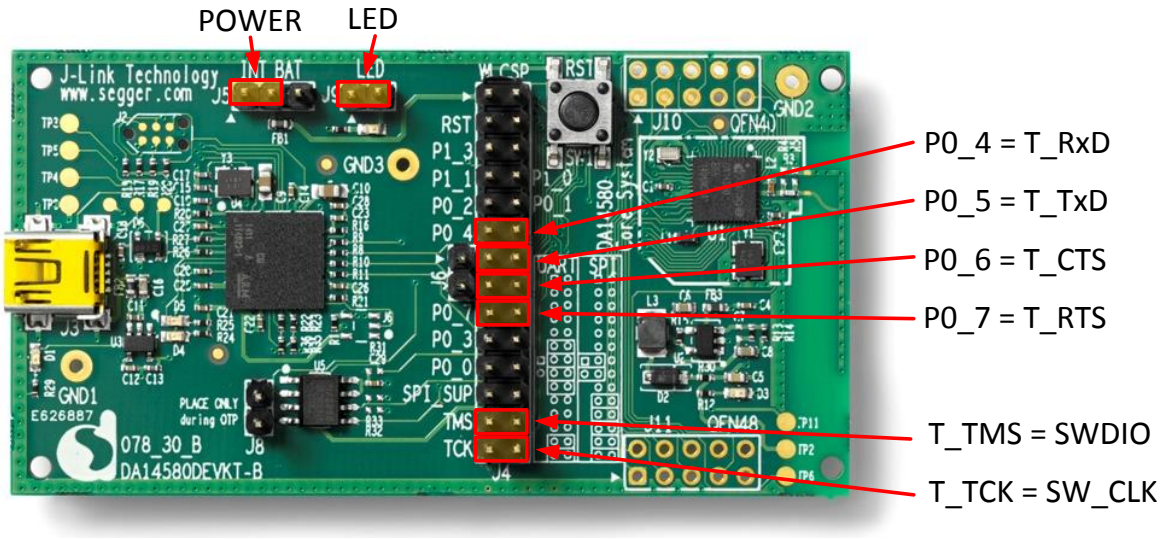


Figure 6: (Fabrication default) UART boot settings (T_TxD = P0_5 and T_RxD = P0_4)

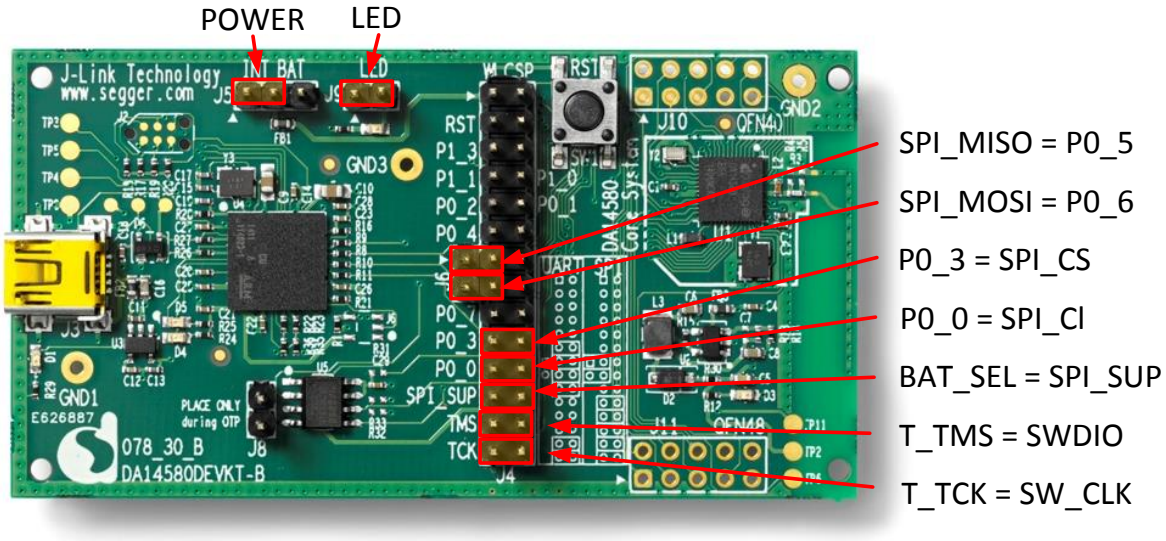


Figure 7: Boot from external SPI memory

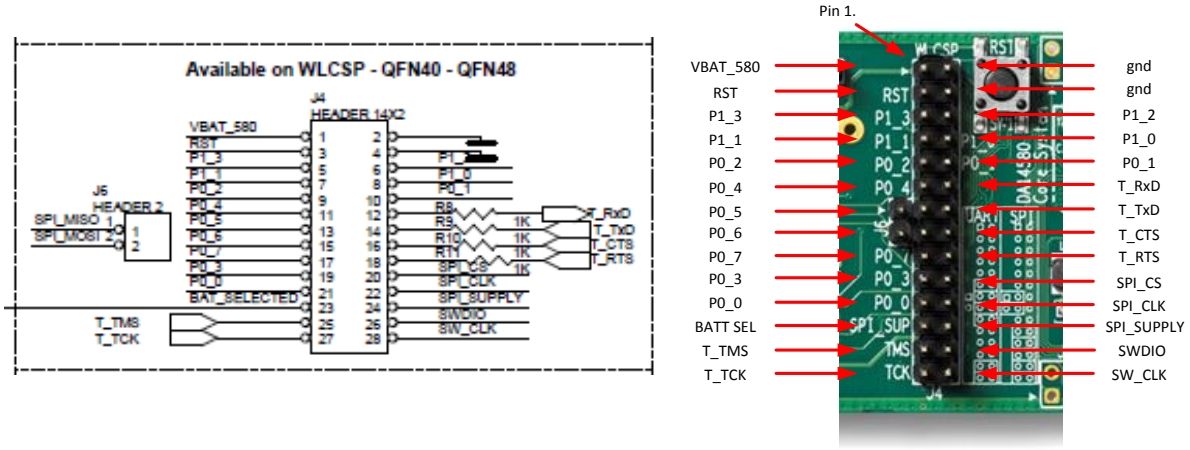


Figure 8: Layout of headers J4 and J6

Example: when jumper J4 (27-28) is placed, connection T_CK = SW_CLK is made.

In Figure 6 and Figure 7 the connections are added next to the arrows.

The board is equipped with a QFN48 package.

On this board only the Buck mode is used. A choice can be made between 3V3 (via USB)(J5 1-2) or Vdd (via a coin cell)(J5 2-3). The battery (coin cell) is placed on the back of the PCB. No battery is needed when running via the USB-mini-cable.

3.3.4 Schematics and layout

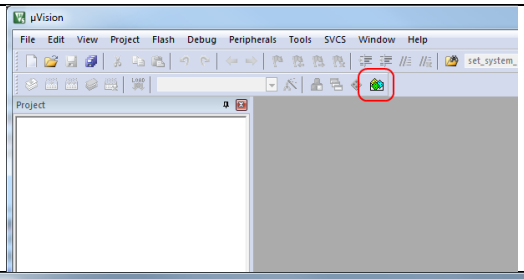
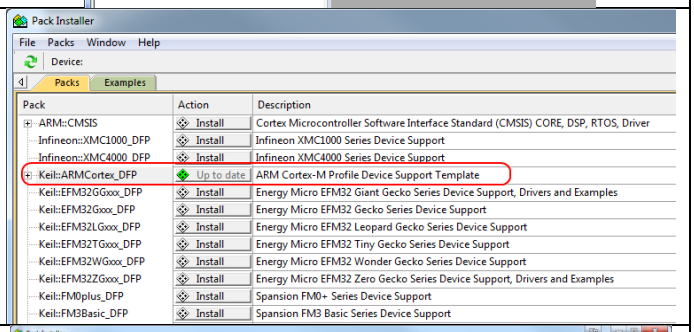
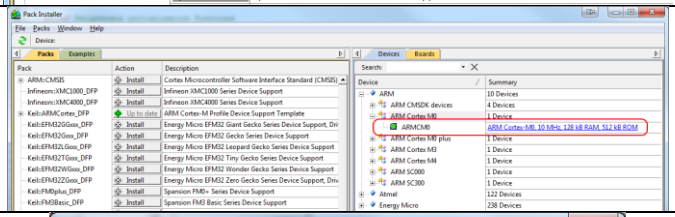
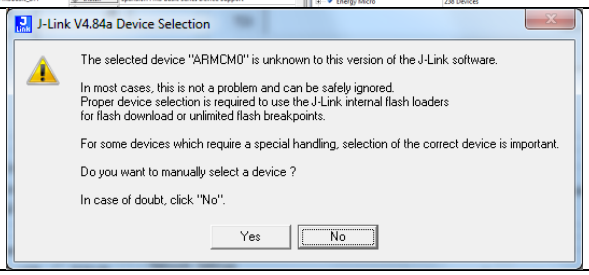
For the schematics and layout of the board please refer to the respective documents on the portal.

See link: www.dialog-semiconductor.com/support.

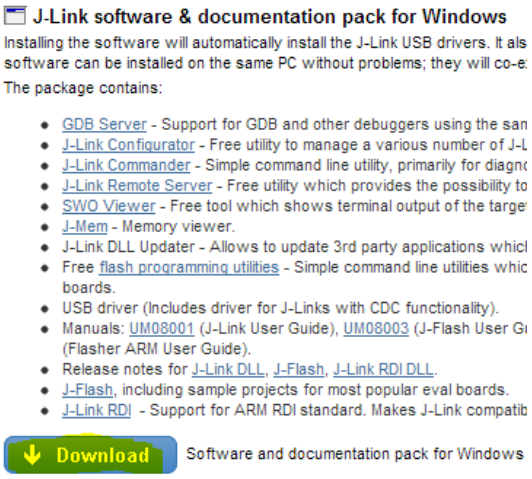
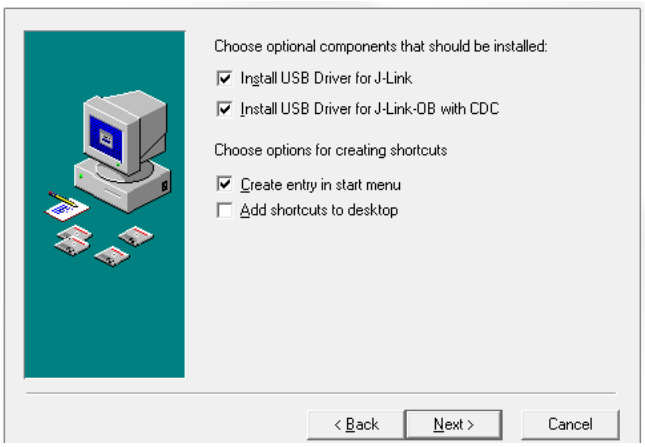
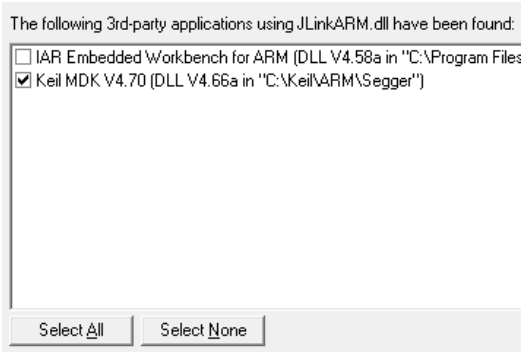
4 Installation of tools and drivers

To install the Software development environment, please follow the steps as shown in Table 3.

Table 3: Installation tools and drivers

4.1 Keil		
4.1.1	Download and install Keil MDK-ARM uVision IDE. Both uVision ver. 4.0 and ver. 5.0 are supported.	Keil: https://www.keil.com/download/product/ Keil MDK-ARM Version 5 – Installation: http://www2.keil.com/mdk5/install Starters Guide: http://www.keil.com/uvision/ide_ov_starting.asp
4.1.2	For uVision Version 5 you have to install the ARM Cortex M profile package (see also http://www.keil.com/dd2/Pack/)	
4.1.3	You should see a list of packs as shown on the right. If you do not see this list, please click the “Packs” menu item and select the “Check for Updates” option to download an updated list. Click on the “Install” button to the right of “Keil::ARMCortex_DFP” package.	
4.1.4	If the installation is successful, the pack installer window should look like this.	
4.1.5	This screen may occur. Select ‘yes’. Select ‘M0’.	

4.2 SEGGER Jlink driver

4.2.1	Download and install the Jlink software & documentation pack for Windows. *Please note that your SEGGER Jlink serial number is required for downloading.* *Use sticker-number on backside of board.*  .	 http://www.segger.com/jlink-software.html
4.2.2	In order to have the USB controller properly recognized by Windows as a J-Link device, you have to install the driver with the settings shown in the side figure.	
4.2.3	At the end of the installation, please tick the IDE (Keil MDK Vxx) that you are using.	

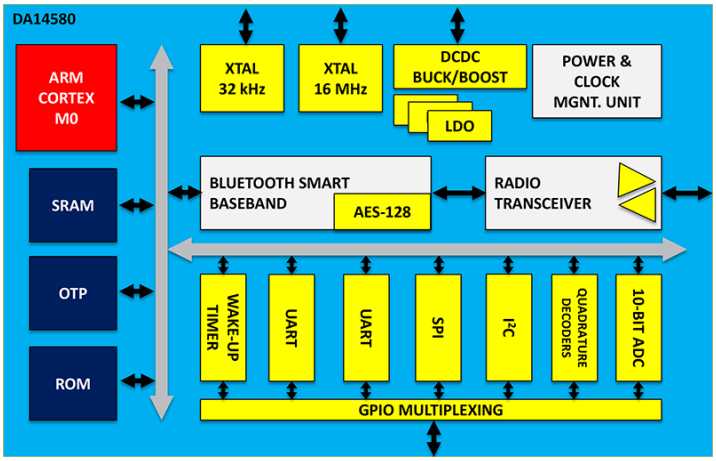
4.3 Tera Term

4.3.1	Download and install Tera Term on your PC.	Tera Term: http://en.sourceforge.jp/projects/ttssh2/releases/

5 Memory and tools

The DA14580DEVKT –Basic is equipped with, on the chip, SRAM (42k) and OTP (32k). Mounted on the board is external SPI flash memory (2Mbit).

Figure 9: Block diagram with different memory locations



Software can be downloaded to:

- SRAM
  - Keil IDE
  - SmartSnippets
  - Command Line Interface (CLI)
  - Connection Manager
- OTP
  - SmartSnippets
  - CLI
- SPI (flash)
  - SmartSnippets
  - CLI

Example: loading software (hex-file) by using SmartSnippets

- PC → UART → DA14580
- PC → UART → DA14580 → SPI (flash)
- PC → UART → DA14580 → OTP

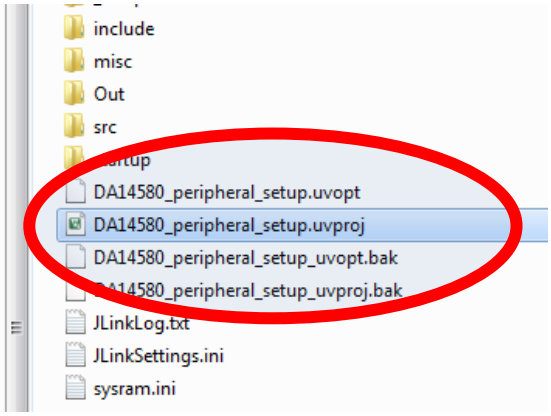
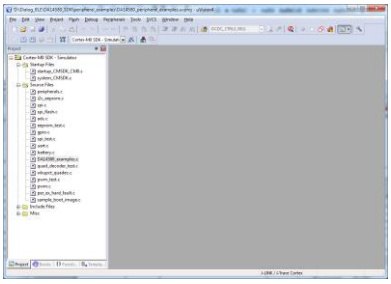
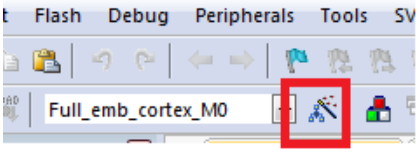
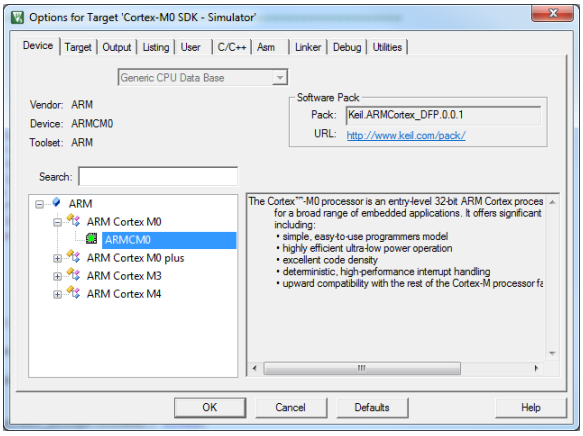
For the settings of the jumpers see Figure 6.

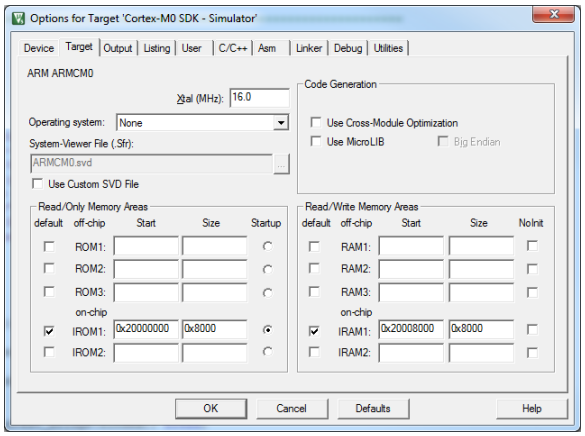
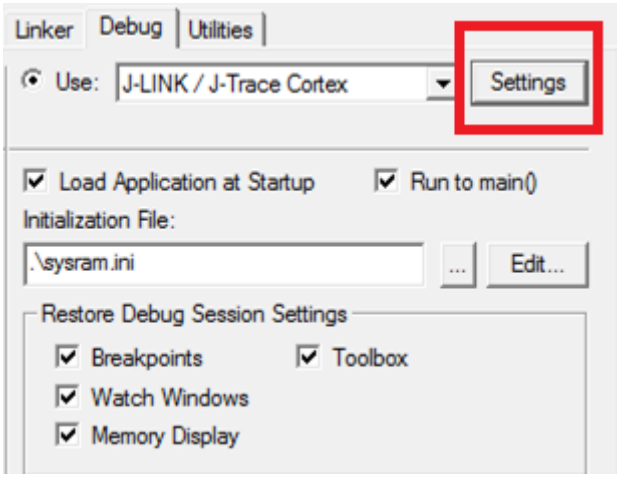
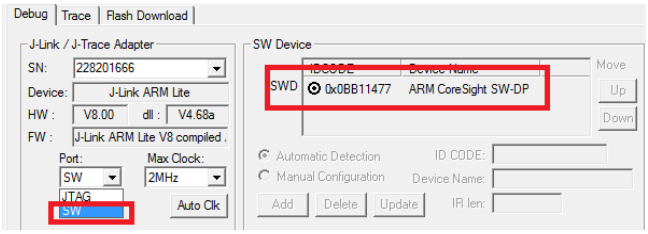
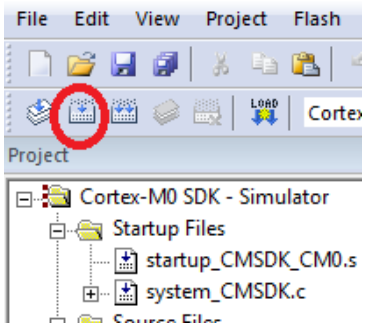
An example of the **CLI** is shown in Appendix C

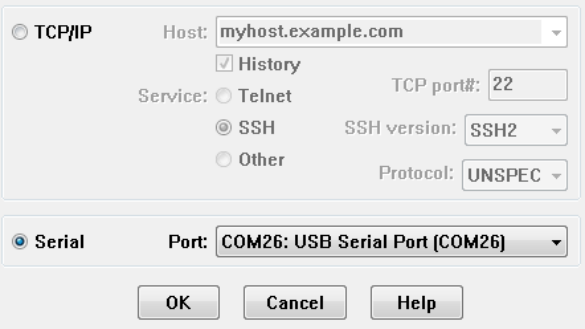
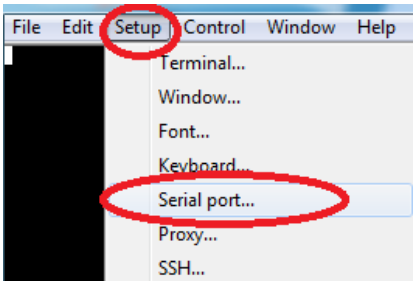
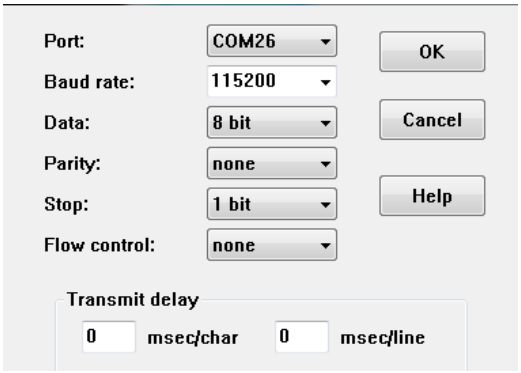
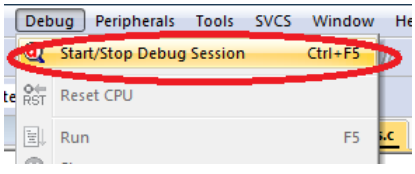
6 Using the demo kit

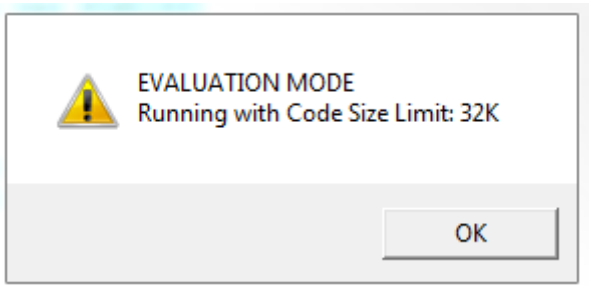
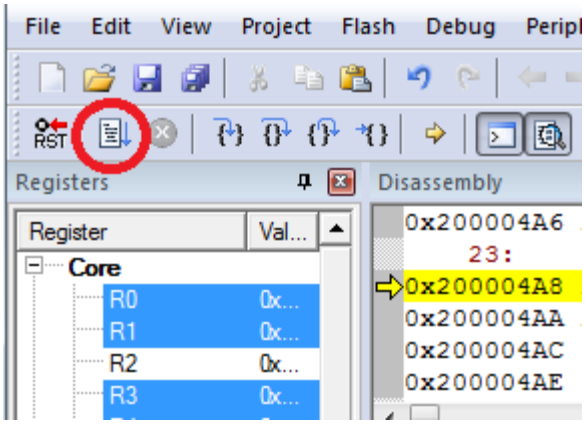
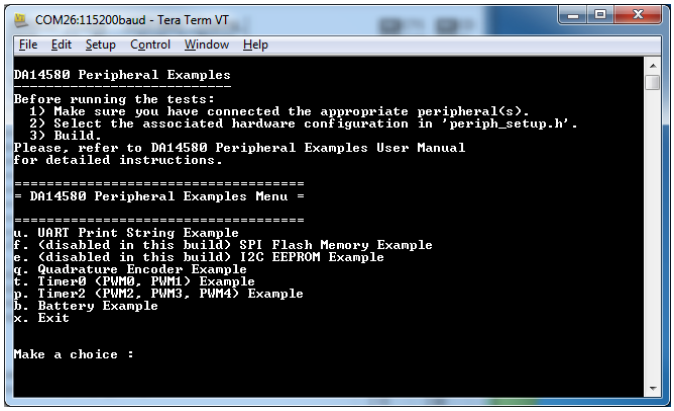
Follow these steps shown in Table 4 to easily create a working demo kit.

Table 4: Run an example on DA14580

6.1 Run an example on DA14580		
6.1.1	After you download the SDK at www.dialog-semiconductor.com/support you can find a source code example in example directory called "*peripheral_examples\peripheral_setup*". Double click "580_peripheral_setup.**uvproj**", as shown in the image to the right. Mind remark 3.2.1.3!	
6.1.2	The development environment should look like this when the project is opened with Keil.	
6.1.3	Click on the '**Options for Target**' button	
6.1.4	'Options for Target' → 'Device' -screen should look like this.	

6.1.5	‘Options for Target’ → ‘Target’ -screen looks like this. Mind the ‘Memory’ areas.	
6.1.6	Make sure “J-LINK/J-Trace Cortex” is selected as shown and the initialization file field is set correctly to “.\sysram.ini”.	
6.1.7	Click on the “Setting” button above to make sure the SW Device has been detected correctly.	
6.1.8	Click “OK” to save the settings.	All settings have been saved properly now, and you can continue to build the example.
6.1.9	Build the project by pressing “F7” key, or click the build button as shown in following picture	

6.1.10	Make sure you have a UART connection between your PC and a mother board, as shown in 6.1.5. Check the “COM” number on you PC.	Go to the Windows Control Panel →Administrative Tools →Computer Management → Device Manager → Ports → USB Serial Port # (connect or disconnect to see the COM port of that module)
6.1.11	Open the Tera Term serial terminal on you PC.	
6.1.12	Open Tera Term and choose a COM port, which you have found in step 3, and click OK	
6.1.13	Choose Setup->Serial port to configure the Baud rate etc.	
6.1.14	Set “Baud rate” to 115200, ”Data” to 8 bit, ”Parity” to None, ”Stop” to 1 bit and “Flow control” to none. Click OK. Now we have a properly configured UART terminal on our PC.	
6.1.15	Go back to Keil Project. In the menu bar, select Debug->Start/Stop Debug Session.	

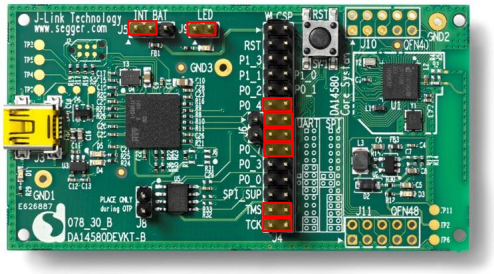
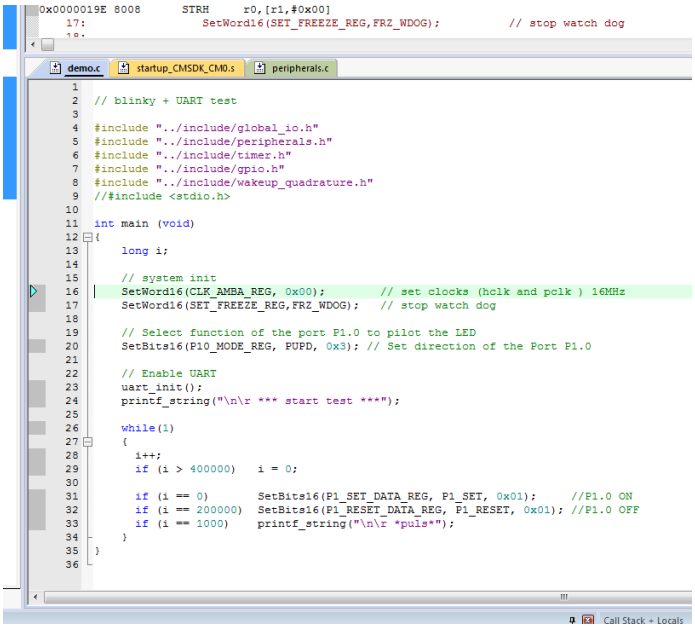
6.1.16	A dialog window pops up, like the one on the right. Please click “OK”.	
6.1.17	Press F5 key or click execution button as shown in following picture, to start code execution.	
6.1.18	Then you can see a *hello* message on your UART terminal screen. That means you have successfully programmed and started the peripheral program on DA14580 Demo board. The *peripheral_setup* demo consists of a small suite of tests that encompasses some of the most commonly used peripherals such as I2C EEPROM, SPI Flash, Rotary Encoder, audio buzzer etc. For more detailed info and technical details please refer to the *UM-B-005: DA14580 Peripheral Examples* as well as the source code of the *peripheral_setup* demo. Remark: this software is originally written for the ‘SmartBond™ DA14580 -- Development Kit – Expert’. See: http://support.dialog-semiconductor.com/resources	

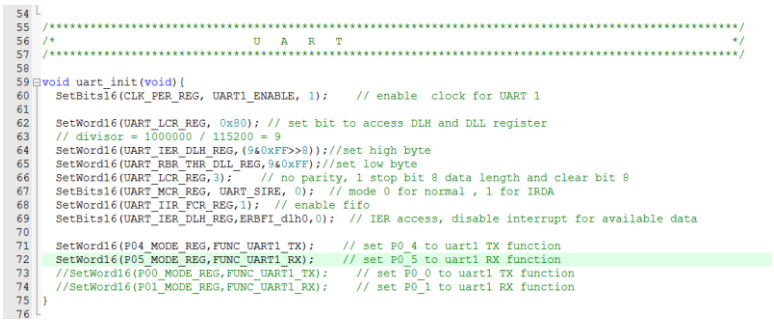
6.2 Run ‘Blinky’ and using the UART.

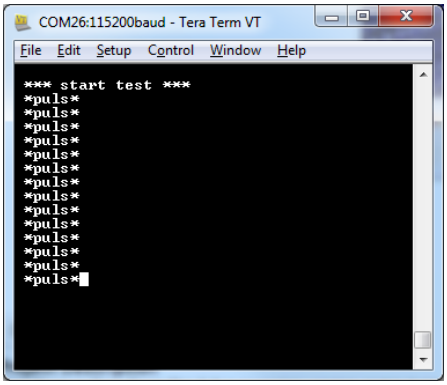
Way of working: the Keil IDE is used to load software into the DA14580. As shown in 6.1 software is downloaded to SRAM through SWD (Serial Wire Debug). After downloading the software, it can be debugged via the same IDE.

For software: see added ‘Blinky_UART_code’ zip-file.

Table 5: Way of Working (WoW) loading ‘Blinky’-code

<table>
<tr><th colspan="3">Preparations</th></tr>
<tr><td colspan="3">This chapter shows the user how to quickly set up the software development environment of the BLE (Bluetooth Low Energy) demo</td></tr>
<tr><td>6.2.1</td><td>Attach mini USB cable to PC and demo board.
Functionality of this cable:
● power
● programming
● debugging
● UART</td><td>  </td></tr>
<tr><td>6.2.2</td><td>Add 8 jumpers, as shown in Figure 6.</td><td>Jumpers: J5 (1-2), J9, J4 TMS (25-26), J4 TCK (27-28), J4 (11-12), J4 (13-14), J4 (15-16) and J4 (17-18).</td></tr>
<tr><td>6.2.3</td><td>Start Keil IDE software and load ‘Blinky’ example code. Via this Blinky-code the LED on the board starts blinking. Also added in this code, is the UART sending a text string.</td><td> 
</td></tr>
</table>

6.2.4	In function 'uart_init(void)' default **P0_4** Tx and **P0.5** Rx are used as IO-ports.
	See lines 71 and 72.
		
		54
		55 /***/
		56 /* U A R T */
		57 /***/
		58
		59 void uart_init(void){
		60 SetBits16(CLK_PER_REG, UART1_ENABLE, 1); // enable clock for UART 1
		61
		62 SetWord16(UART_LCR_REG, 0x80); // set bit to access DLH and DLL register
		63 // divisor = 1000000 / 115200 = 9
		64 SetWord16(UART_IER_DLH_REG,(9&0xFF>>8));//set high byte
		65 SetWord16(UART_RBR_THR_DLL_REG,9&0xFF);//set low byte
		66 SetWord16(UART_LCR_REG,3); // no parity, 1 stop bit 8 data length and clear bit 8
		67 SetBits16(UART_MCR_REG, UART_SIRE, 0); // mode 0 for normal , 1 for IRDA
		68 SetWord16(UART_IIR_FCR_REG,1); // enable fifo
		69 SetBits16(UART_IER_DLH_REG,ERBFI_dlh0,0); // IER access, disable interrupt for available data
		70
		71 SetWord16(P04_MODE_REG,FUNC_UART1_TX); // set P0_4 to uart1 TX function
		72 SetWord16(P05_MODE_REG,FUNC_UART1_RX); // set P0_5 to uart1 RX function
		73 //SetWord16(P00_MODE_REG,FUNC_UART1_TX); // set P0_0 to uart1 TX function
		74 //SetWord16(P01_MODE_REG,FUNC_UART1_RX); // set P0_1 to uart1 RX function
		75 }
		76

6.2.5	Compile and Debug the software. The blinking led D7 is visible and the TeraTerm screen is showing 'popping up' lines. Choose the right com-port via the Device Manager of your PC.	 COM26:115200baud - Tera Term VT File Edit Setup Control Window Help *** start test *** *puls* *puls* *puls* *puls* *puls* *puls* *puls* *puls* *puls* *puls* *puls* *puls* *puls*

7 Power Management: measuring current

The design of this DA14580DEVKT –Basic is made in such a way that the microcontroller can be isolated completely from the rest of the board.

This is illustrated in the block diagram shown in Figure 10. Shown are the connections of the jumpers J4, J5 and J6. For extra info see the electrical schematic on Figure 4.

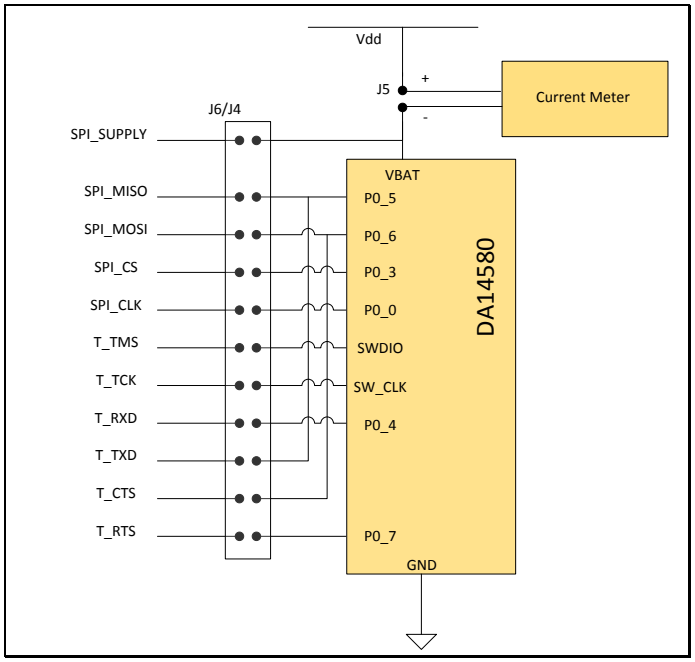


Figure 10: Setup of DA14580DEVKT –BASIC during power measurement

Steps how to do the power measurements:

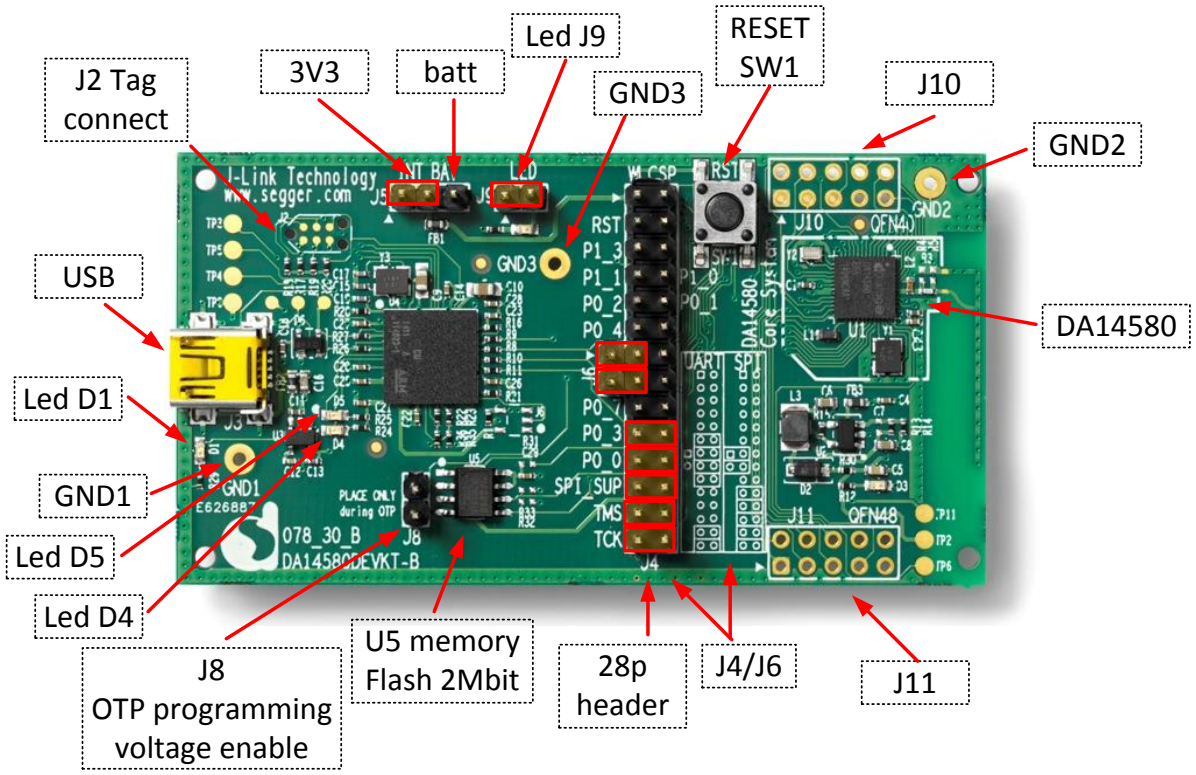
1. Connect the Current Meter to jumper J5.
2. Mount the jumpers needed for downloading the software (see chapter 3.3.3 and chapter 5).
3. Download the software.
4. Start the software.
5. Wait till software has reached ‘Deep Sleep’.
6. Dismount **all** the jumpers.
   Now almost all the DA14580 pins are isolated and only the current meter and GND are connected.
7. Read the current.

For additional info see: AN B – 015 DA14580 Supply current measurement. [12]

See chapter 4.4 Deep Sleep current measurement.

Web-link: http://support.dialog-semiconductor.com/system/files/AN-B-015_DA14580_Current_Measurement.pdf

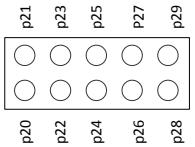
Appendix A Layout



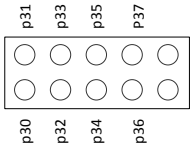
Appendix B Connections of J10 and J11

QFN48
QFN40

J10



QFN48 **J11**



Appendix C Using the SmartSnippets CLI

All the information/syntaxes about the CLI can be found from the **HELP** tab in the SmartSnippets GUI or by written **Smartsnippets –help** in the CLI.
In this example, it is supposed that the SPI memory is using P0_0 as SCK, P0_3 as CS, P0_5 as MISO and P0_6 as MOSI.

First of all, the CLI can send the commands either via UART or JTAG according to the binary file which has to be loaded.

On the one hand, if the commands are going to be sent via UART, the following binary file which can be found from the resources folder of SmartSnippets has to be downloaded into the DA14580:

- *flash_programmer.bin*

On the other hand, if the commands are going to be sent via JTAG, the following binary file which can be found from the resources folder of SmartSnippets has to be downloaded into the DA14580:

- *jtag_programmer.bin*

Secondly, open the CLI by pushing the Shift button and right click on the **'bin'** folder of the SmartSnippet and select '**Open command window here'** as follow:

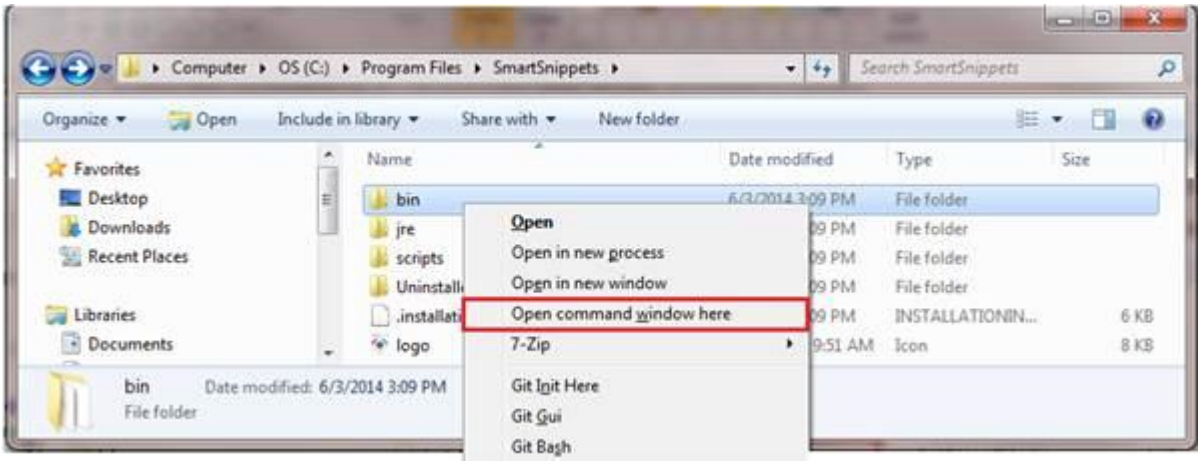


Figure 11: Open the CLI of SmartSnippets

Finally, in order to write a value 0x1347 (example of a bluetooth device address) at the address 0x93 for instance, the following command line can be written:

SmartSnippets.exe -type spi -chip DA14580-01 -jtag 228202458 -cmd write_field -offset 0x93 -data 1347 -firmware "D:\SmartSnippets\resources\jtag_programmer.bin"

The answers should be as follow:

```
Found SWD-DP with ID 0x0BB11477
FPUnit: 4 code (BP) slots and 0 literal slots
Found Cortex-M0 r0p0, Little endian.
BTLE device DA14580 selected.
Using default GPIO pin Id: P1_2.
File c:\users\glagnieu\smartsnippets\ressources\jtag_programmer.bin could not been found.
Using default baudrate: 57600 Bd.
Burned 2 bytes to address 0x00093.
```

8 Web-Links

- Support Dialog Semiconductors e.g. datasheets and software:
  http://support.dialog-semiconductor.com/resources#tools
- SmartBond DA14580:
  http://www.dialog-semiconductor.com/products/bluetooth-smart/smartbond-da14580
- datasheet DA14580:
  http://support.dialog-semiconductor.com
- SmartBond Reference Designs
  http://www.dialog-semiconductor.com/products/bluetooth-smart/smartbond-reference-designs

9 Revision history

Revision	Date	Description
1.0	10-Jun-2014	Initial version for DA14580-01.

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 European Directive 2001/95/EC and from 2 January 2013 onwards to European Directive 2011/65/EU concerning Restriction of Hazardous Substances (RoHS/RoHS2).
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 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 849929

China
Dialog Semiconductor China
Phone: +86 21 5178 2561

Korea
Dialog Semiconductor Korea
Phone: +82 2 3469 8291