User manual

DA1458x Getting started with Development Kit – Basic

UM-B-048

Abstract

This document describes the Bluetooth Smart Basic Development Kit based on DA14580/581/583/585/586. It helps users to set up the hardware development environment, install required software and quickly start product development with help of example source code on SDK v5.x and SDK v6.

DA1458x Getting started with Development Kit – Basic

Contents

Figures

Tables

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

2 References

1. DA14580, Datasheet, Dialog Semiconductor
2. DA14581, Datasheet, Dialog Semiconductor
3. DA14583, Datasheet, Dialog Semiconductor
4. DA14580_CB PXI QFN40 layout, Dialog Semiconductor
5. DA14580_CB_PXI_QFNP40, Dialog Semiconductor
6. DA14580_CB_PXI_WLCSP, Dialog Semiconductor
7. DA14580_CB_PXI_WLCSP_layout, Dialog Semiconductor
8. DA14580_MB_VB_layout, Dialog Semiconductor
9. DA14580_CB PXI_QFN48, Dialog Semiconductor AN-B-015, DA14580 Supply current measurement, Dialog Semiconductor
10. UM-B-012, DA14580 Creation of a secondary boot loader, User manual, Dialog Semiconductor
11. UM-B-051, DA1458x Software Platform Reference
12. AN-B-015 DA14580/581 Supply current measurements.

3 Introduction

DA14580/581/583/585/586 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. The 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 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.

3.1 Hardware content

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

Figure 1: DEVKT – Basic Kit

Table 1: Content of the DEVKT – Basic Kit

DA14580/581/583/585/586 basic board
Battery: CR2032
USB → Mini USB Cable

Note 1 Kits are not pre-programmed so must be programmed before use!

3.2 Basic Development Kit for DA1458x family

Dialog semiconductor DA1458x BLE SoC family consists of DA14580, DA14581, DA14583, DA14585 and DA14586.

3.2.1 Differences between DA14580, DA14581, DA14583, DA14585 and DA14586

Hardware difference between basic development boards, is the design in of QFN48 (580) and the QFN40 (580, 581, 583, 585 and 586). The silkscreen may have small textual differences.

Table 2: DA14580

Product	Memory size	General Purpose I\O's	Package	Key Features	Applications
WL-CSP34	ROM 84kBytes OTP 32kBytes RAM 50kBytes	12	2.5x2.5x0.5mm, pitch 0.4mm	Bluetooth 4.0 + 4.1 Cortex M0 application processor Power supply 0.9 - 3.3V Single pin RF I/O Rich set of analog and digital peripherals	Beacon & Proximity Health & Fitness HID Smart Home
QFN40		24	5x5x0.9mm, pitch 0.4mm		
QFN48		32	6x6x0.9mm, pitch 0.4mm		

The DA14581 uses a dedicated ROM which offers optimizations targeting A4WP and HCI

Table 3: DA14581

Product	Memory size	General Purpose I\O's	Package	Key Features	Applications
WL-CSP34	ROM 84kBytes OTP 32kBytes RAM 50kBytes	12	2.5x2.5x0.5mm, pitch 0.4mm	Bluetooth 4.0 + 4.1 Cortex M0 application processor Power supply 0.9 - 3.3V Single pin RF I/O Rich set of analog and digital peripherals 8 connections Optimized boot time	Wireless charging (A4WP) HCI
QFN40		24	5x5x0.9mm, pitch 0.4mm		

Table 4: DA14583

Product	Memory size	General Purpose I\O's	Package	Key Features	Applications
QFN40	**Flash** 1 Mbit (128kBytes) ROM 84kBytes OTP 32kBytes RAM 50kBytes	24	5x5x0.9mm, pitch 0.4mm	Bluetooth 4.0 + 4.1 Cortex M0 application processor Power supply 2.35 - 3.3V Single pin RF I/O Rich set of analog and digital peripherals	Beacon & Proximity Health & Fitness HID Smart Home

- A4WP wireless charging features:
  - Fast boot time for Power Receiving Unit (PRU)
  - 8 connections for Power Transmitting Unit (PTU)

- HCI features:
  - Optimized code for HCI which fits into the OTP
  - This enables customers/modules makers to provide a pre-programmed HCI module

Remark: DA14583 can run in BUCK mode only!

The DA14583 is a DA14580 plus SPI Flash Memory of 1Mbit in the same package.

Table 5: DA14585

Product	Memory size	General Purpose I\O's	Package	Key Features	Applications
WL-CSP34	ROM 128kBytes OTP 64 kBytes RAM 96 kBytes	14	2.40 mm x 2.66 mm, pitch 0.4mm	Complies with Bluetooth V5.0, Cortex M0 application processor Power supply 0.9 - 3.3V with 1.8V cold boot support Single pin RF I/O Rich set of analog and digital peripherals 8 connections Optimized boot time	Voice-controlled remote controls Beacons (Multi-sensor) Wearable devices: - Fitness trackers - Consumer health Smartwatches Human interface devices: - Keyboard - Mouse Toys, Consumer appliances
QFN40		25	5x5x0.9mm, pitch 0.4mm		

Table 6: DA14586

Product	Memory size	General Purpose I\O's	Package	Key Features	Applications
QFN40	Flash 2Mbits (256kBytes) ROM 128kBytes OTP 64 kBytes RAM 96 kBytes	24	5x5x0.9mm, pitch 0.4mm	Complies with Bluetooth V5.0, Cortex M0 application processor Power supply 0.9 - 3.3V with 1.8V cold boot support Single pin RF I/O Rich set of analog and digital peripherals 8 connections Optimized boot time	Voice-controlled remote controls Beacons (Multi-sensor) Wearable devices: - Fitness trackers - Consumer health Smartwatches Human interface devices: - Keyboard - Mouse Toys, Consumer appliances

The DA14586 is a DA14585 plus SPI Flash Memory of 2Mbit in the same package.

3.3 Software Development Tools

3.3.1 SmartSnippets Studio Introduction

Figure 2: SmartSnippets Studio platform

Dialog SmartSnippets Studio™ is a royalty-free software development platform for Smartbond™ devices. It fully supports the DA1468x and DA1458x family of devices.
SmartSnippets Studio™ contains:

- SmartSnippets™ Toolbox: A tool suite covering all software developer needs, including:
  - Power profiling
  - Programming and loading of firmware into SRAM, OTP and Flash
- SmartSnippets™ IDE: Eclipse CDT based IDE pre-configured plugins allowing easy out of the box set-up of build/debug environment
- SmartSnippets™ DA1458x SDK
- SmartSnippets™ Documentation
- RF master is a tool supporting the Bluetooth SIG defined Direct Test Mode for RF PHY testing of Bluetooth low energy devices. In addition it supports a number of vendor specific HCI commands for device testing. User can access RF Master by selecting RF Master under the Layout tab of the ribbon menu, which loads RF Master with Log. Alternatively, user can select RF Master tool under Tools tab of the ribbon.

The SmartSnippets™ IDE is supported by an on-board debugger from SEGGER. This offers standard debugging capabilities such as single stepping, setting breakpoints, software download and many more. For more details on the debugger capabilities, visit https://www.segger.com/.

3.3.2 Installation

The description of the needed steps is presented:

- Download the SmartSnippets tools from the Dialog Support Website: https://support.dialog-semiconductor.com (registration required).
- Unzip the zip.
- Run SmartSnippets Studio installer (.msi).
  - Install the recommended version of SEGGER J-Link GDB server.
  - Select the destination folder for the SmartSnippets Studio.
- Run the application.

3.3.3 Starting SmartSnippets Studio

When SmartSnippets Studio starts for the first time, the user must configure it. The necessary configurations are the following:

- Select the workspace folder for SmartSnippets. The dialog is shown in Figure 3. It is suggested to choose the root directory of the SmartSnippets DA1458x SDK. If this is not done correctly, then the message “you have not selected the latest SDK” will be shown. The correct message should be: “You have not yet selected a SDK. Click the Browse button above and select the SDK location”.

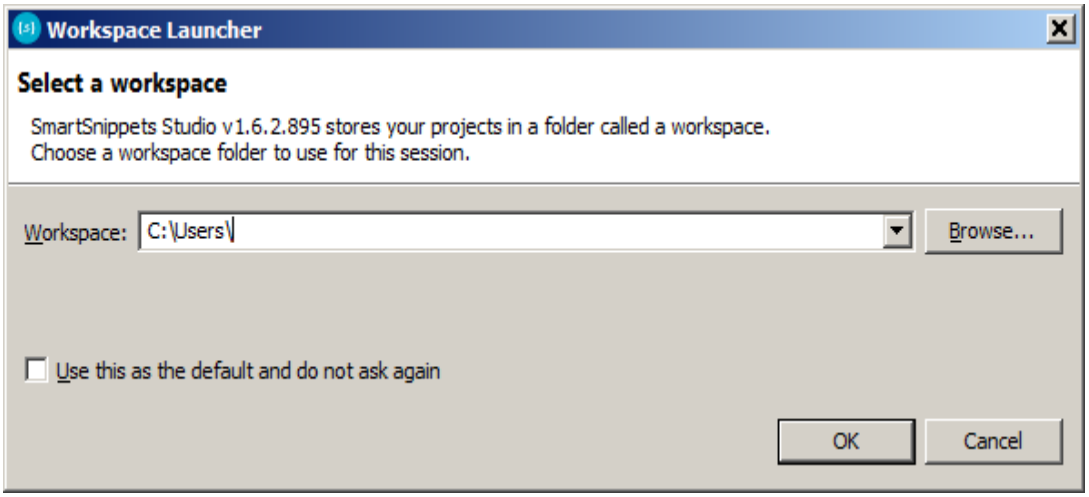


Figure 3: Dialog for Selecting Workspace

If necessary, specify how the selected workspace should be treated. (DA1468x 1.0.8 SDK or DA1458x 5.0.4 SDK)

- A set of tools required for all SDKs are getting automatically detected, such as GNU ARM GCC, SmartSnippets Toolbox, etc. If required version of software cannot be found, they will be downloaded and installed by the SDK Tools Installer (See Figure 4).

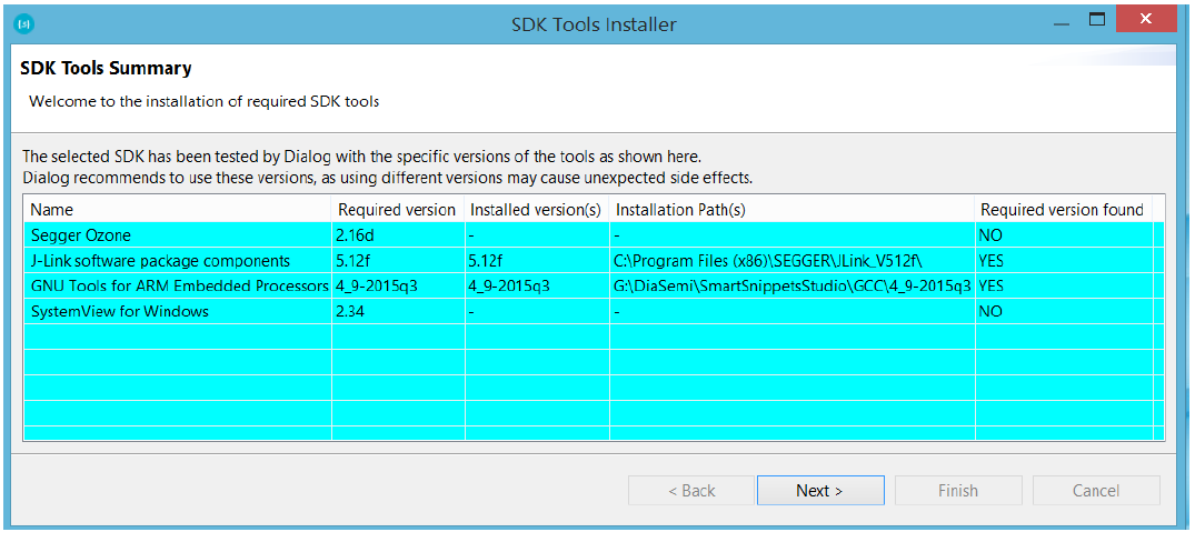
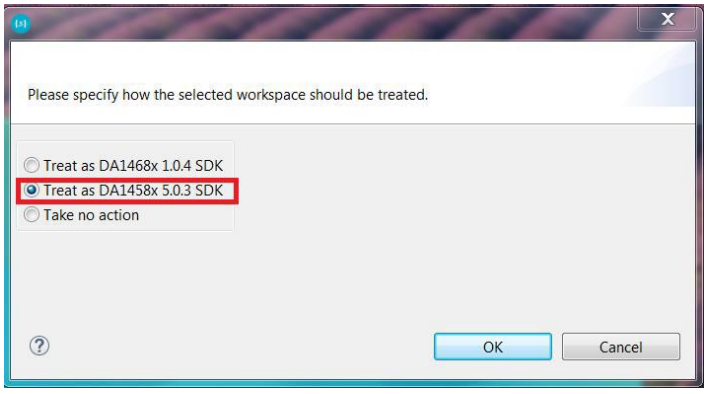
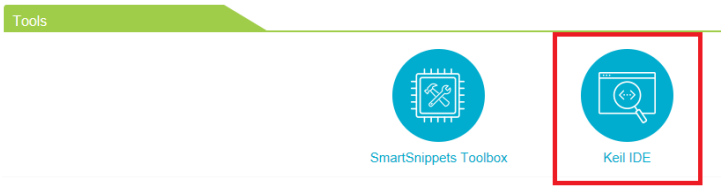


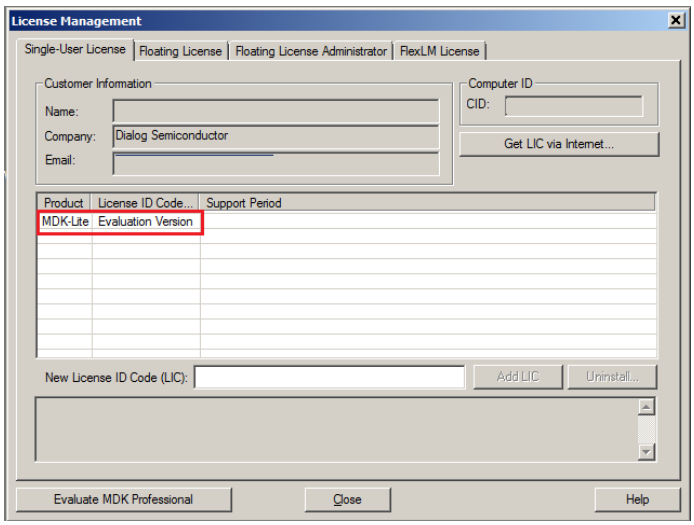
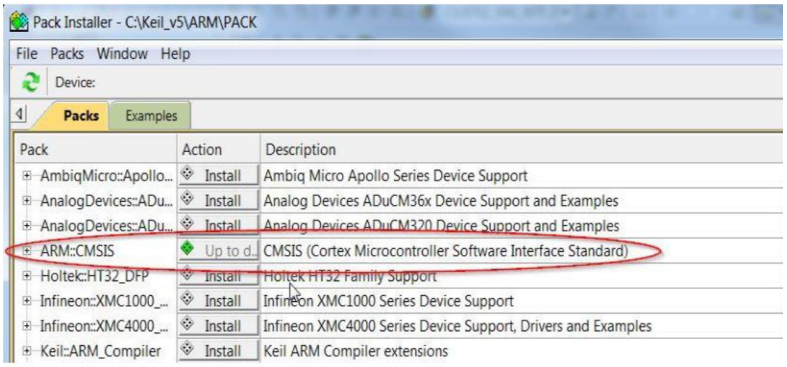
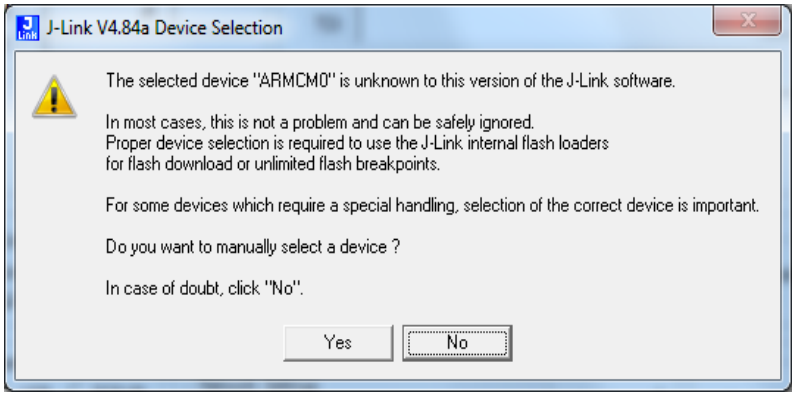
Figure 4: SDK tools installer

And now, the SmartSnippets Studio is ready for use.

Note, that Ozone and GNU tool chain are not needed for the DA1458x family.

Table 7: Installation tools and drivers

3.4 Setting up your PC		
1	Register yourself on the Dialog website http://support.dialog-semiconductor.com/	
2	Download the newest SDK http://support.dialog-semiconductor.com/product/da14580 http://support.dialog-semiconductor.com/product/da14585	
3	Download SmartSnippets Studio https://support.dialog-semiconductor.com/resource/smartsnippetsstudiov153-windows-os https://support.dialog-semiconductor.com/resource/smartsnippetsstudiov153-linux-os	
4	To install the Software development environment, please follow the steps as shown below.	
3.4.1 SmartSnippets Studio		
1	Open SmartSnippets Studio and select: Treat as DA1458x	
2	Press: Keil IDE from the Tools section of the welcome menu	

3	The MDK-Lite version of KEIL can be used. It supports applications up to 32 KB code-size.	
4	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” or the “Update” button to the right of “ARM::CMSIS” package if not up to date.	
5	If the installation is successful, the pack installer window should look like this.	
6	Having installed the SEGGER JLink Software, this screen may occur, so: Select ‘yes’ and in the next window Select ‘M0’.	

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

3.5 Software Development Kit content

3.5.1 Tools

Web-link: www.dialog-semiconductor.com/support, go to section **Products** for selecting chip (e.g. DA14585) then **Software & Tools**, and finally **Tools** section.

SmartSnippets Toolbox

SmartSnippets is a framework of PC based tools to control DA14580/581/583/585/586 development kit, consisting of:

- Power Profiler : Real time current consumption measurement to for the DA14580/581/583 motherboard
- OTP Programmer: Tool for OTP memory programming
- UART/JTAG booter: Tool for downloading hex files to DA14580/581/583 SRAM over UART or JTAG
- SPI & EEPROM programmer: A tool for SPI & EEPROM flash programming
- Sleep Mode Advisor: Calculation tool to determine most optimal sleep modes
- RF Master. This tool supports the Bluetooth SIG defined Direct Test Mode for RF PHY testing of Bluetooth low energy devices. In addition it supports a number of vendor specific HCI commands for device testing.

3.5.2 SDK documents

- UM-B-003, DA14580_581_583 Software development guide
- UM-B-004, DA14580_581_583 Peripheral drivers
- UM-B-005, DA14580_581_583 Peripheral examples
- UM-B-006, DA14580 Sleep mode configuration
- UM-B-007, DA14580 Software Patching over the Air (SPOTA)
- UM-B-008, DA14580_581_583 Production test tool
- UM-B-010, DA14580_581_583 Proximity application
- UM-B-011, DA14580 Memory map – scatter file
- UM-B-012, DA14580 Secondary boot loader
- UM-B-013, DA14580 External Processor Interface over SPI
- UM-B-014, DA14580_581 Development Kit
- UM-B-015, DA14580_581_583 Software architecture
- UM-B-017, DA14580 GTL interface Integrated Processor Application
- UM-B-079 DA14585 & DA14586 Software Platform Reference (SDK 6.0.4)
- UM-B-080 DA14585 & DA14586 Software Developer's Guide (SDK 6.0.4)
- UM-B-082 DA14585/586 SDK5.0.4 to SDK6 Porting Guide_1v0

3.5.3 SDK source code examples (created with Keil IDE)

Web-link:

- This folder holds all the necessary folders needed for DA14580/581/583/585/586 application development.
  - **projects\target_apps\ble_examples**

The folder contains the following subfolders and in each one of them resides the respective project file. DA14585/586, only Keil_5 is supported:

Table 8: SDK Examples

Folder	Project File	Description
prox_monitor_ext\Keil_5	prox_monitor_ext.uvproj	Proximity Monitor (External processor configuration) (*see below for device selection) USB MONITOR (**see below)
prox_reporter_ext\Keil_5	prox_reporter_ext.uvprojx	Proximity Reporter (External processor configuration) (*see below for device selection) USB MONITOR (**see below)
prox_reporter\Keil_5	prox_reporter.uvprojx	Proximity Reporter (Integrated processor configuration) (*see below for device selection)
ble_app_barebone\Keil_5	ble_app_barebone.uvprojx	Barebone project (Integrated processor configuration) (*see below for device selection)
ble_app_peripheral\Keil_5	ble_app_peripheral.uvprojx	Peripheral (Integrated processor configuration) (*see below for device selection)
ble_app_profile\Keil_5	ble_app_profile.uvprojx	Profiles (Integrated processor configuration) (*see below for device selection)
prox_reporter_ext_spi\Keil_5	prox_reporter_ext_spi.uvprojx	Proximity Reporter (External processor) SPI version (*see below for device selection)

The device can be easily selected as shown below:.

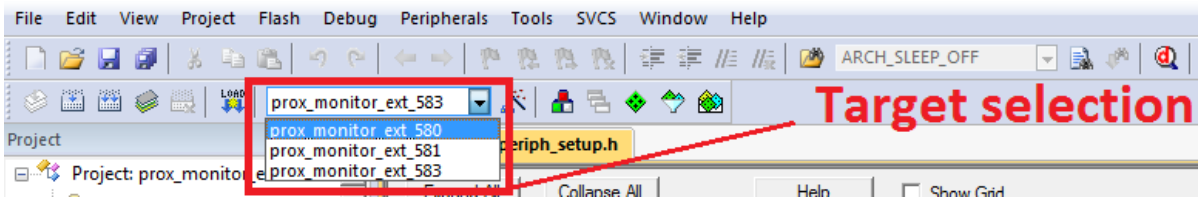


Figure 5: DA14580 Target selection

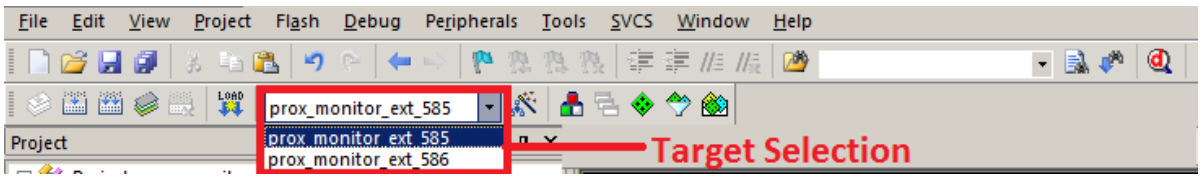


Figure 6: da14585 Target selection

**USB MONITOR/USB REPORTER can be easily selected as shown below:
Open: user_config/user_periph_setup.h and select ‘Configuration Wizard’

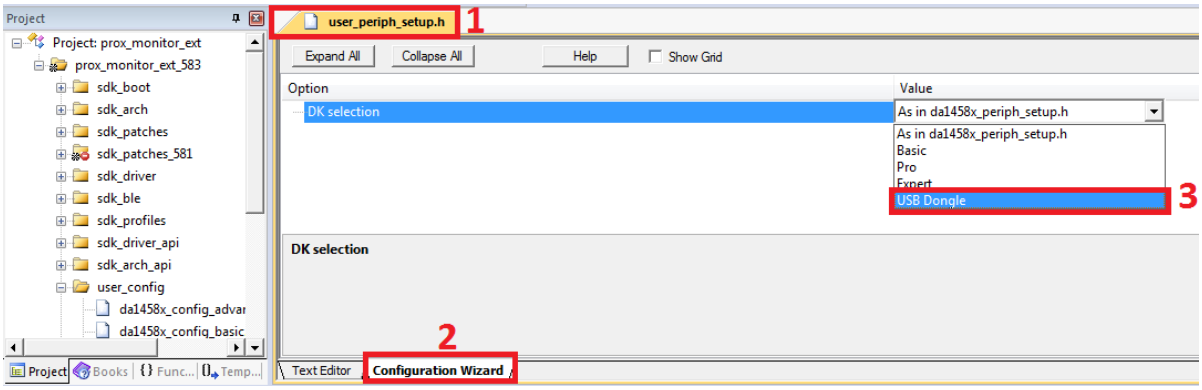


Figure 7: DA14580 USB selection

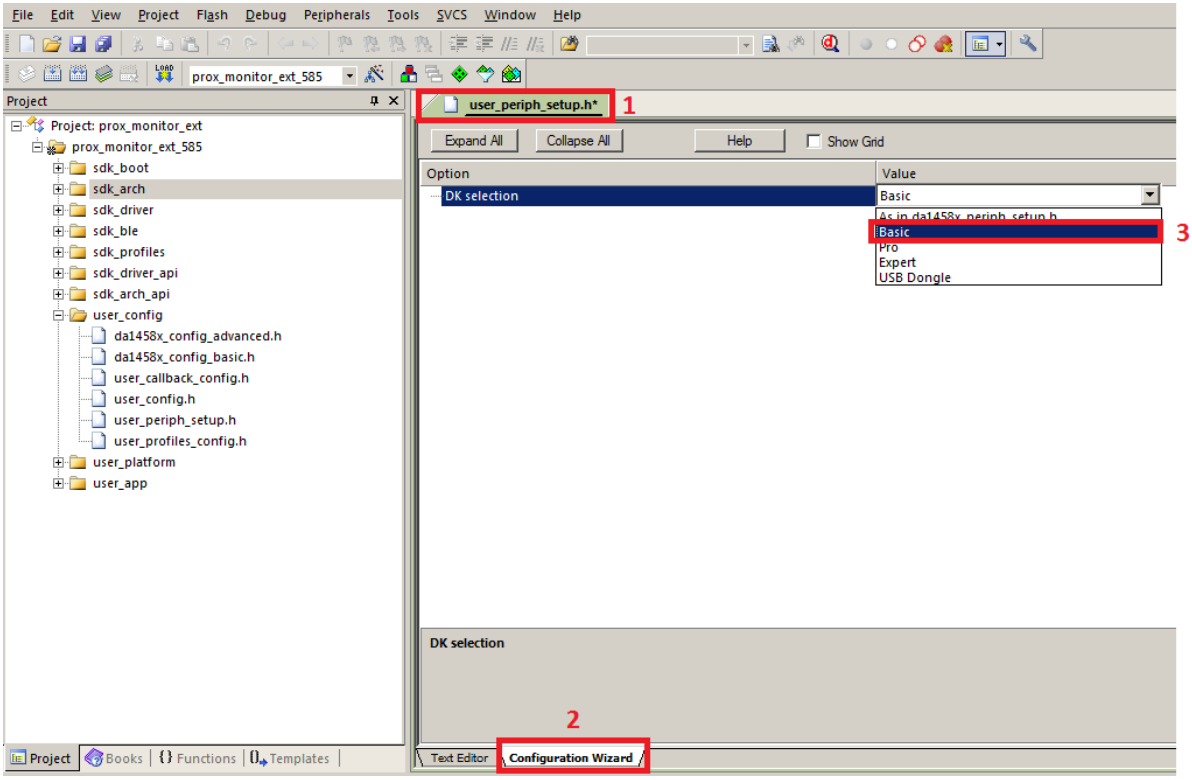


Figure 8: DA14585 USB selection

- **projects\target_apps\prod_test:** This folder includes the source code of the production test firmware. Refer to UM-B-008_DA14580_581_583_Production_test_tool.pdf for more information how to build and use it.

- **5.0.1.140:** This folder holds the DA14580/581/583 PC applications:
  - ***projects\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.
  - **binaries\host\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.

- **projects\target_apps\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].

- **6.0.x:** This folder holds the DA14585/586 PC applications:
  - ***projects\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.
  - **binaries\host\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.
  - **projects\target_apps\peripheral_examples:** This folder includes sample code of how to use the peripheral blocks of the DA14585/586 (e.g. UART, SPI, I2C etc.) bundled to a demo-kit.

- **utilities:**
  - **utilities\prod_test\prod_test_cmds:** This folder includes the source code of the production test tool. Refer to UM-B-008_DA14580_581_583_Production_test_tool.pdf for more information how to build and use it.

    For DA14585/586, refer to UM-B-79_DA14585_586_SDK_6_Software_Platform_Reference.

3.6 Pinning

Below the pin assignment information for the Dialog semiconductor DA1458x BLE SoC family (DA14580, DA14581, DA14583, DA14585 and DA14586) are provided as references for the user.

3.6.1 DA14580 pin assignment

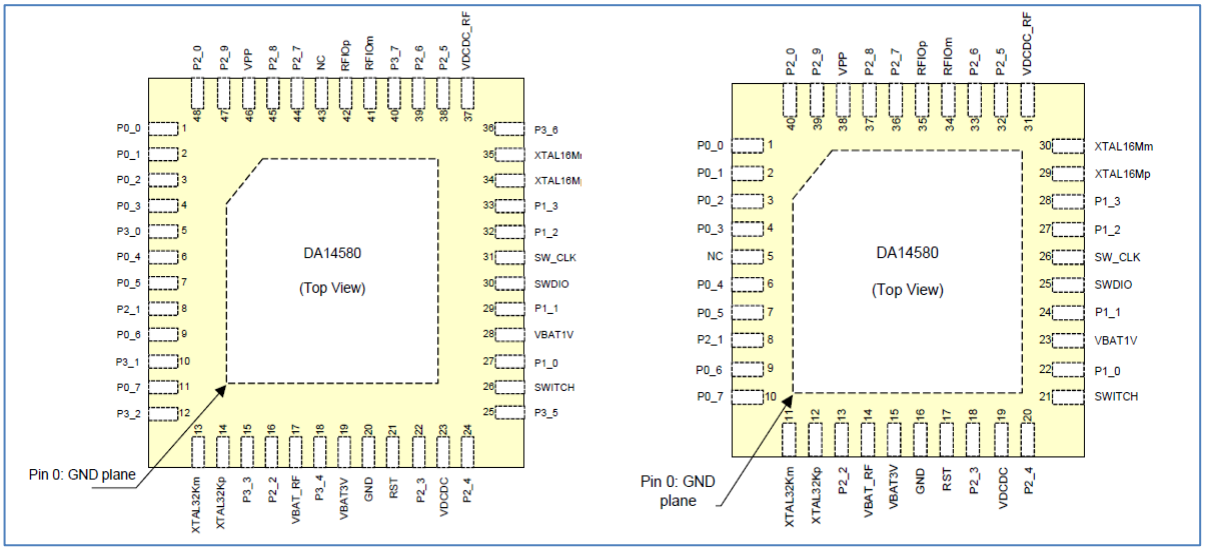


Figure 9: DA14580 - QFN48 (left) and QFN40 (right) Pin assignment

3.6.2 DA14581 pin assignment

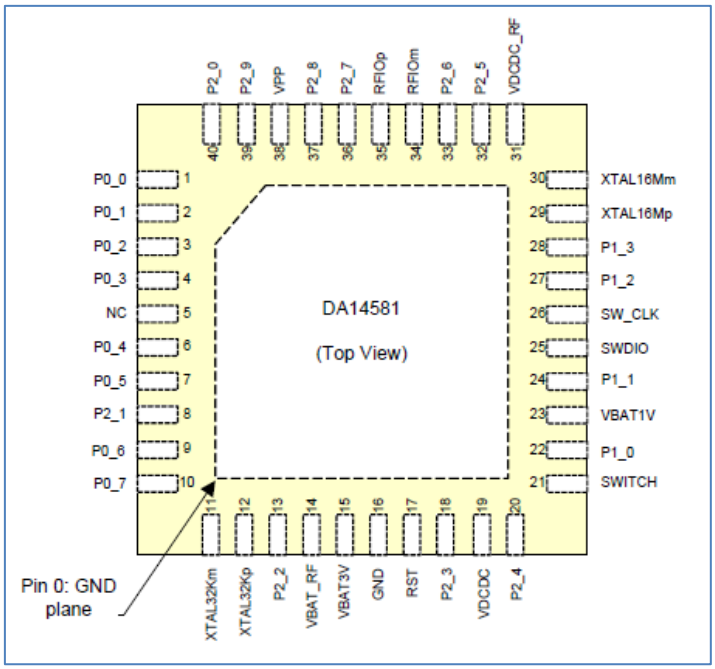


Figure 10: DA14581- QFN40 Pin assignment

3.6.3 DA14583 pin assignment

In Figure 11 the pin out of the DA14583 is shown. Different, compared to the DA14580/581, are the connections to the internal SPI flash memory.

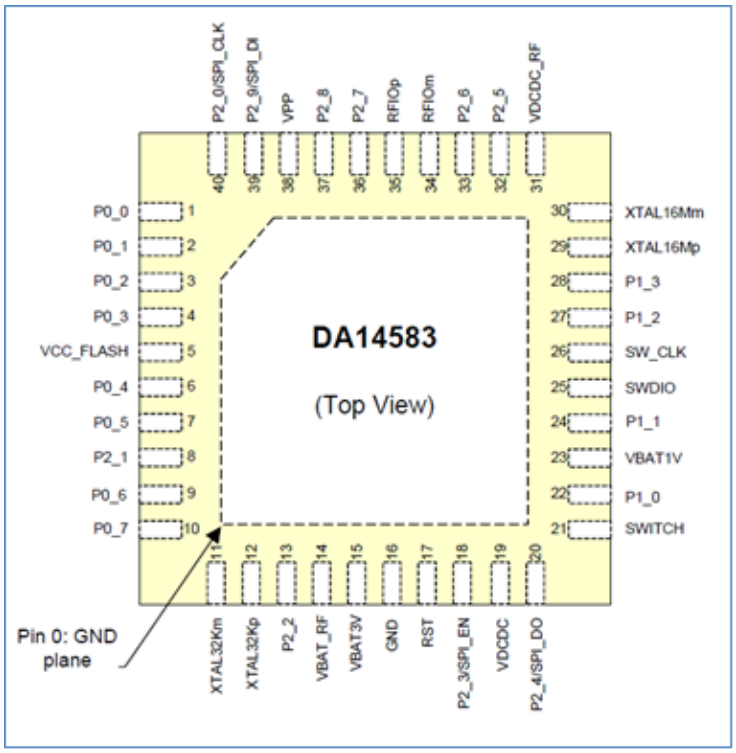


Figure 11: QFN40 pin assignment from datasheet

Table 9: SPI connections

port DA14583	function	remark
P2_0	SPI_CLK	SCLK (Note 1)
P2_9	SPI_DI	MOSI (Note 1)
P2_4	SPI_DO	MISO (Note 1)
P2_3	SPI_EN	not to be used for external SPI (!)
	VCC_FLASH	power for internal Flash Memory
	GND	

Note 1 shared with internal flash memory

When external SPI components are used, SPI_EN is occupied for internal use. Another pin should be chosen for SPI_EN of the external component.

By using a Secondary Bootloader the proper pins are programmed to load the booting software from the SPI-memory at startup.
See more info from UM-B-012 [10].

3.6.4 DA14585 pin assignment

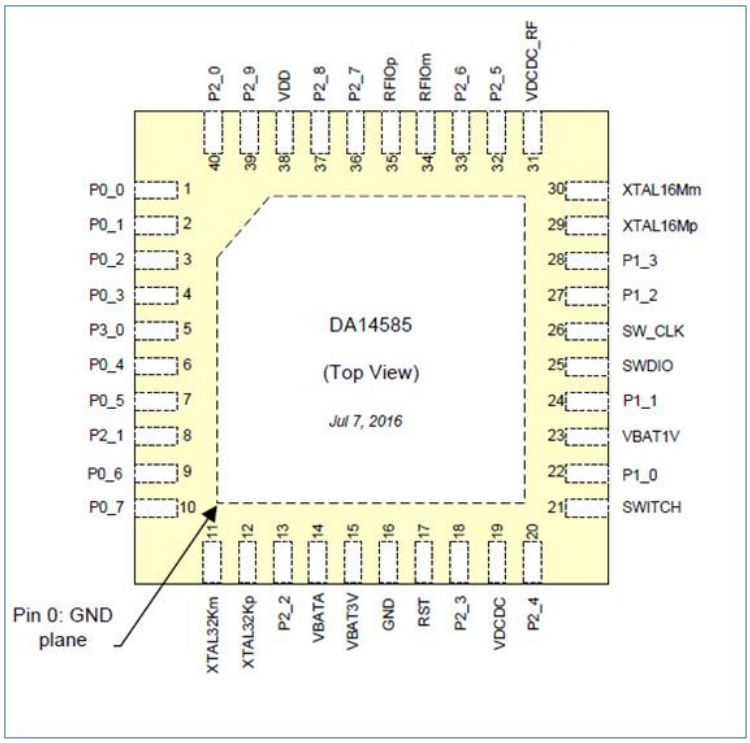


Figure 12: DA14585 - QFN40 pin assignment

DA14585 - QFN40 pins assignment differentiation:

- DA14580/1/3: **Pin 38**, **VDD** is assigned instead of **VPP**. VDD is input and it is used for testing purposes only. In normal operation this pin must left floating. Please notice that for DA14585 OTP programming, the 6.8V external voltage is not required.
- To DA14583/6: **Pin 5**, is assigned to P3_0 instead of VCC_ Flash for DA14583 and DA14586. On DA14585, this is a general purpose IO pin.

3.6.5 DA14586 pin assignment

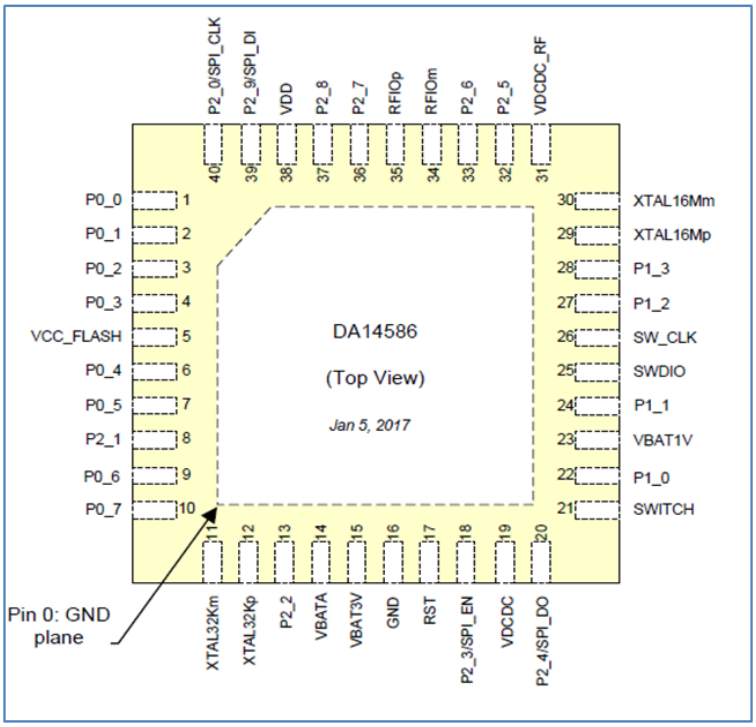


Figure 13: DA14586 - QFN40 pin assignment

DA14586 - QFN40 pins assignment differentiation

- To DA14580/1/3: **Pin 38**, VDD is assigned instead of **VPP**. VDD is input and it is used for testing purposes only. In normal operation this pin must left floating. Please notice that for DA14586 OTP programming, the 6.8V external voltage is not required.
- To DA14580/1/5: **Pin 5**, **VCC_Flash** is assigned instead of **P3_0.** VCC_ Flash is used for supplying the internal flash memory for DA14586. Same assignment is valid for DA14583.

In addition, as DA14586 incorporates a 2Mbit flash memory, four pins are multiplexed with internal flash data pins:

Table 10: SPI connections

Pin number DA14586	Port DA14586	function	Remark
40	P2_0	SPI_CLK	SCLK (Note 1)
39	P2_9	SPI_DI	MOSI (Note 1)
20	P2_4	SPI_DO	MISO (Note 1)
18	P2_3	SPI_EN	Not to be used for external SPI (!)
5		VCC_FLASH	power for internal Flash Memory
		GND	

Note 1 shared with internal flash memory

When external SPI components are used, SPI_EN is occupied for internal use. Another pin should be chosen for SPI_EN of the external component.

By using a Secondary Bootloader the proper pins are programmed to load the booting software from the SPI-memory at startup.

3.7 PCB design and functionalities

3.7.1 DA14581DEVKT-B (228-01-A)

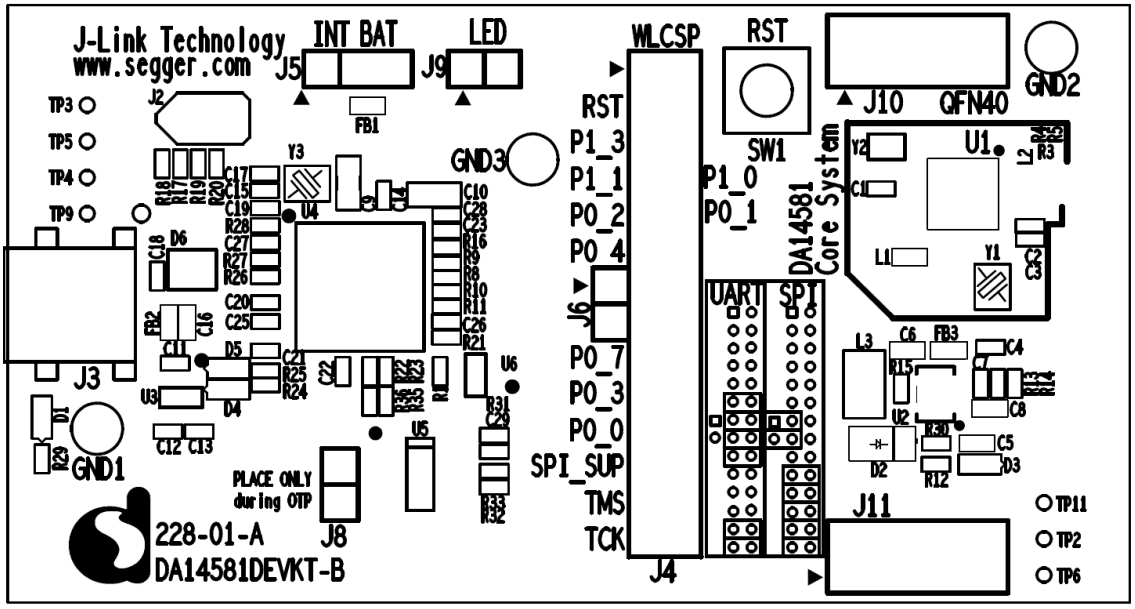


Figure 14: Components on PCB top layer of the DA14580/581Basic kit

The different components and functionalities are shown in Figure 15.
A larger picture is shown in Appendix A

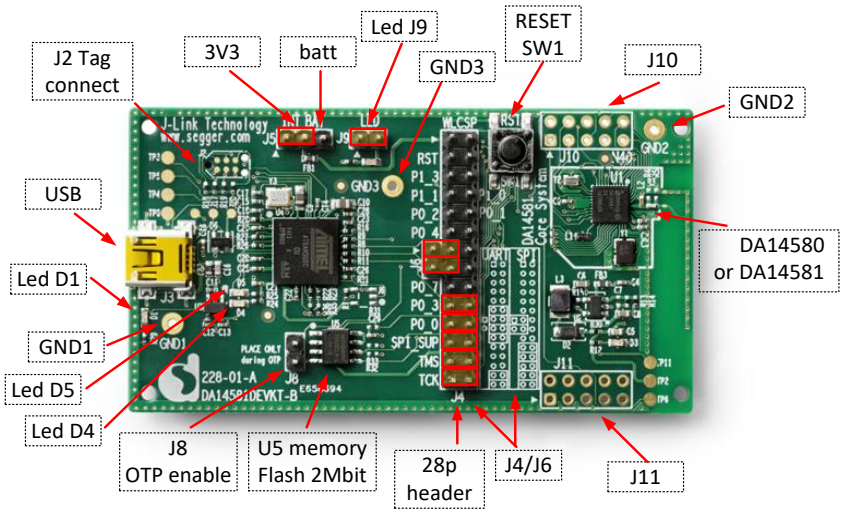


Figure 15: Top view of PCB DA14580/581 with components and functionalities

3.7.2 DA14583DEVKT-B (285-02-A)

The top-screen layer of the DA14583 Basic Kit PCB is shown in Figure 16.

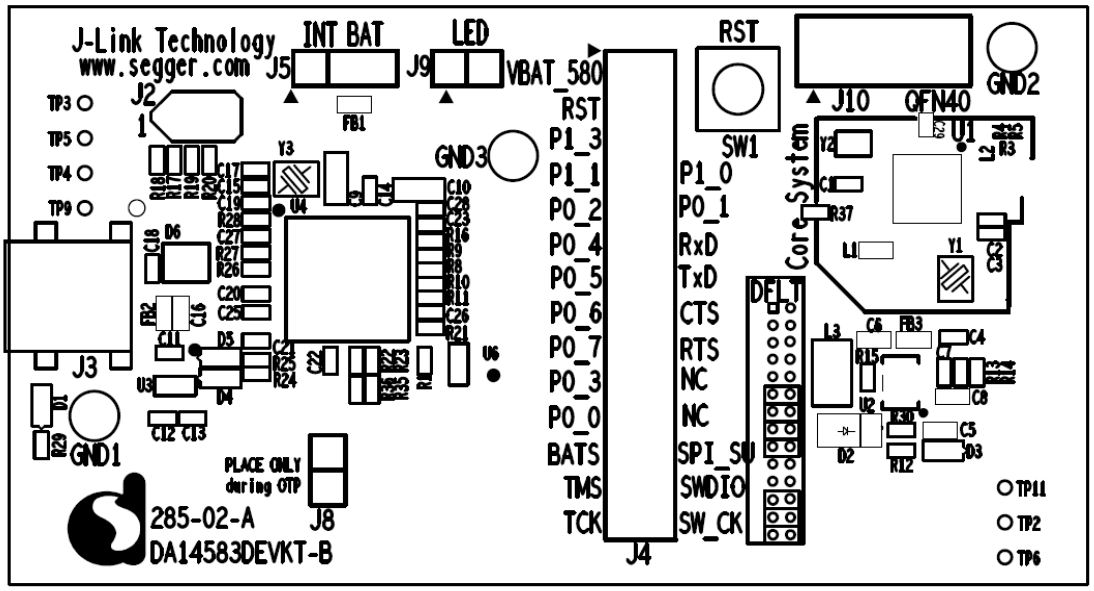


Figure 16: Components on PCB top layer of the DA14583 Basic kit

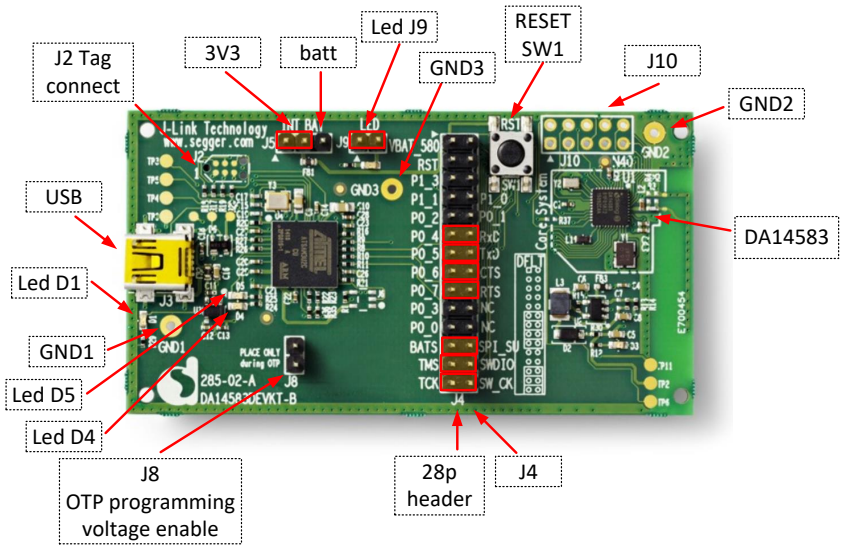


Figure 17: Top view of PCB DA14583 with components and functionalities

3.7.3 DA1458xDEVKT-B (321-04-A)

DA1458xDEVKT-B supports all SoCs of DA1458x family. The way that this is accomplished is presented on Appendix H and Appendix I. In this section, DA1585 and DA14586 will be presented.

Basic block diagram of the hardware implementation of DA1458xDEVKT-B below:

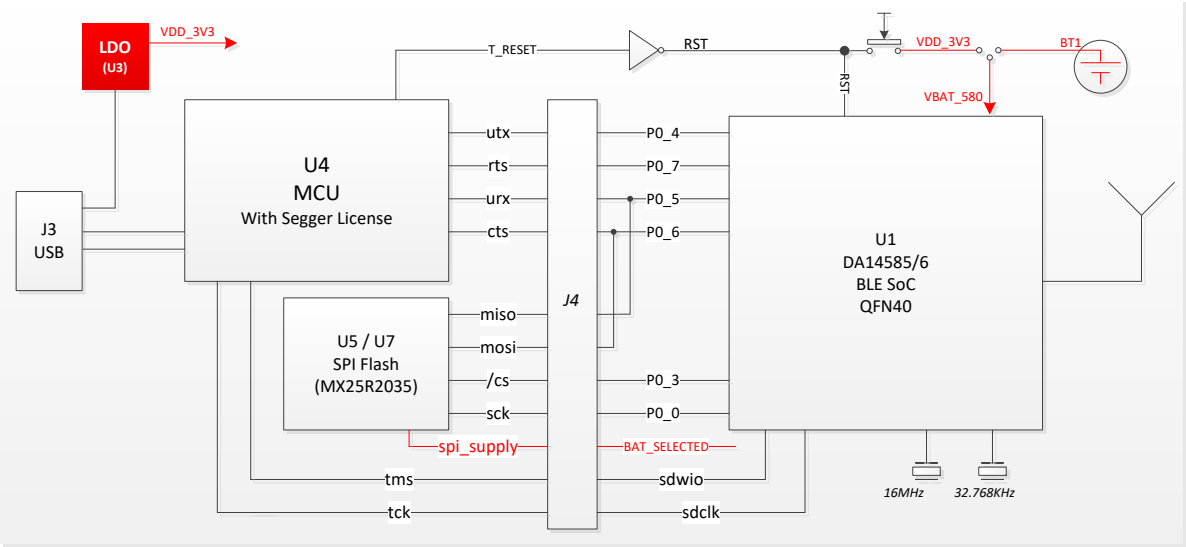


Figure 18: DA1458x DEKT-B block diagram

The top-screen layer of the DA1458x Basic Kit PCB is shown in Figure 19. Schematics are presented on section 3.8.4

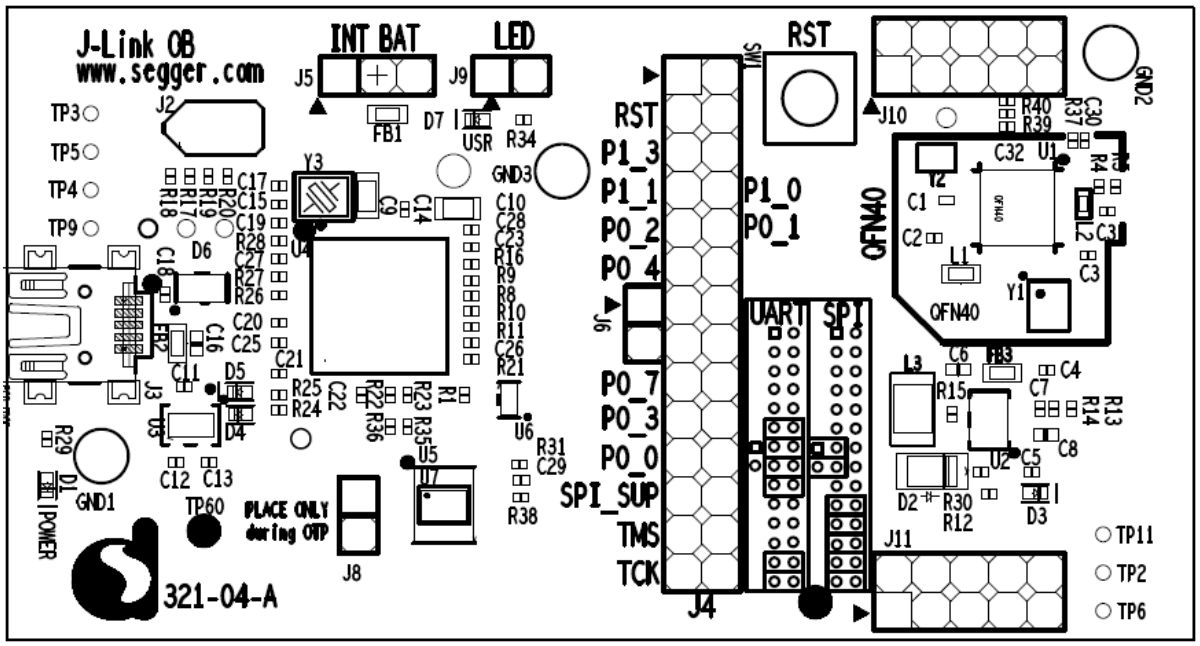


Figure 19: Components on PCB top layer of the DA14585/6 Basic kit (321-04-A)

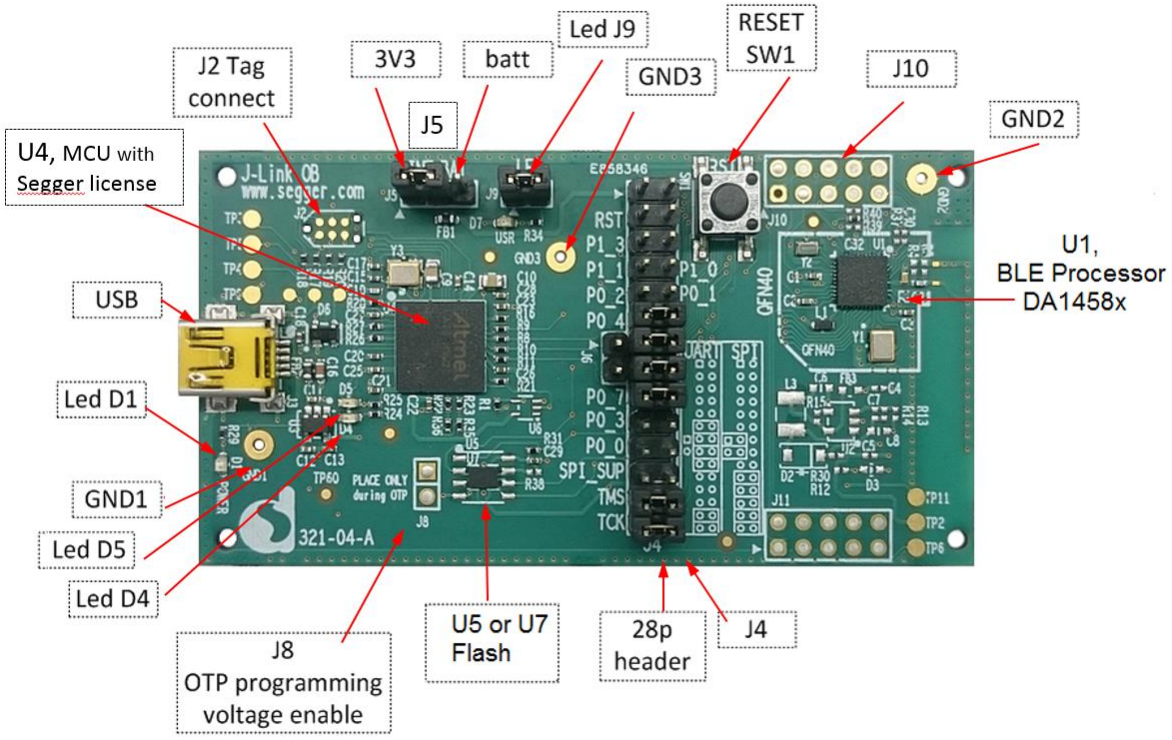


Figure 20: Top view of PCB DA1458x with components and functionalities

Components of Basic development kit:

- U1: Dialog BLE SoC, DA14585 or DA14586 in QFN40 package. DCDC converter of SoC is configured in Buck mode. Two crystals are used, 16MHz and 32.768KHz.
- U5 or U7: SPI data flash, MX25R2035. Two packages are available, U5 –WSON MX25R2035FM1IH0 and U7-SOIC8 - MX25R2035FZUIL0 – (default). For DA14586, flash memory is not mounted.
- U4: MCU where Segger license is applied. UART and JTAG connectivity is provided to DA14585/6 SoC
- SW1: Reset push button. By pressing the button, used can hardware reset the processor
- J4: 28 pins header. Signal and power pins of DA14585/6 are provided. Also the interconnection between BLE processor and Segger license MCU for JTAG and UART is done thru jumpers.
- J5: 3 pins connector. DA14585/6 can be supplied by either 3.3V generated from on the development kit or by coin cell battery applied on the bottom side of the development kit.
- J9, two pins header. It enables user LED. LED is driven from P1_0.
- J10: 10 pins header. It is used for monitoring DA14585/6 port 2 pins.
- J11: 10 pins header. Only P3_0 is provided on pin 1. Please notice that for DA14586, P3_0 is not used.

3.8 Schematics and layouts

3.8.1 DA14580DEVKT-B schematic

For the schematics and layout of the board in full detail, please refer to the respective documents on the portal.

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

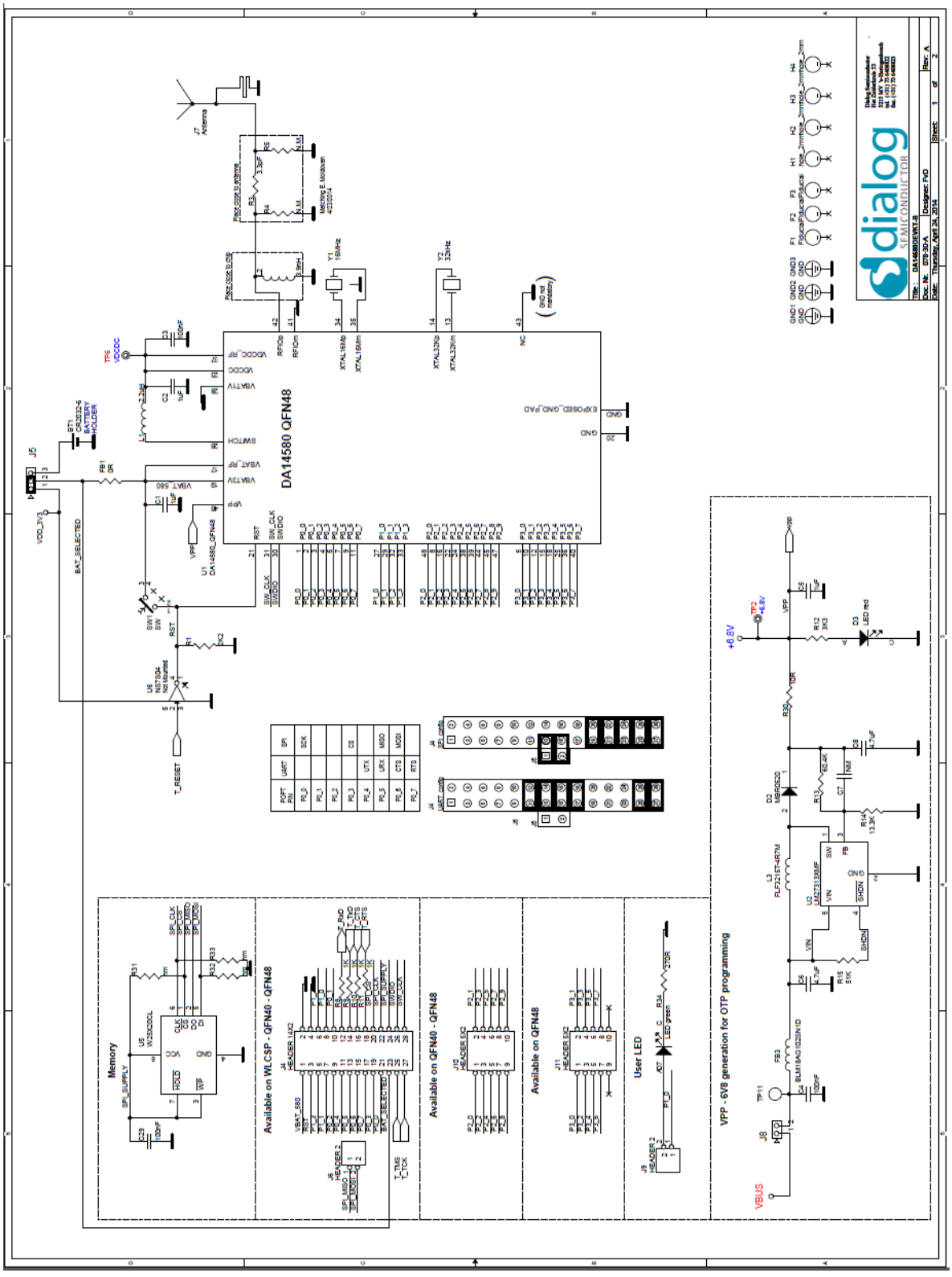


Figure 21: DA14580DEVKT-B_vb part 1

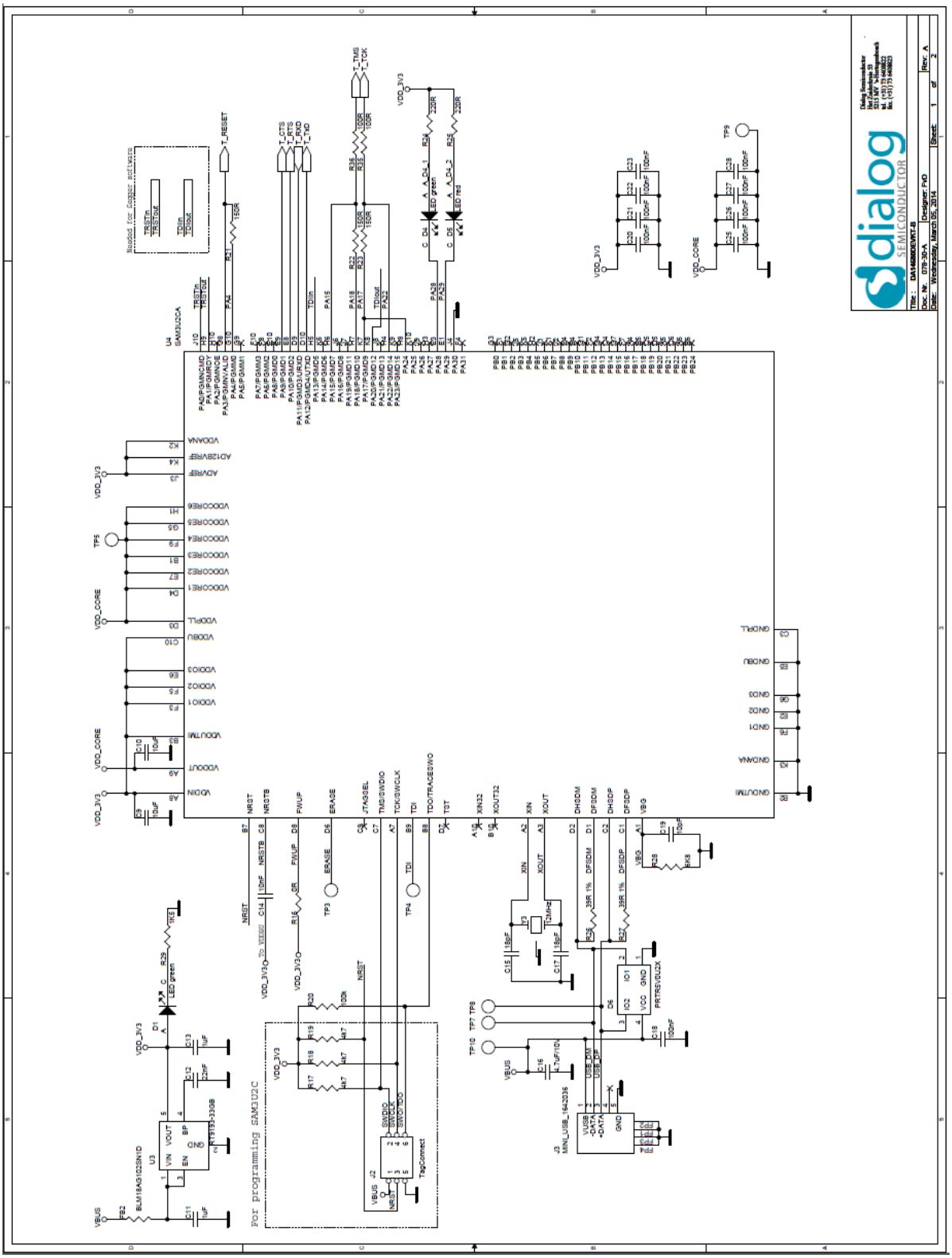


Figure 22: DA14580DEVKT-B_vb part 2

3.8.2 DA14581DEVKT-B_va schematic

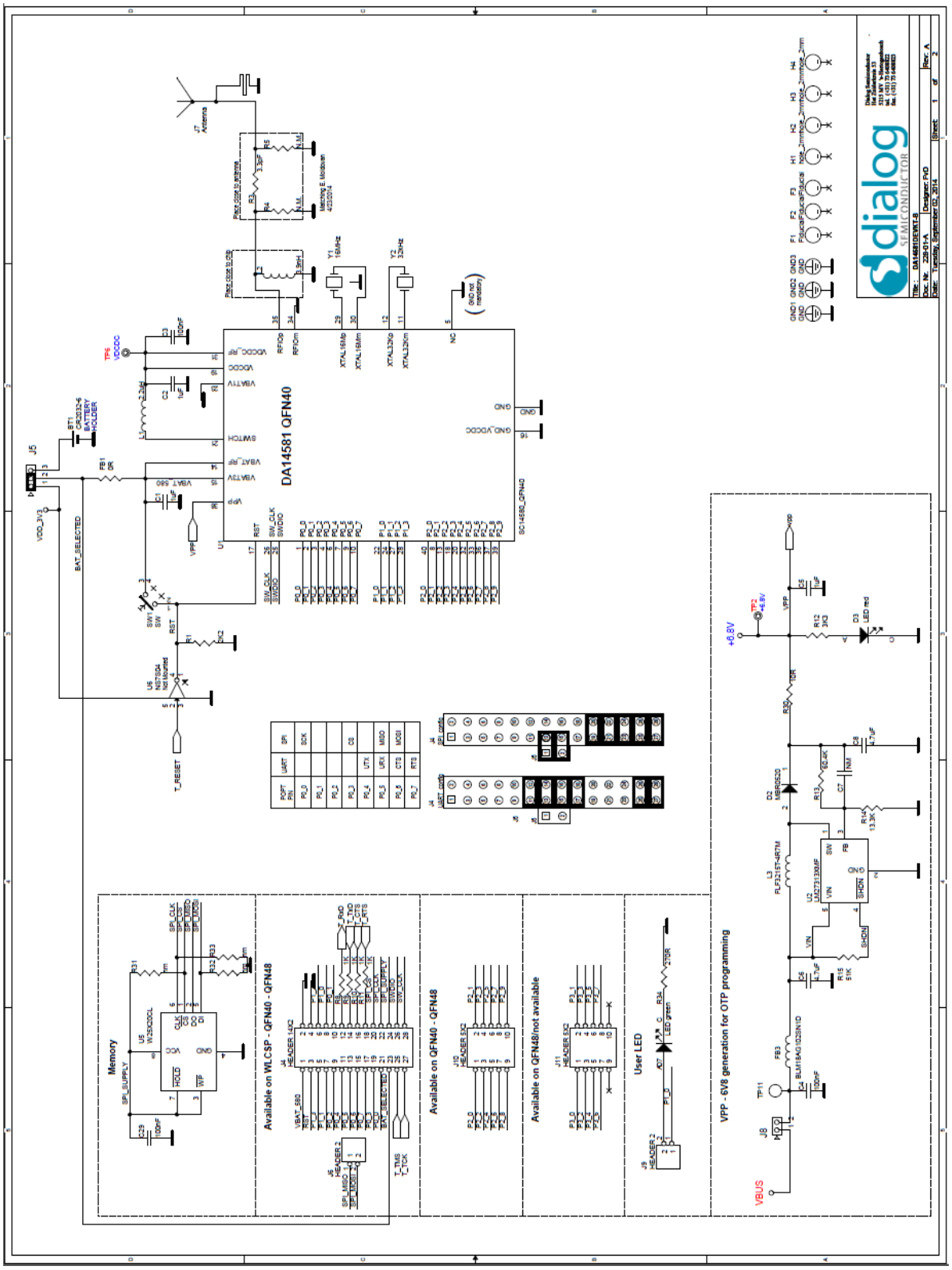


Figure 23: DA14581DEVKT-B_va part 1

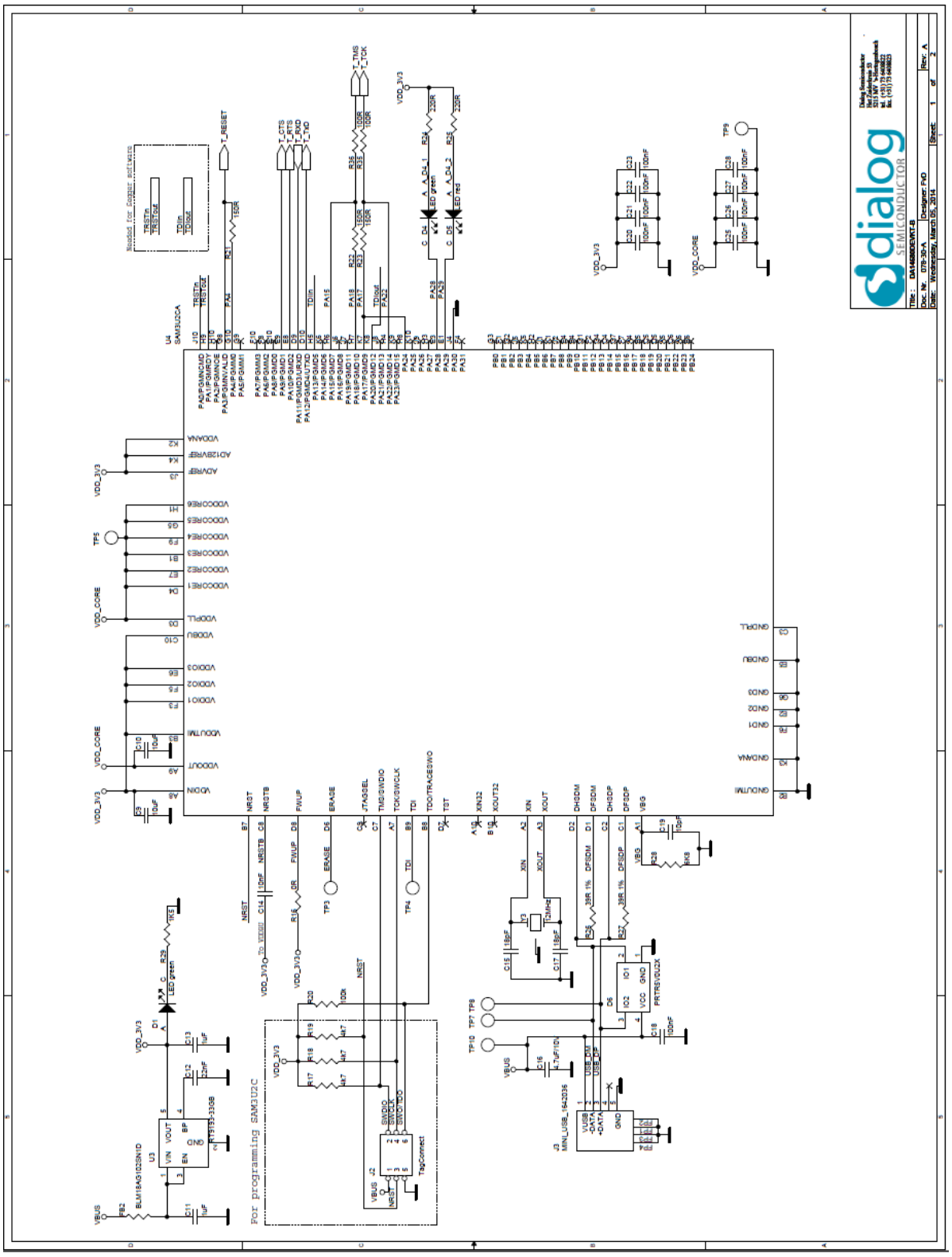


Figure 24: DA14581DEVKT-B_va part 2

3.8.3 DA14583DEVKT-B_va schematic

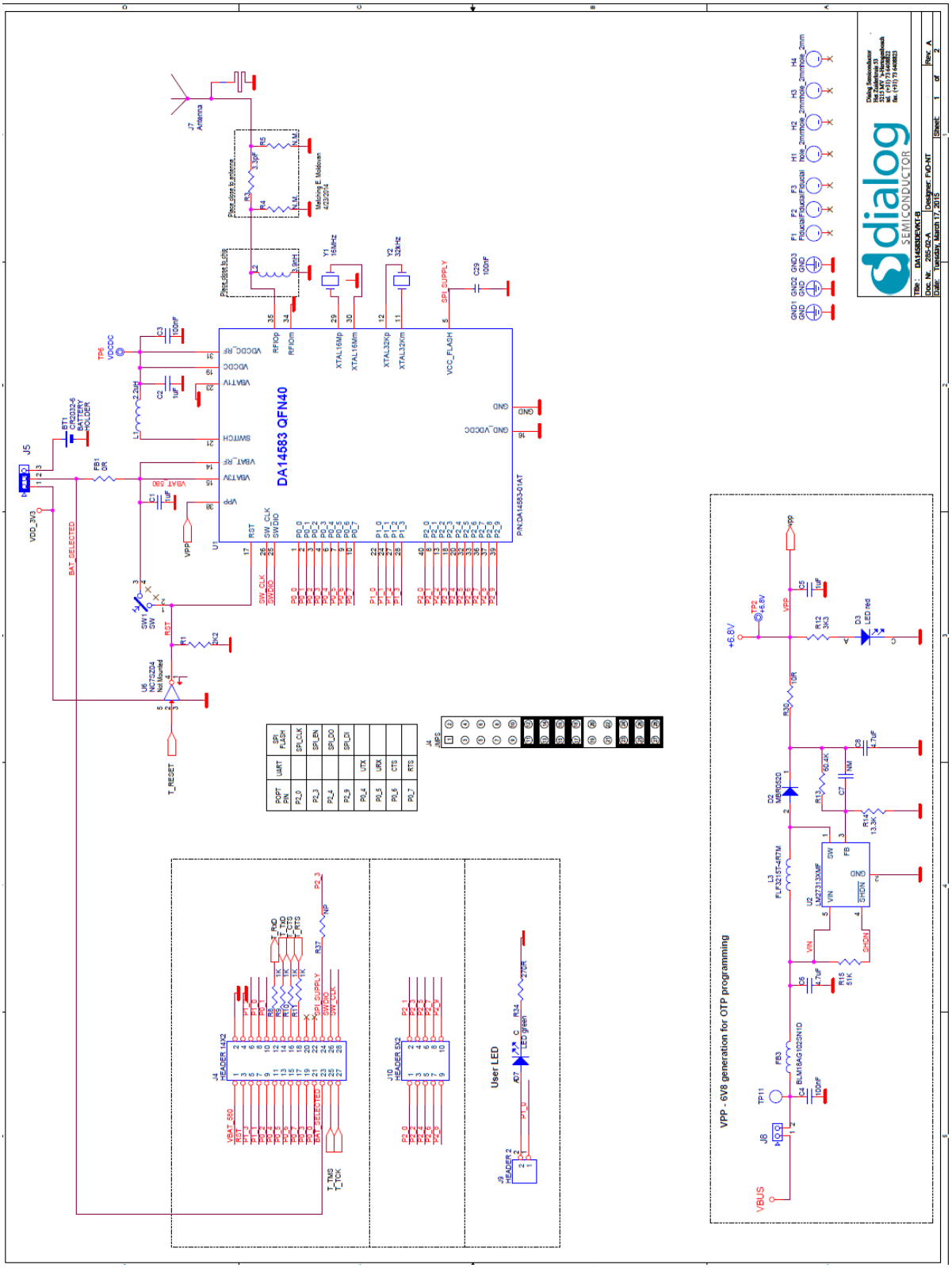


Figure 25: DA14583DEVKT-B_va part 1

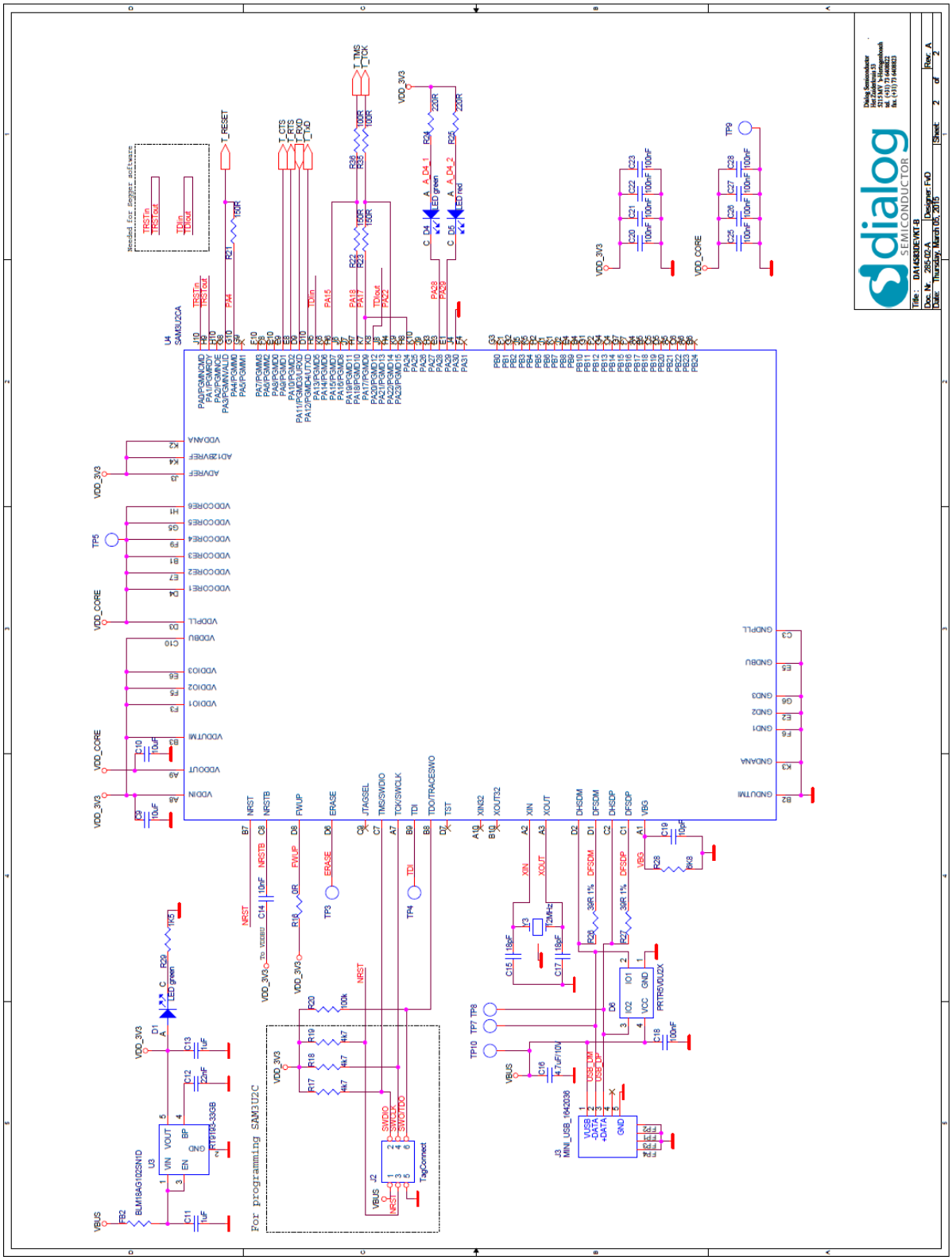


Figure 26: DA14583DEVKT-B_va part 2

3.8.4 DA1458xDEVKT-B (321-04-A) schematic

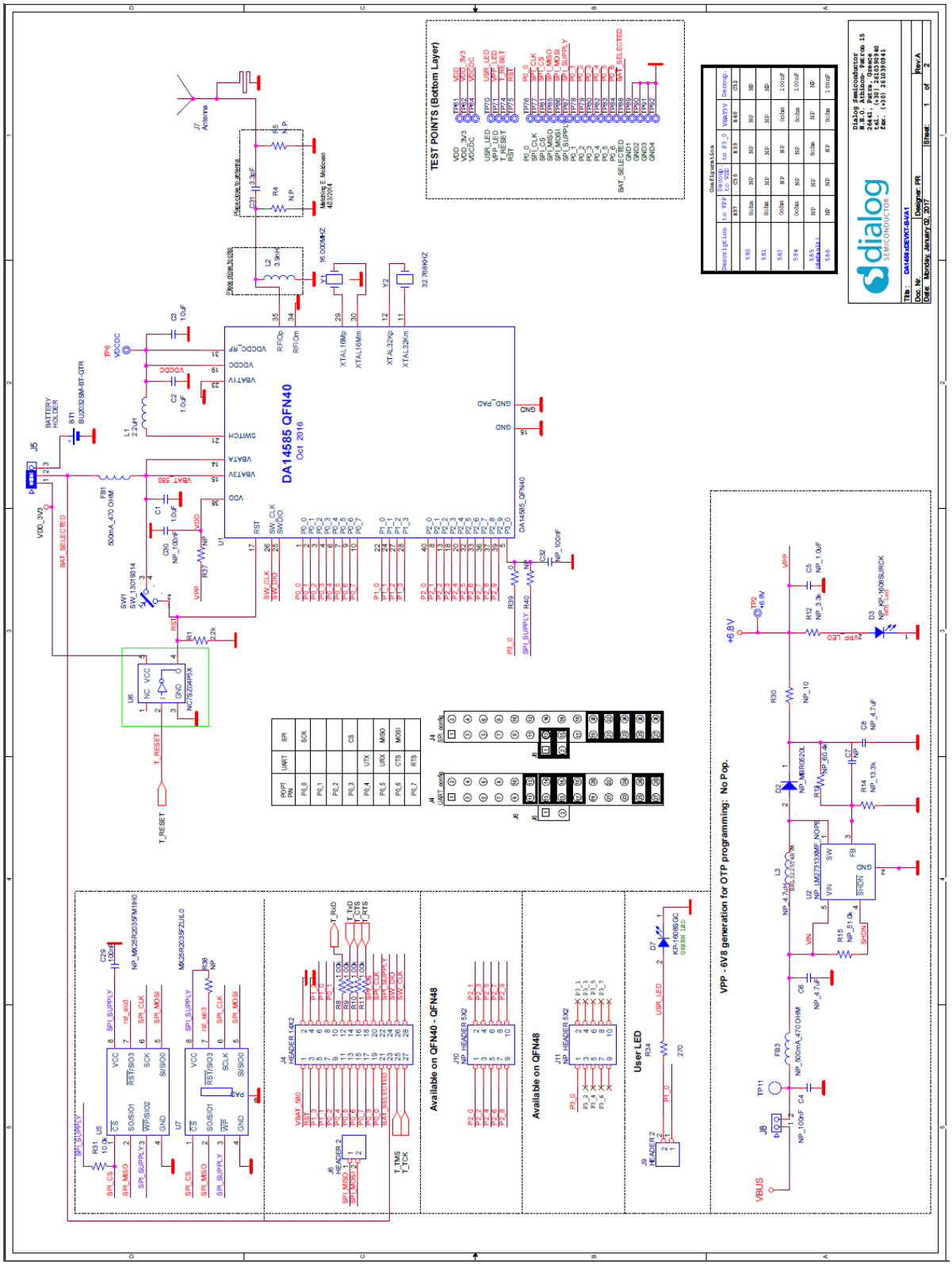


Figure 27: DA1458xDEVKT-B_va part 1

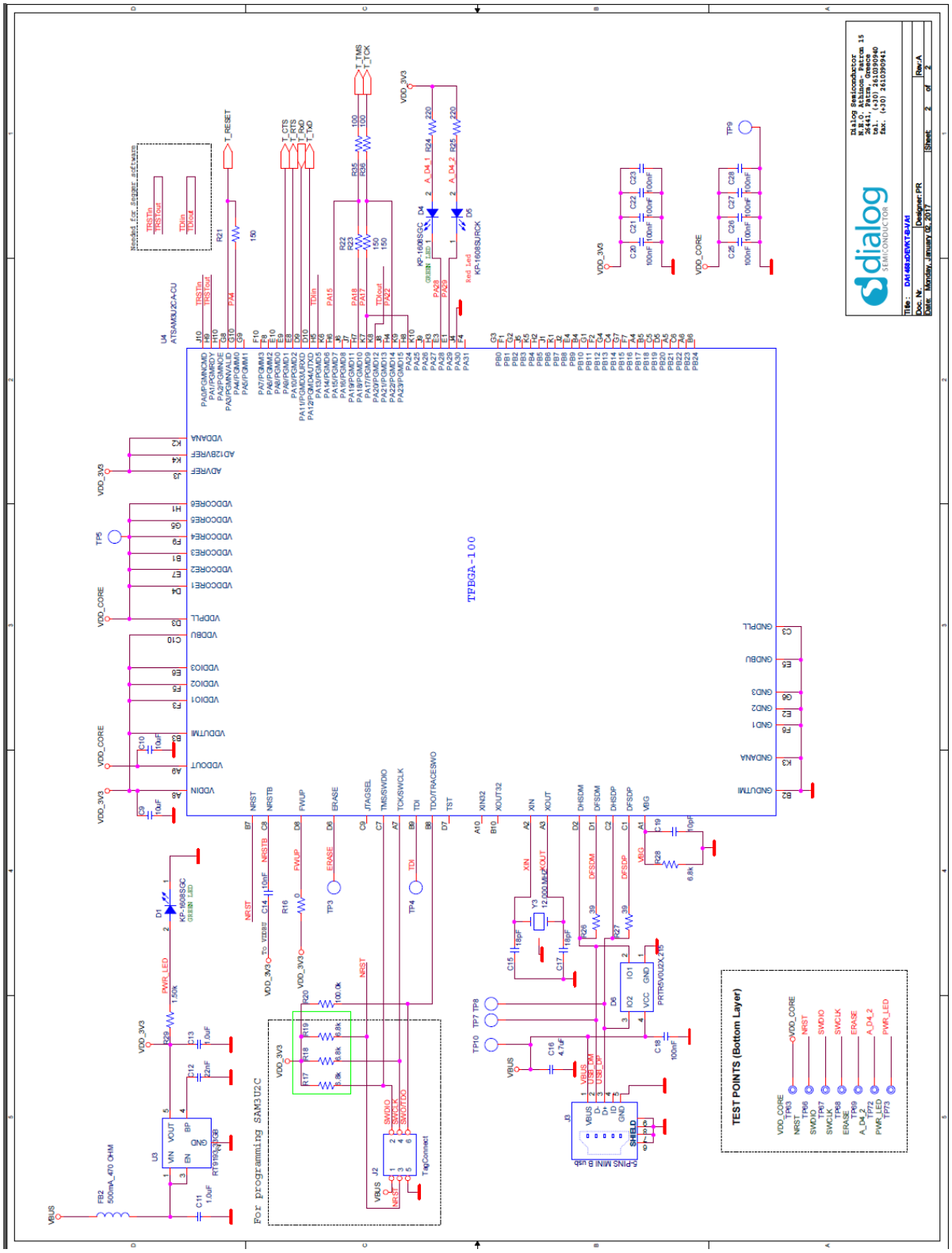


Figure 28: DA158xDEVKT-B_va part 2

3.9 Configuring basic development kit-board by jumper settings

3.9.1 Configuring the DA14580/581 basic kit-board by jumper settings

Different functionalities are shown in Appendix A, and the jumper settings are displayed below.

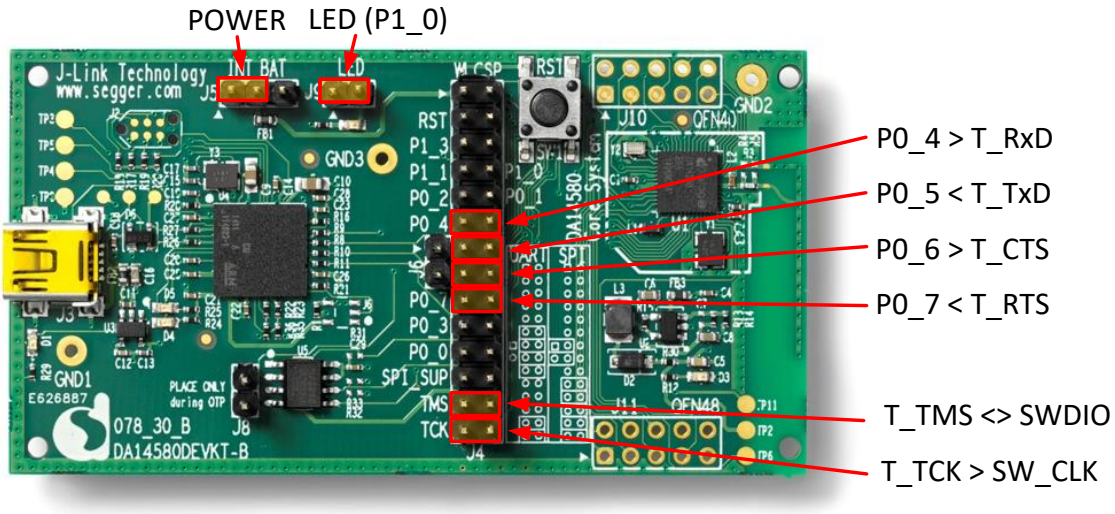


Figure 29: DA14580/581 (Fabrication Default) UART boot settings (Tx P0_4 and Rx P0_5)

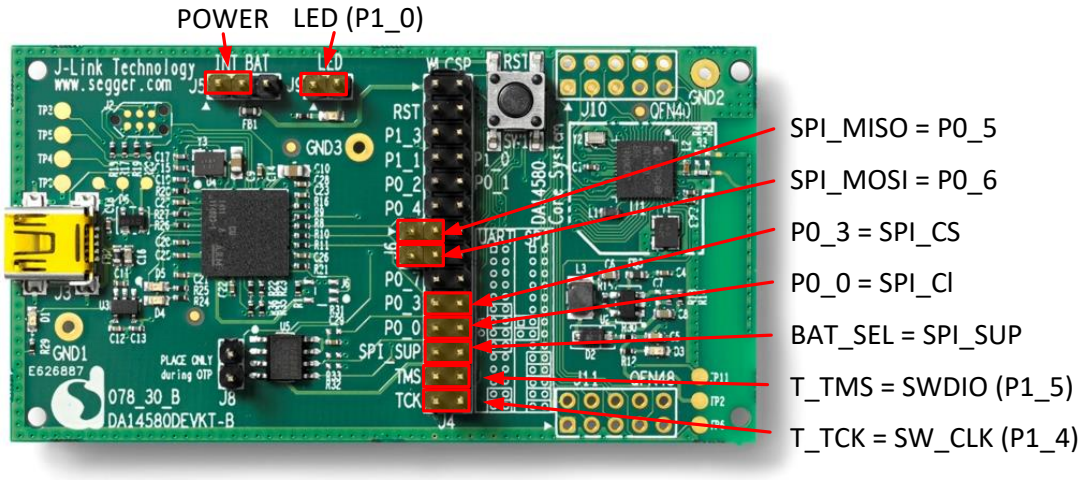


Figure 30: DA14580/581Boot from external SPI memory

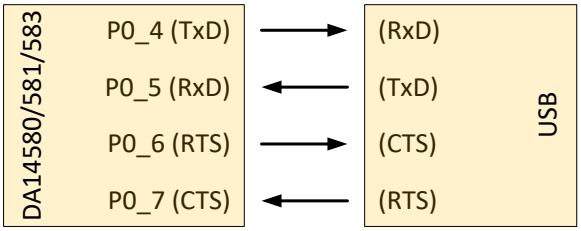


Figure 31: Data direction of UART within J4

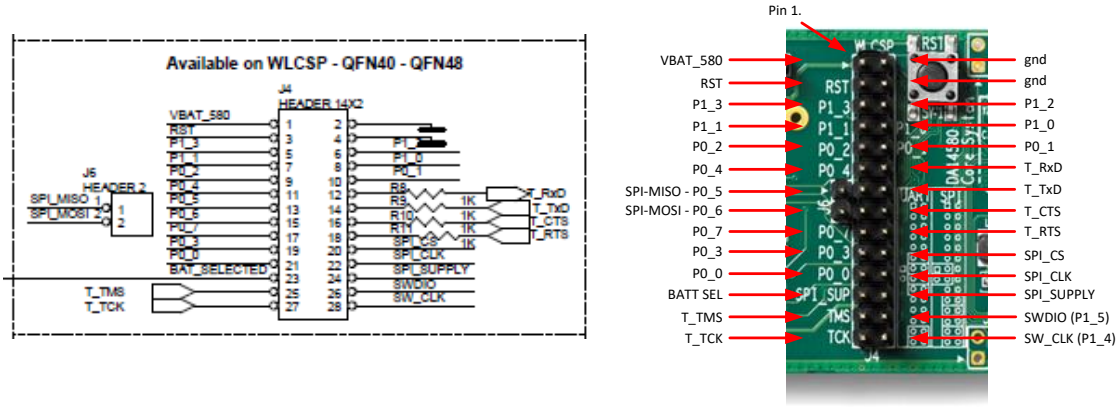


Figure 32: DA14580/581 layout of headers J4 and J6

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

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 in the battery-socket on the back of the PCB.
No battery is needed when running via the USB-mini-cable.

Note 1 The DA14580 is equipped with a QFN48 device and the DA14581 has a QFN40 die mounted.

3.9.2 Configuring the DA14583 basic kit-board by jumper settings

Different functionalities are shown in Appendix A, and the jumper settings are displayed below.

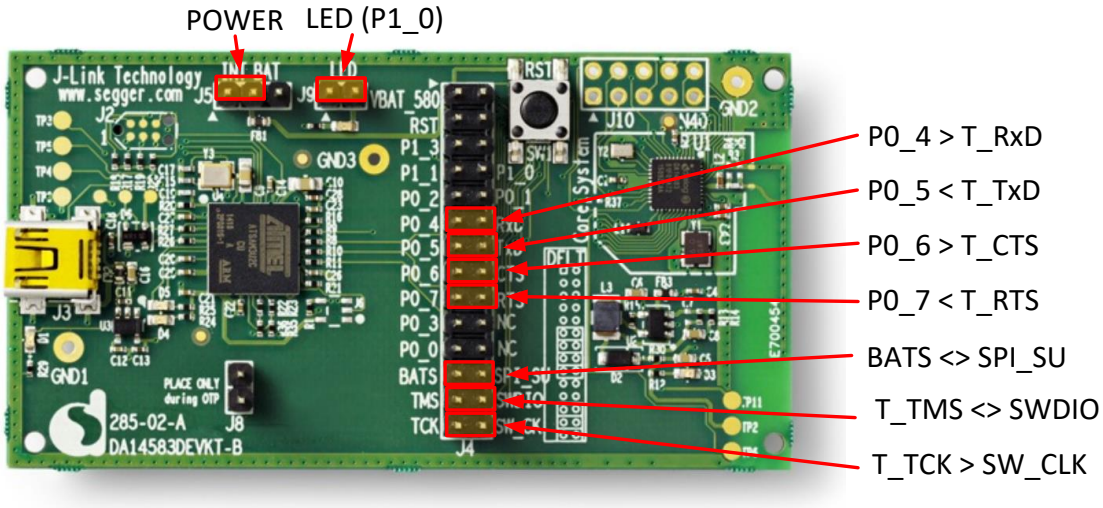


Figure 33: (Fabrication Default) DA14583 UART boot settings (Tx P0_4 and Rx P0_5)

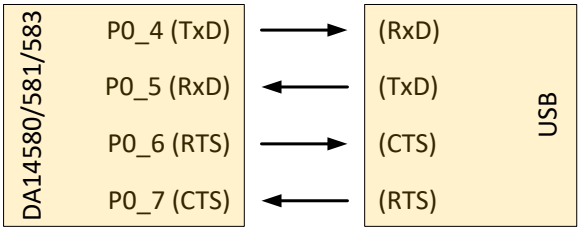


Figure 34: Data direction of UART within J4

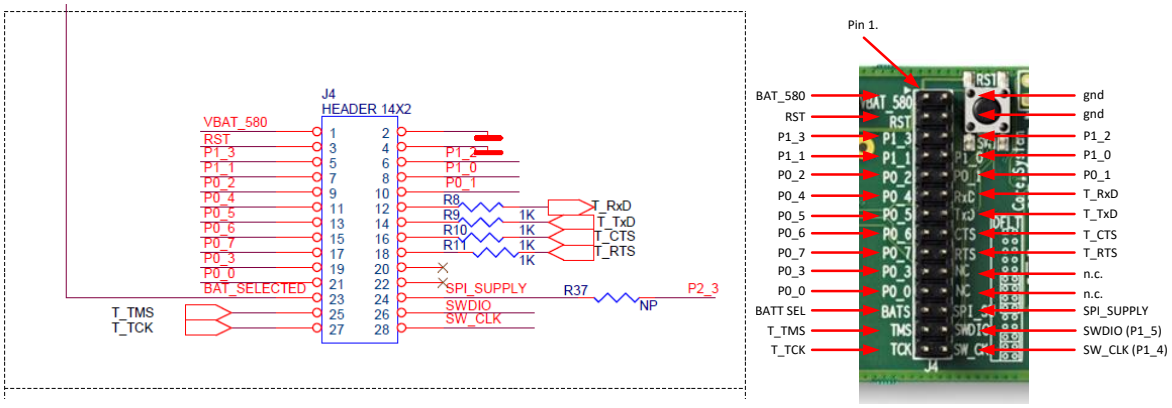


Figure 35: DA14583 Layout of headers J4 and J6

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

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 in the battery-socket on the back of the PCB.
No battery is needed when running via the USB-mini-cable.

Note 1 The DA14583 is equipped with a QFN40 device.

3.9.3 Configuring the DA1458x basic kit-board by jumper settings

DA14585/6 signals are physically separated from the rest of PCB. Connectivity is enabled by shorting specific pins on header J4.

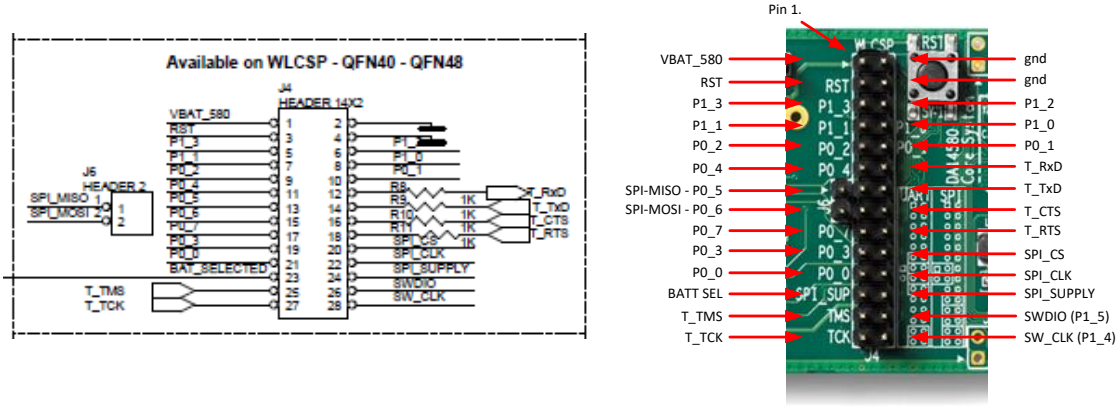


Figure 36: DA1458x layout of headers J4 and J6

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

Please notice that the signals connected on the DA14585/6 SoC are located on the left pins row of header J4, Figure 36.

Different functionalities are enabled with jumper settings.

UART connectivity is enabled by shorting header as presented in Figure 37 below:

UART connection between DA14585/6 and the MCU that provides communication over USB to a PC is realized thru P0_4, P0_5 (for null mode), P0_06 and P0_7 (for full UART),

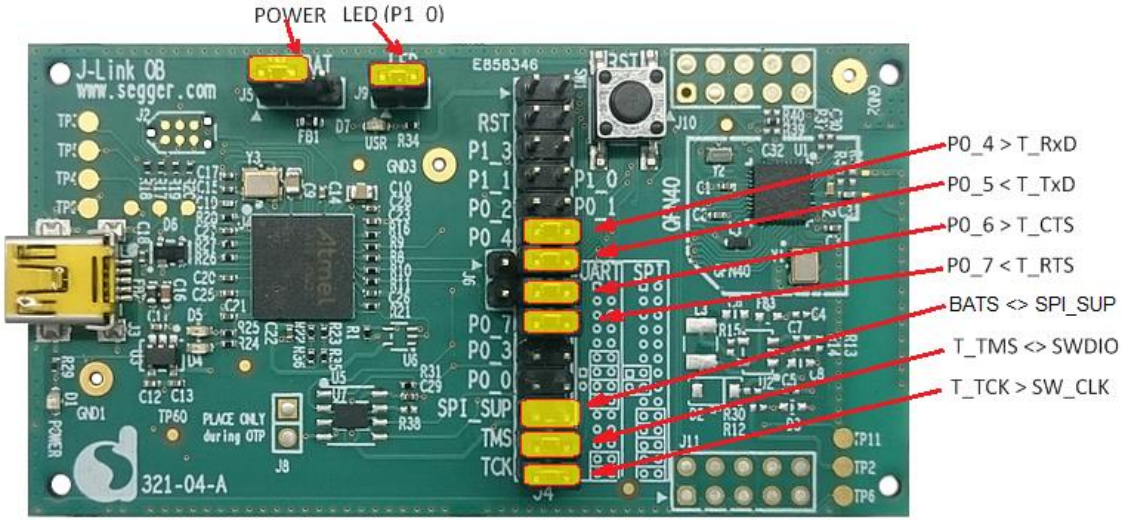


Figure 37: DA1458x (Fabrication Default) UART boot settings (Tx P0_4 and Rx P0_5)

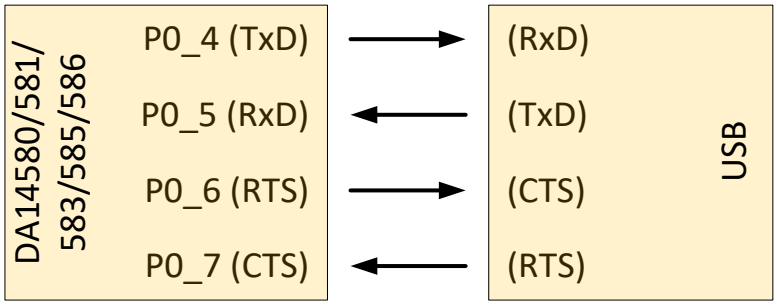


Figure 38: Data direction of UART within J4

For DA14585, data flash through SPI connectivity is enabled with the jumper configuration which is presented in Figure 39. Please notice that for DA14586, this configuration is meaningless and it must be avoided. DA14586 SoC incorporates the flash internally.

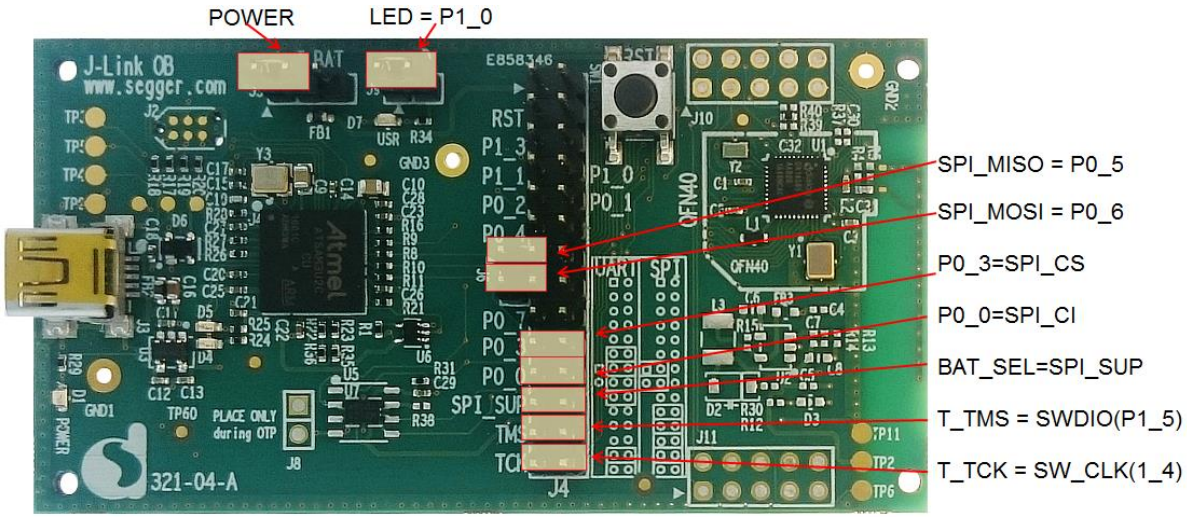


Figure 39: DA1458x Boot from external SPI memory. It is valid for DA14580/1/5

Power configuration: DA14585/6 SoC is configured only in Buck mode on this development kit. DA14585/6 system can be supplied either from 3V3 (via USB) (J5 1-2) or Vdd (via a coin cell) (J5 2-3). The battery (coin cell) is placed in the battery-socket on the solder side of the PCB. No battery is needed when running via the USB-mini-cable.

3.10 Block diagram & downloading software

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

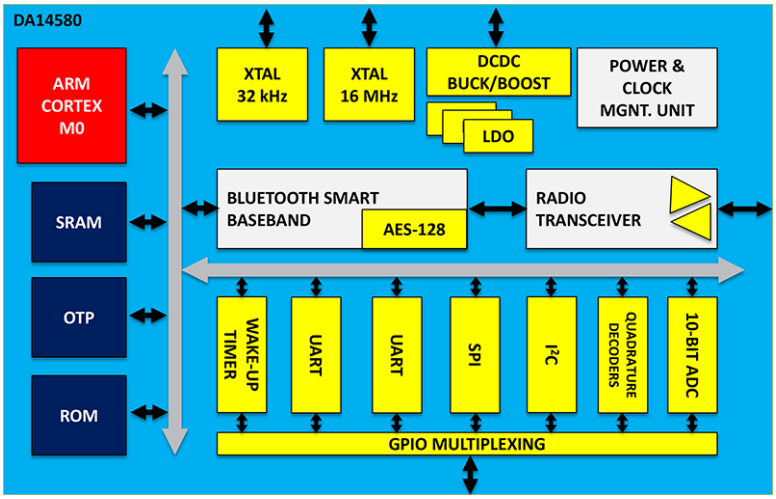


Figure 40: Block diagram with different memory locations

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 -> DA14580/581/583
- PC -> UART -> DA14580/581/583 -> SPI (flash)
- PC -> UART -> DA14580/581/583 -> OTP

For the settings of the jumpers see Figure 29.

The DA14585/DA14586 is equipped with: (on the chip) SRAM (96k) and OTP (64k).

Mounted on the board is external SPI flash memory (2Mbit). External Flash is not used for DA14586.

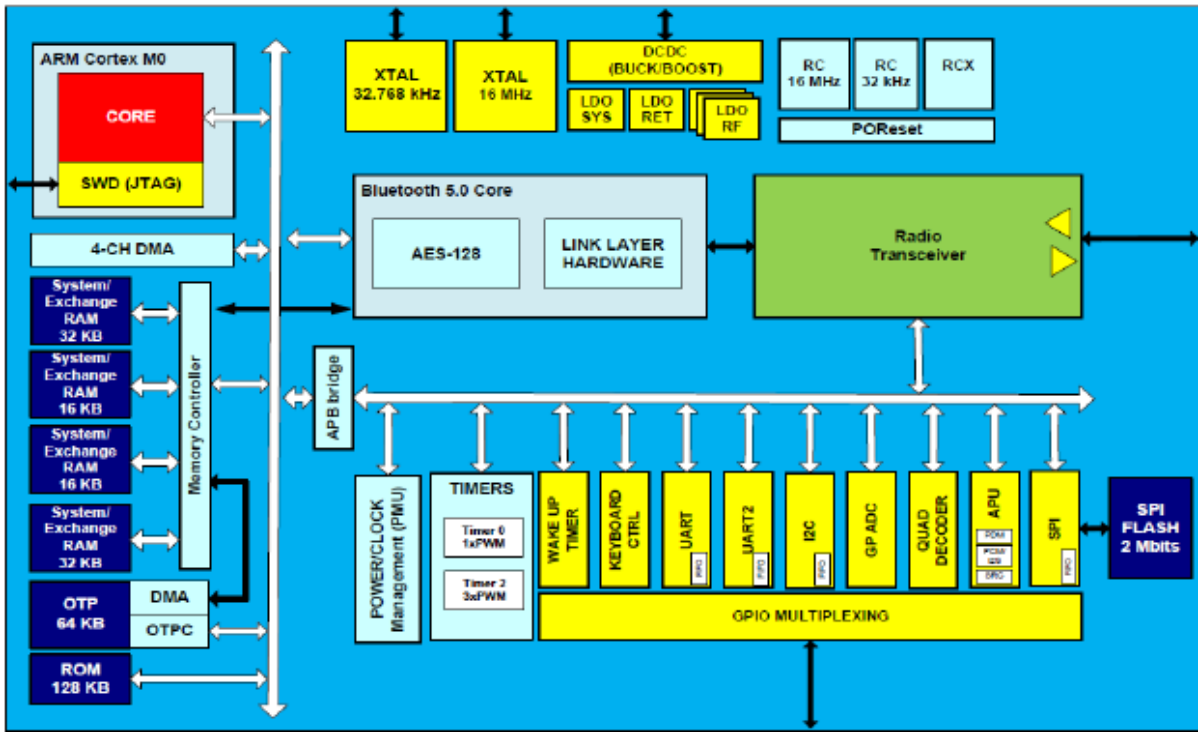


Figure 41: DA14585/586 block diagram

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

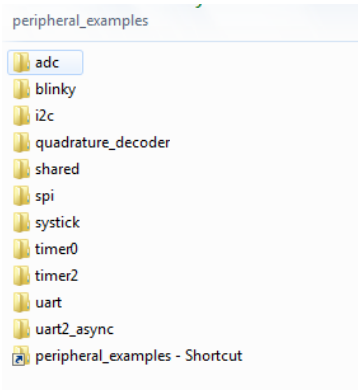
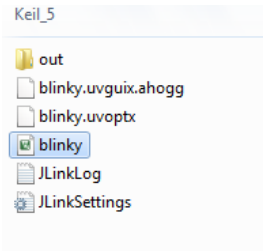
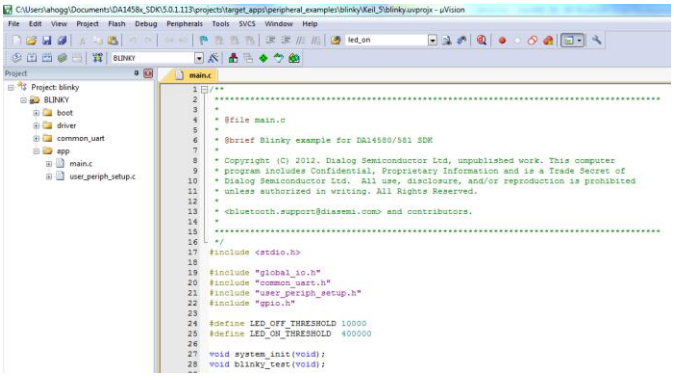
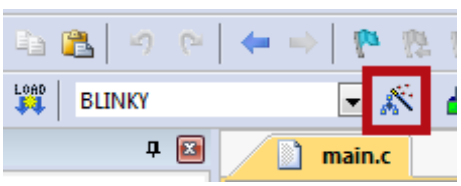
- PC → UART → DA14585/586
- PC → UART → DA14585/586 → SPI (flash)
- PC → UART → DA14585/586 → OTP

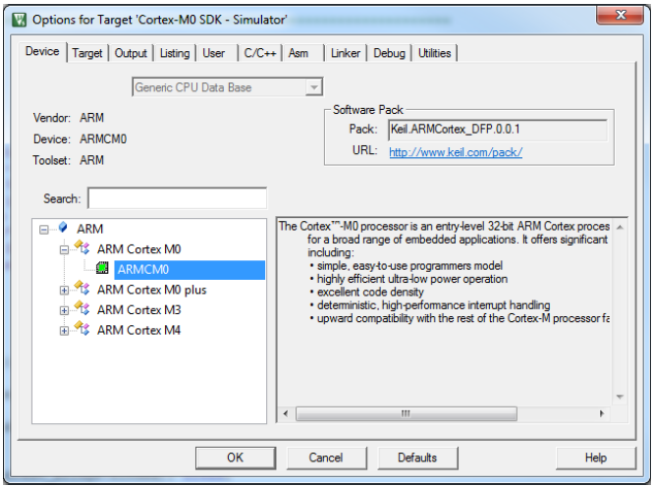
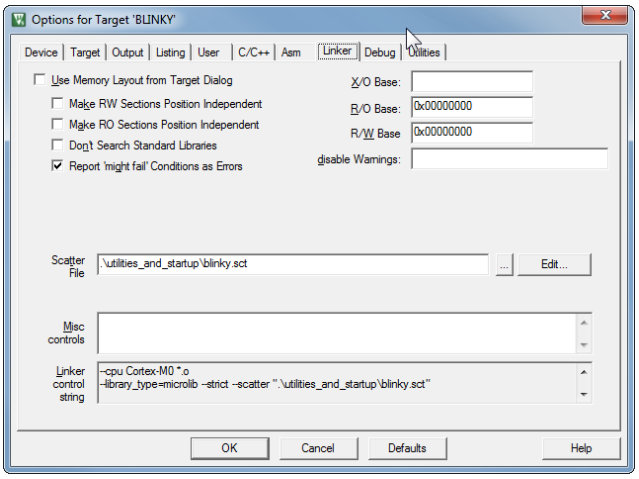
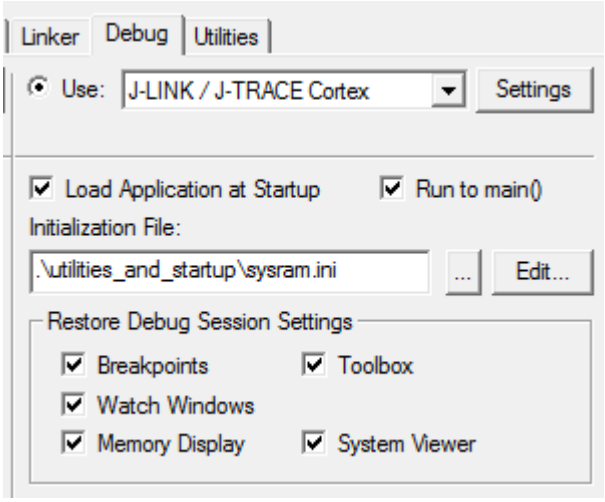
For the settings of the jumpers see Figure 37.

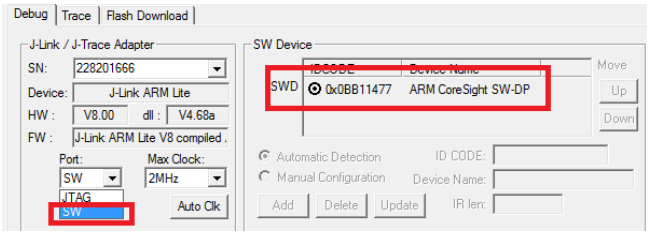
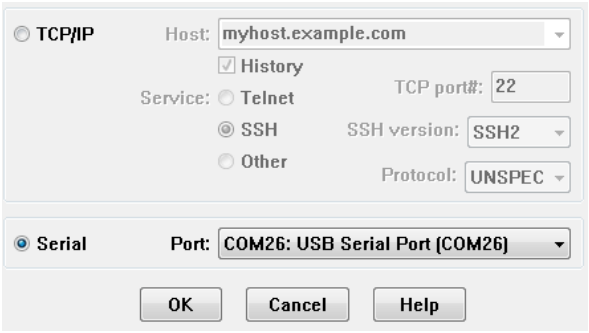
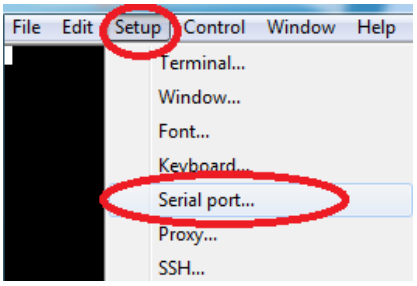
4 Using the demo kit

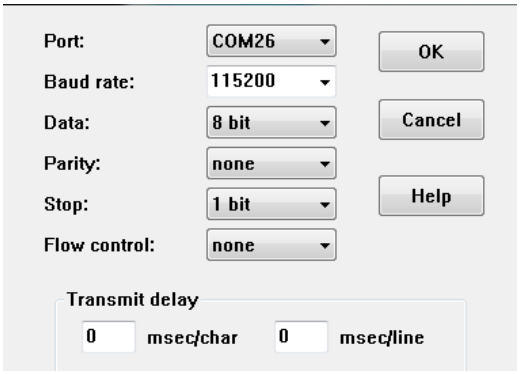
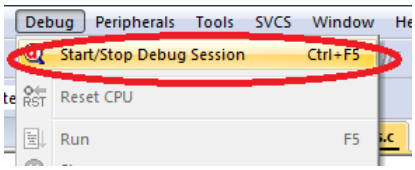
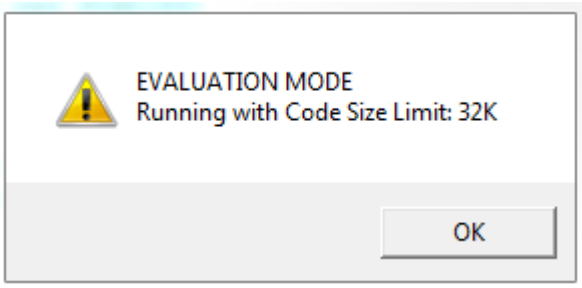
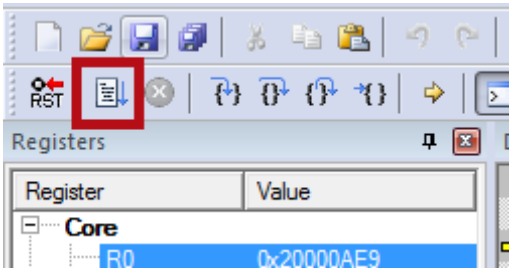
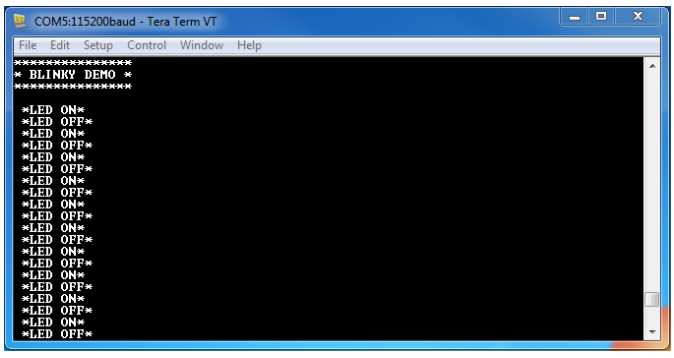
Follow these steps shown in Table 11 to easily create a working demo kit.

Table 11: Run an example on DA14580/581/583/585/586

4.1 Run an example on the DA14580/581/583/585/586		
1	After you download the SDK at www.dialog-semiconductor.com/support The source code example can be found in the example directory called "*peripheral_examples*\". Go to projects\target_apps\peripheral_examples\blinky\Keil_5 Double click "blinky.**uvproj**	 
2	The development environment should look like this when the project is opened with Keil.	
3	Click on the "Target Options" button	

4	‘Options for Target’ → ‘Device’ -screen should look like this.	
5	Scatterfiles (.sct) are used for selecting memory areas.	 scatter file selection in ‘Options for Target’
6	Make sure “J-LINK/J-Trace Cortex” is selected as shown and the initialization file field is set correctly to “.\sysram.ini”. Click on “**Settings**” for next screen.	

7	After clicking the “**Settings**”-button above, make sure the SW Device has been detected correctly.	
8	Click “**OK**” to save the settings.	All settings have been saved properly now, and you can continue to build the example.
9	Build the project by pressing “F7” key, or click the build button as shown in following picture	
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)
11	Open the “**Tera Term**” serial terminal on you PC.	
12	Open Tera Term and choose a COM port, which you have found in step 3, and click OK	
13	Choose Setup->Serial port to configure the Baud rate etc.	

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.	
15	Go back to Keil Project. In the menu bar, select Debug->Start/Stop Debug Session.	
16	A dialog window pops up, like the one on the right. Please click “OK”	
17	Press F5 key or click execution button as shown in following picture, to start code execution.	
18	Then you can see a *blinky* message on your UART terminal screen. That means you have successfully programmed and started the blinky program on DA14580/581 as well as DA14585/586 Basic development board.	

5 Power Management: measuring current

The design of this DA14580 and DA14581 DEVKT –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 42. Shown are the connections of the jumpers J4, J5 and J6. For extra info see the electrical schematic in section 3.8.

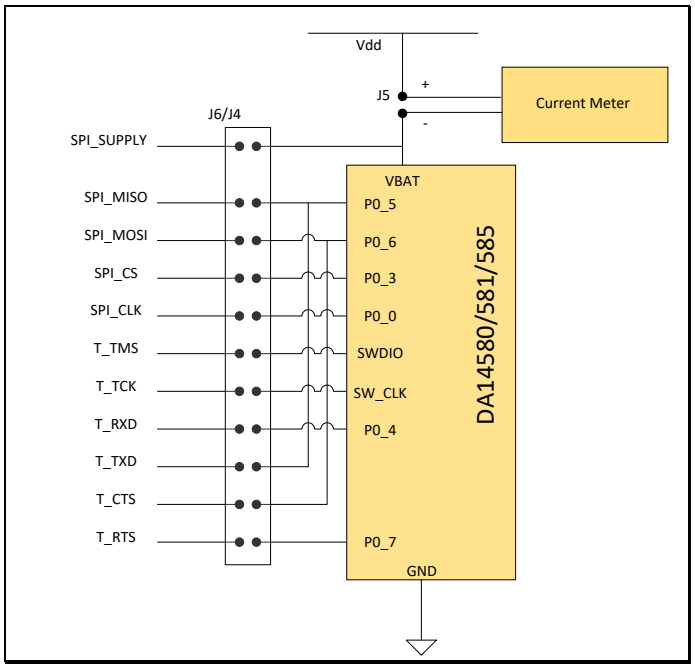


Figure 42: Setup of DA14580/581/585 DEVKT – BASIC during power measurement

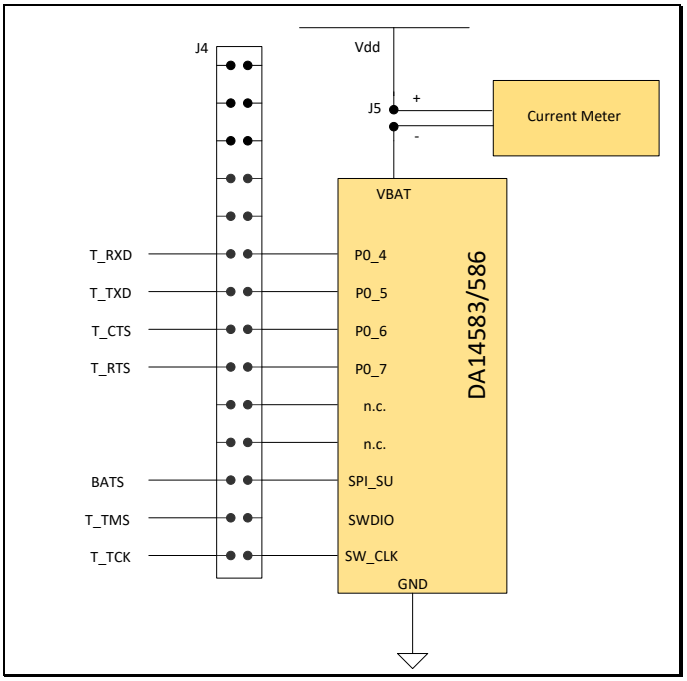


Figure 43: Setup of DA14583/586 DEVKT – BASIC during power measurement

Steps how to do the power measurements:

1. Connect the Current Meter to jumper J5.
2. Mount the jumpers needed for downloading the software.
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 DA14580/581/583/585/586 pins are isolated and only the current meter and GND are connected.
7. Read the current.

For additional info: see AN-B-015 DA14580/581 Supply current measurements. [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 DA14580/581

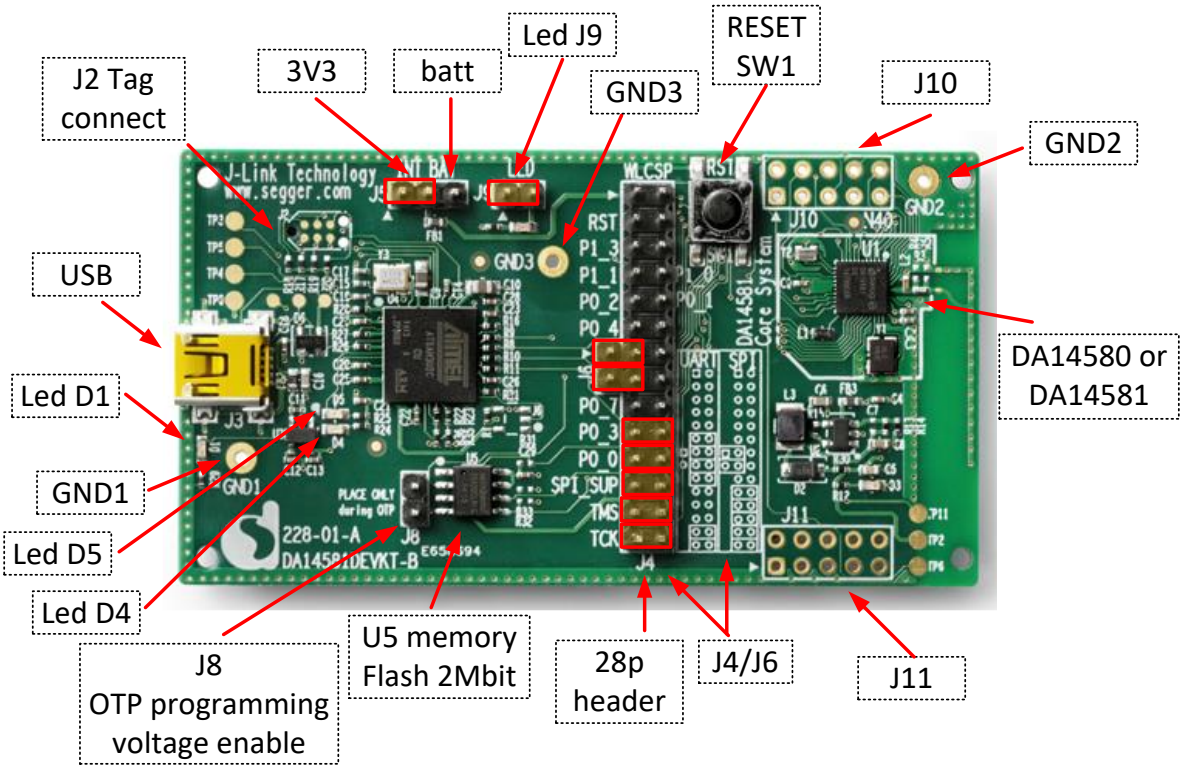


Figure 44: A general overview of the DA14581 development board

Appendix B Connections of J10 and J11

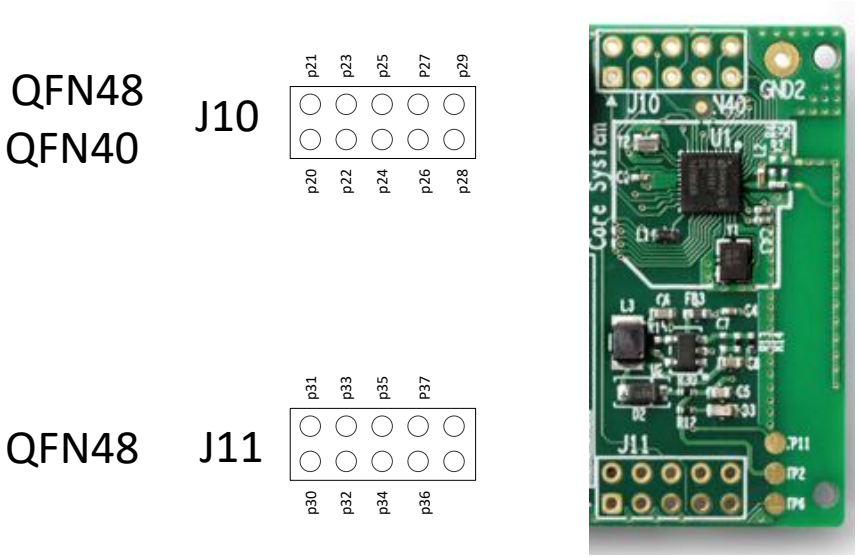


Figure 45: DA14580/581 Connection diagram of J10 and J11

Note 1 The QFN48 package is not available for the DA14581.

Appendix C Layout DA14583

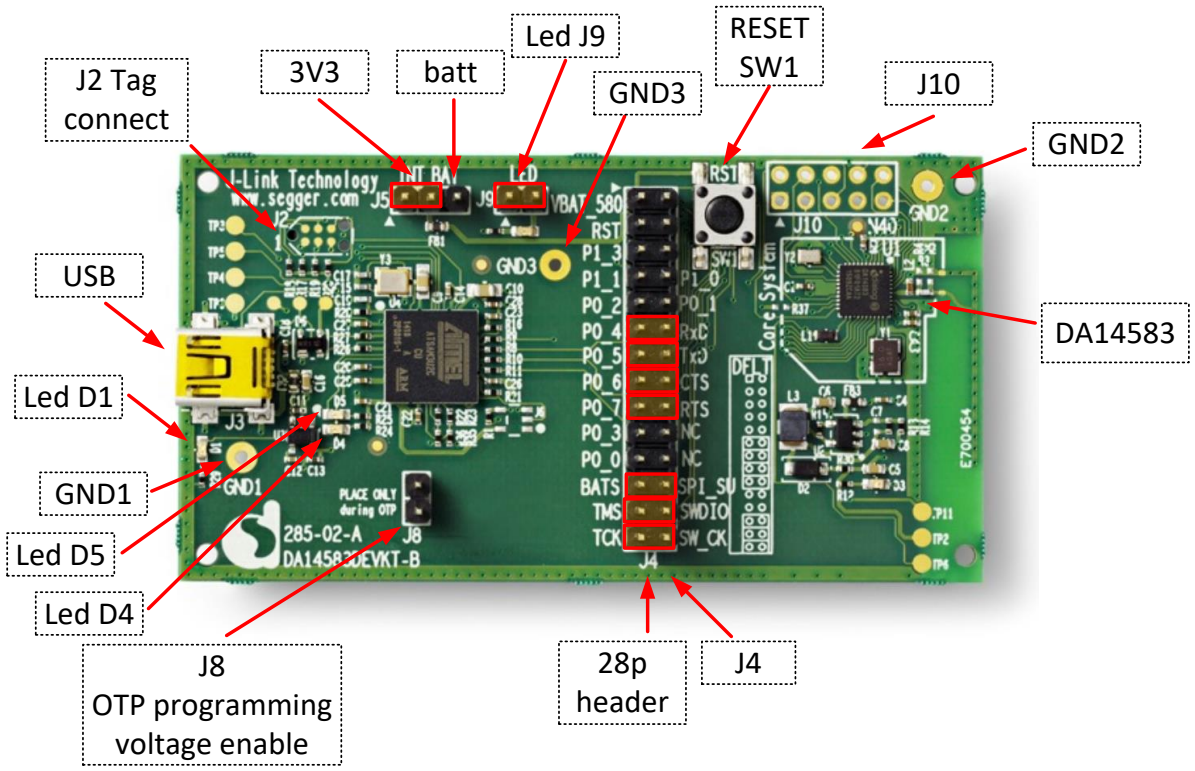


Figure 46: A general overview of the DA14583 development board

Appendix D Connections of J10

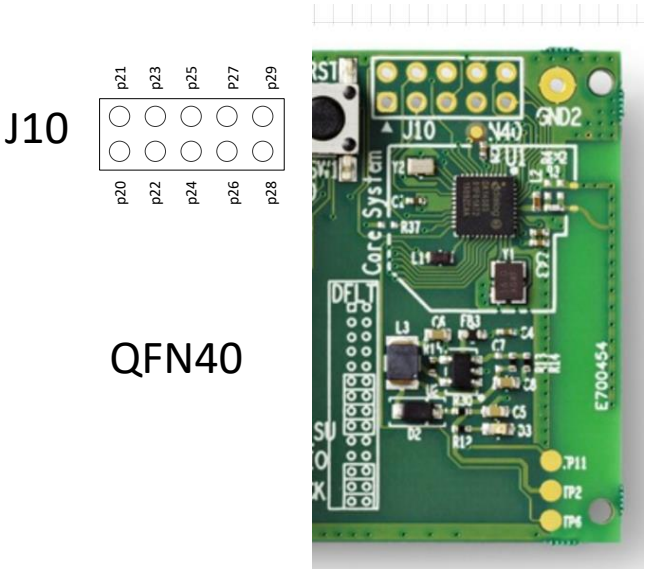


Figure 47: DA14583 Connection diagram of J10

Appendix E Using the smart snippets CLI

All the information/syntaxes about the CLI can be found under the **HELP** tab in the SmartSnippets GUI or by writing **SmartSnippets –help** in the CLI. In this example, it is assumed 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 that has been loaded. On the one hand, if the commands are going to be sent via UART, the following binary file must be used:

- `flash_programmer.bin`

On the other hand, if the commands are going to be sent via JTAG, the following binary file must be used:

- `jtag_programmer.bin`

Note 2 The files can be found in the SmartSnippets resources folder and must be to be downloaded into the DA14580.

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'**:

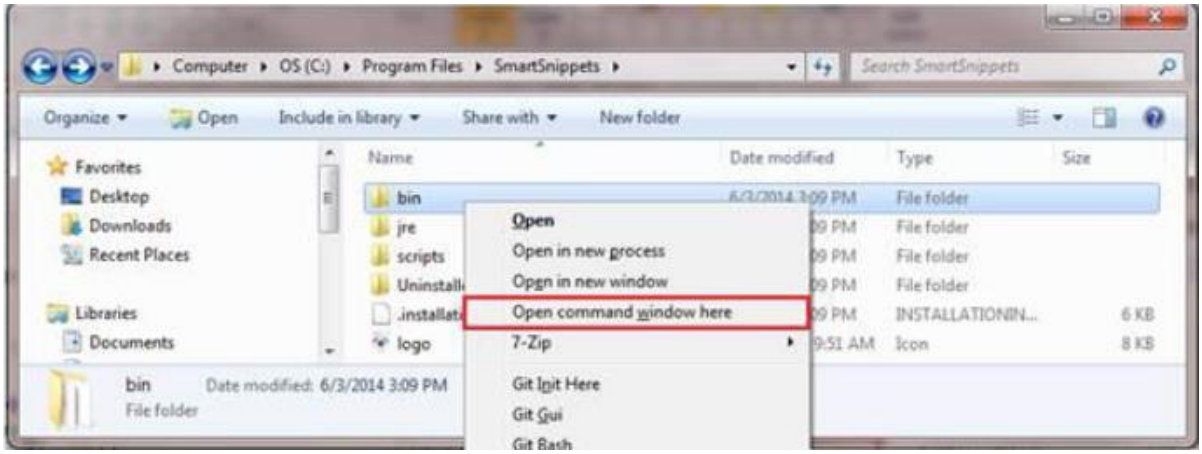


Figure 48: Open the CLI of SmartSnippets

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 output should be:

```
Found SWD-DP with ID 0x0BB11477
FPUnit: 4 code (BP) slots and 0 literal slots
Found Cortex-M0 r0p0, Little endian.
BILE device DA14580 selected.
Using default GPIO pin Id: P1_2.
Using default baudrate: 57600 Bd.
Burned 2 bytes to address 0x00093.
```

Figure 49: SmartSnippets output

Appendix F 'Blinky' Test software

```
/**
****************************************************************************************
 *
 * @file main.c
 *
 * @brief Blinky example for DA14580/581 SDK
 *
 * Copyright (C) 2012. Dialog Semiconductor Ltd, unpublished work. This computer
 * program includes Confidential, Proprietary Information and is a Trade Secret of
 * Dialog Semiconductor Ltd.  All use, disclosure, and/or reproduction is prohibited
 * unless authorized in writing. All Rights Reserved.
 *
 * <bluetooth.support@diasemi.com> and contributors.
 * ****************************************************************************************
 */
#include <stdio.h>
#include "global_io.h"
#include "common_uart.h"
#include "user_periph_setup.h"
#include "gpio.h"

#define LED_OFF_THRESHOLD 10000
#define LED_ON_THRESHOLD  400000

void system_init(void);
void blinky_test(void);
/**
****************************************************************************************
 * @brief  Main routine of the UART example
 *
 ****************************************************************************************
 */
int main (void)
{
    system_init();
    periph_init();
    blinky_test();
    while(1);
}

/**
****************************************************************************************
 * @brief System Initiialization
 *
 *
 ****************************************************************************************
 */
void system_init(void)
{
    SetWord16(CLK_AMBA_REG, 0x00);                  // set clocks (hclk and pclk ) 16MHz
    SetWord16(SET_FREEZE_REG,FRZ_WDOG);             // stop watch dog
    SetBits16(SYS_CTRL_REG,PAD_LATCH_EN,1);         // open pads
    SetBits16(SYS_CTRL_REG,DEBUGGER_ENABLE,1);      // open debugger
    SetBits16(PMU_CTRL_REG, PERIPH_SLEEP,0);        // exit peripheral power down
}

/**
 ****************************************************************************************
 * @brief Blinky test fucntion
 *
 *
 ****************************************************************************************
 */

void blinky_test(void)
{
    int i=0;
    // Select function of the port P1.0 to pilot the LED
    printf_string("\n\r\n\r");
    printf_string("***************\n\r");
```



```
    printf_string("* BLINKY DEMO *\n\r");
    printf_string("***************\n\r");

    while(1)
{
        i++;
        if (LED_OFF_THRESHOLD   == i) {
            GPIO_SetActive( LED_PORT, LED_PIN);
            printf_string("\n\r *LED ON* ");
        }
        if (LED_ON_THRESHOLD == i) {
            GPIO_SetInactive(LED_PORT, LED_PIN);
            printf_string("\n\r *LED OFF* ");
        }
    if (i== 2*LED_ON_THRESHOLD){
    i=0;
                          }
    }
}
```

Appendix G Latency Timer of FTDI cable

If an external FTDI cable is used to burn the OTP (or to download the image into the external memory), the Latency Timer of the FTDI cable has to be changed from 15ms to <10ms.
To change the Latency Timer:
Device Manager → COM port → Right click on the COM port chosen → Properties → Port Settings → Advanced → Latency Timer: set it <10ms.

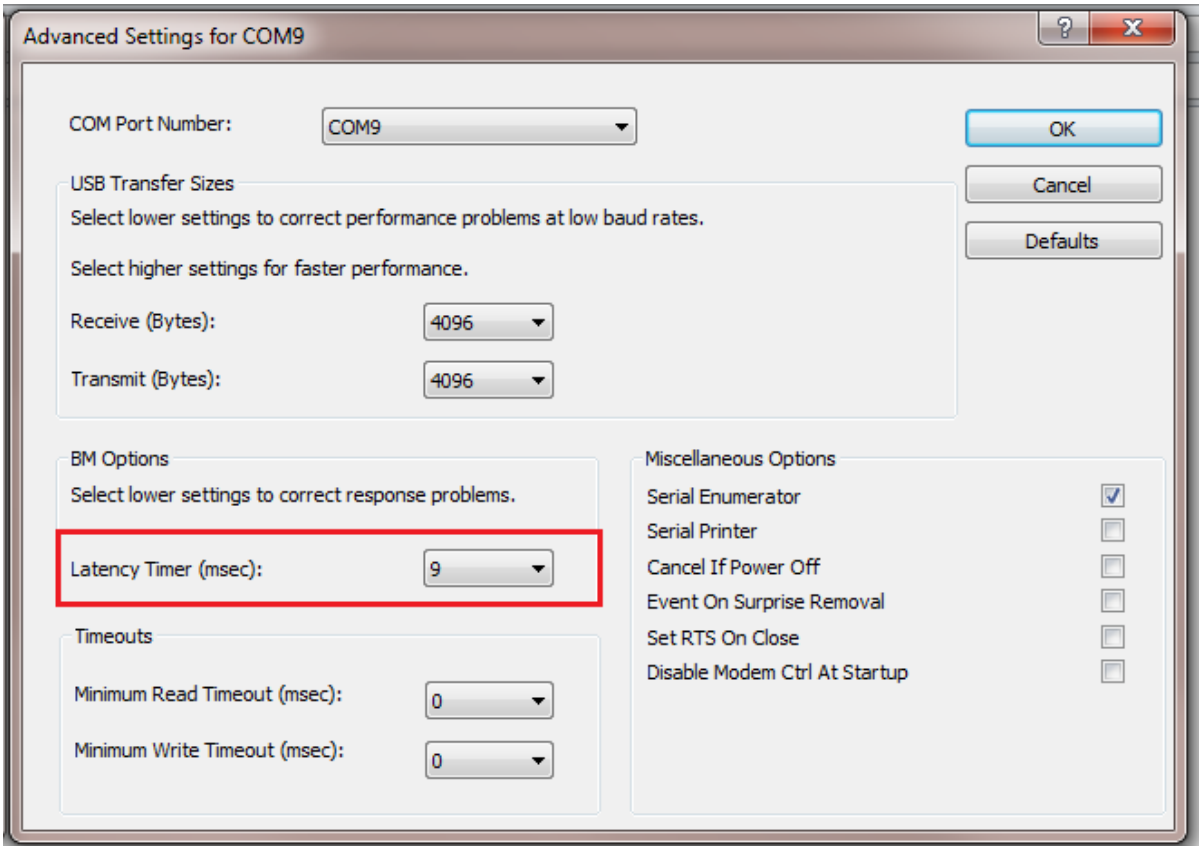


Figure 50: FTDI Latency Timer

Appendix H DA1458xDEVKT-B hardware differences to DA14580, DA14581 and DA14583 hardware versions

Table 12: DA1458xDEVKT-B differences to previous hardware versions

Differences	DA1458xDEVKT-B	Previous hardware versions: DA1580DEVKT-B DA14581DEVKT-B (228-01-A) DA14583DEVKT-B (285-02-A)	Comments
U7, Flash Memory	MX25R2035, 2Mbits	W25X20CL (1)	Provides better power consumption
U6, NC7SZ04P5X	Mounted	Not mounted	Provides the capability to software reset DA1458x though debugger chip
VPP circuit	enabled or disabled by: 1. Not mounting the whole VPP generation circuitry ***or*** 2. by mounting or not resistor R37	Always enabled	DA14585/6 doesn't need external 6.8V voltage rail (VPP).

(1) Flash is not existed on DA14583DEVKT-B

Appendix I DA1458xDEVKT-B BOM modifications for supporting DA1458x SoC family

DA1458xDEVKT-B with PCB number 321-04-A has been designed to support the whole DA1458x family with package QFN40. (DA14580, DA14581, DA14583, DA14585, DA14586). User can indicate which DA1458x SoC is used by checking the labels assembled on the board (PCBA). By minor modifications on Bill of Materials (BOM), the options required for each chip are enabled. The top-screen layer of the DA1458x Basic Kit PCB is shown in Figure 19

Table 13: DA1458x BOM modifications for enabling DA1458x family chips

	Configuration				
U1, (BLE processor)	R37 (1) (connected to VPP)	R39 (connected to P3_0)	R40 (connected to P3_0)	C32 (connected to P3_0)	Flash type
DA14580	0 Ohm	Not Mount	Not Mount	Not Mount	
DA14581	0 Ohm	Not Mount	Not Mount	Not Mount	
DA14583	0 Ohm	Not Mount	0 Ohm	100 nF	Not Mount
DA14585	Not Mount	0 Ohm	Not Mount	Not Mount	
DA14586	Not Mount	Not Mount	0 Ohm	100 nF	Not Mount

(1) VPP circuitry is not needed to be populated for DA14585/6

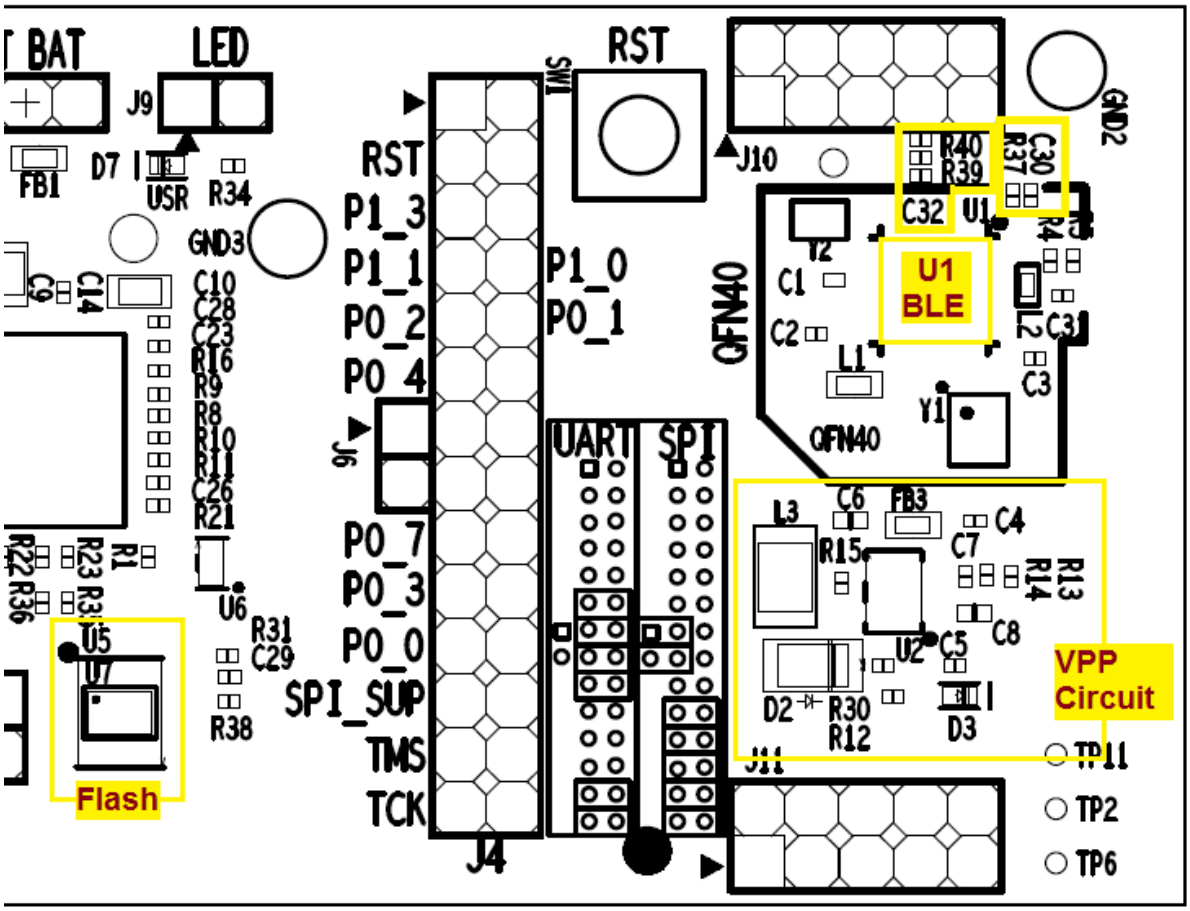


Figure 51: Location of the components that enables various versions of DA1458x family

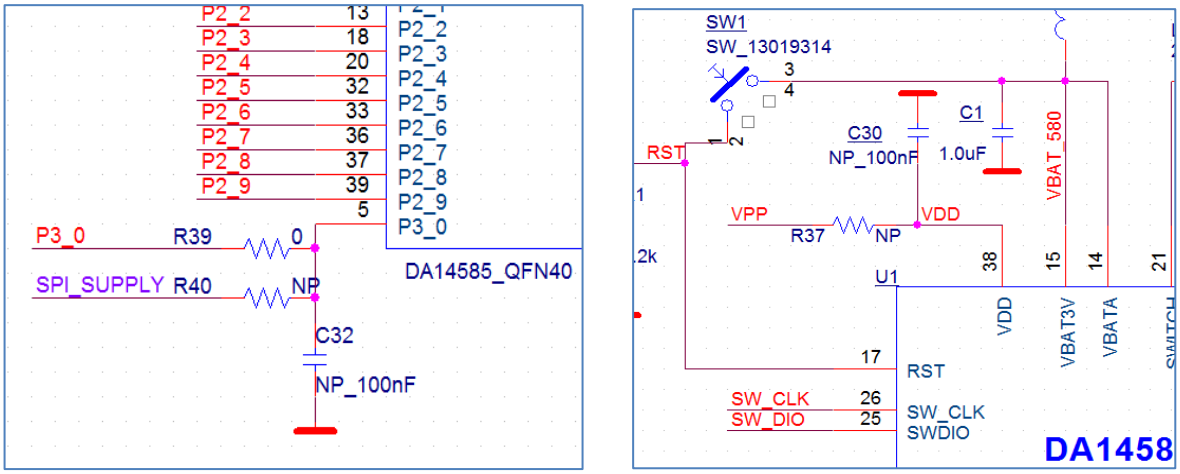


Figure 52: Configuration components for enabling the various versions of DA1458x family.

6 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

Revision history

Revision	Date	Description
1.3	26-01-2022	Update logo, disclaimer, copyright.
1.2	22-06-2017	Update for DA14585/586 and SDK 6
1.1	20-10-2015	Minor updates in Keil Installation instructions.
1.0	27-08-2015	Initial version for DA1458x family with SDK 5.

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.

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.