# 標註摘要

由 *kasoarcat* 在 3 頁上做的 3 條註解

x3

| DAUGHTERBOARD | | |
|---|---|---|
| Signal Name | FTDI Pin Number | DA14580 GPIO |
| UTX | 17 | P0_4 |
| URX | 16 | P0_5 |
| UCTS | 18 | P0_6 |



#1 p.20

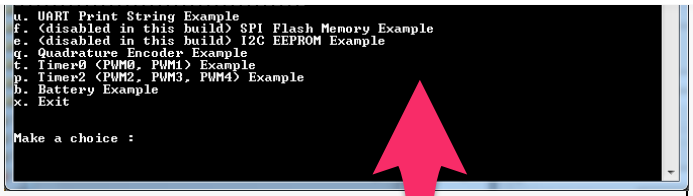
## 4.2 Installation of the tools and drivers

To install the Software development environment please follow the steps below.

### 4.2.1 Keil

| | | |
|---|---|---|
| | Download and install Keil MDK | Keil: https://www.keil.com/download/product/ |

#2 p.21

| | | |
|---|---|---|
| | 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. |   |

#3 p.29

# User manual

## DA14580 Bluetooth® Smart Development Kit - Expert

## UM-B-014


### Abstract

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

# Contents



# Figures



# Tables



# 1 Terms and definitions

| | |
|---|---|
| BLE | Bluetooth Low Energy |
| CS | Chip Select |
| DK | Development Kit |
| EEPROM | Electrically Erasable Programmable Memory |
| FTDI | Quad-Flat No-leads |
| GPIO | General Purpose Input Output |
| QFN | Quad-Flat No-leads |
| OTP | One Time Programmable |
| SDK | Software Development Kit |
| SPI | Serial Peripheral Interface |
| SRAM | Static Random Access Memory |
| USB | Universal Serial Bus |
| UTX | Quad-Flat No-leads |
| URX | Quad-Flat No-leads |
| URTS | Quad-Flat No-leads |
| UCTS | Quad-Flat No-leads |
| UCTS | Quad-Flat No-leads |
| UART | Universal Asynchronous Receiver/Transceiver |
| WLCSP | Wafer Level Chip Scale Packaging |

# 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

# 3 Introduction

DA14580 is a Bluetooth Smart (low energy) 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 - expert includes a set of hardware (e.g. development boards and 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 customers to set up hardware/software development environment, install required software and quickly start product development based with help of example source code.

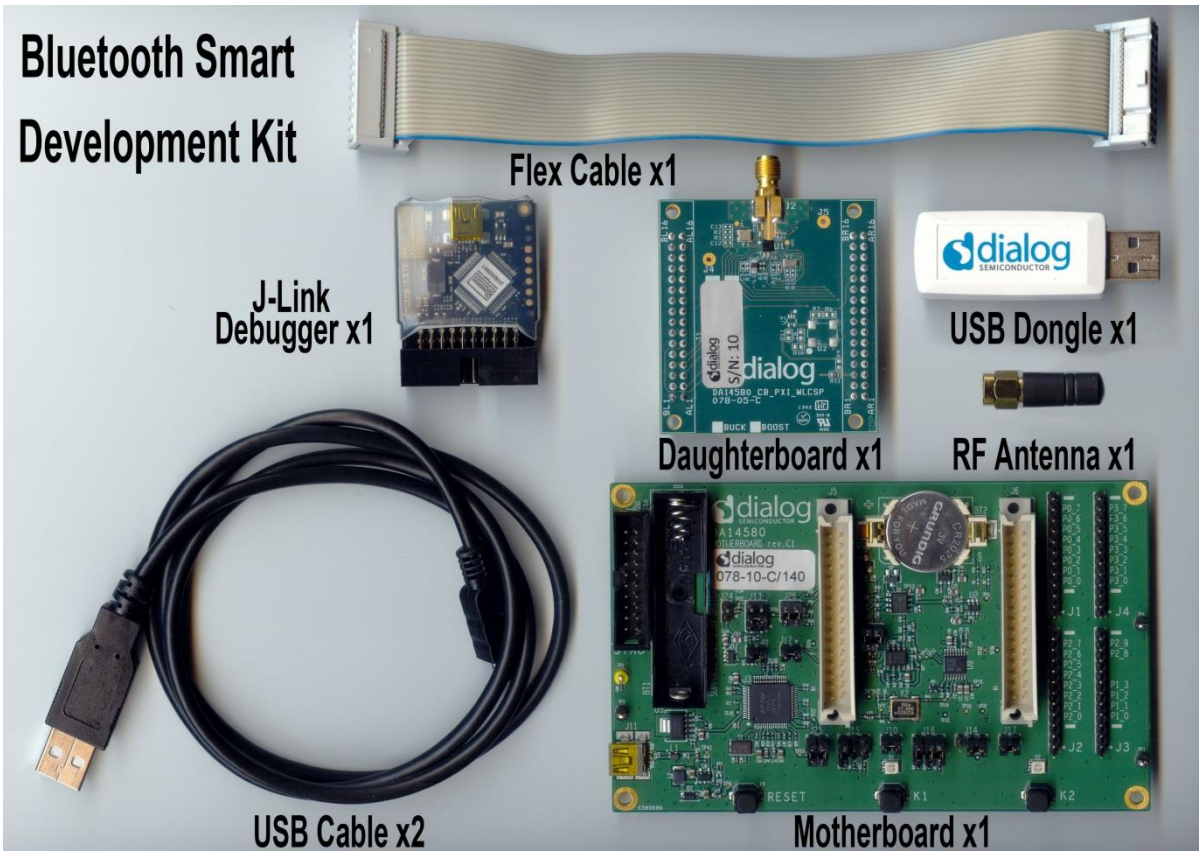
Web content can be downloaded at support.diasemi.com

Product information about DA14580 can be found at:

http://www.dialog-semiconductor.com/products/bluetooth-smart

## 3.1 Hardware content

The following picture presents the building blocks of the various the DA14580 DK Kits.




**Figur1: DA14580 Development Kit**

These are:

- Kit Components
  - DA14580 Motherboard
  - DA14580 Daughterboard WLCSP or DA14580 Daughterboard QFN40 or DA14580 Daughterboard QFN48
  - DA14580 USB Dongle

- Kit Peripherals
  - Segger Jlink Debugger
  - Flex Cable
  - USB Cable
  - RF Antenna
  - Battery – Type AAA
  - Battery – Coin Type

The aforementioned material is combined to provide the following products:

**Table 1: Default Development Kit components**

| | KIT COMPONENTS | | | | | KIT PERIPHERALS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **KITS** | **MB** | **DB WLCSP** | **DB QFN40** | **DB QFN48** | **USB DONGLE** | **Segger JLink** | **Flex Cable** | **USB Cable** | **RF Antenna** | **Battery AAA** | **Battery Coin** |
| **MAIN KIT** | 1 | - | - | - | 1 | 1 | 1 | 2 | 1 | 1 | 1 |
| **WLCSP KIT** | - | 3 | - | - | - | - | - | - | - | - | - |
| **QFN40 KIT** | - | - | 3 | - | - | - | - | - | - | - | - |
| **QFN48 KIT** | - | - | - | 3 | - | - | - | - | - | - | - |

## 3.2 Web content

### 3.2.1 Software Development Kit content

#### 3.2.1.1 Tools

**Smart Snippets** (a framework of PC based tools to control DA14580 development kit), consisting of

- Power Profiler : Real time current consumption measurement to for the DA14580 motherboard
- OTP Programmer: Tool for OTP memory programming
- UART/JTAG booter : Tool for downloading hex files to DA14580 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

**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 Bluetooth Smart Development Kit - Expert
- UM-B-015, DA14580 Software architecture
- UM-B-016, DA14580 Software Porting Guide
- UM-B-017, DA14580 GTL interface Integrated Processor Application

#### 3.2.1.3 SDK source code examples (created in Keil)

- **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) |
| **monitor_fe_usb** | fe_usb_proxm_sdk.uvproj | Proximity Monitor (External processor)<br>Version for USB dongle |
| **reporter_fe_usb** | fe_usb_proxr.uvproj | Proximity Reporter (External processor)<br>Version for USB dongle |
| **Reporter_fe_spi** | fe_proxr_spi.uvproj | Proximity Reporter (External processor) SPI version |

  - **dk_apps\keil_projects\prod_test:** These folders include the source code of the production test firmware. Refer to DA14580_Production_Test_Tool.docx for more information how to build and use it.
  - **dk_apps\keil_projects\template:** These folders include a template as a starting point of a new application.

For details, please read [9].

- **host_apps:** This folder holds the DA14580 PC applications:
  - ***host_apps\windows\proximity:*** The 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:** The folder includes two pre-compiled Windows executables which correspond to the C++ applications described right above and are included for user convenience.
  - **peripheral_examples:** The folder includes sample code of how to use peripheral blocks of the DA14580 (e.g. UART, SPI, I2C etc.) bundled to a demo-kit. For details, please refer to [10].
- Tools

- **tools\prod_test\prod_test_cmds:** This folder includes the source code of the production test tool. Refer to DA14580_Production_Test_Tool.docx for more information how to build and use it.

### 3.2.2 Schematics and PCB layout

Schematics and PCB layout of the Motherboard, Daughterboard options and USB dongle are available on the customer support portal.

## 3.3 Daughterboard

### 3.3.1 Block diagram

The daughterboard comes in three different types depending on the type of package of the DA14580:

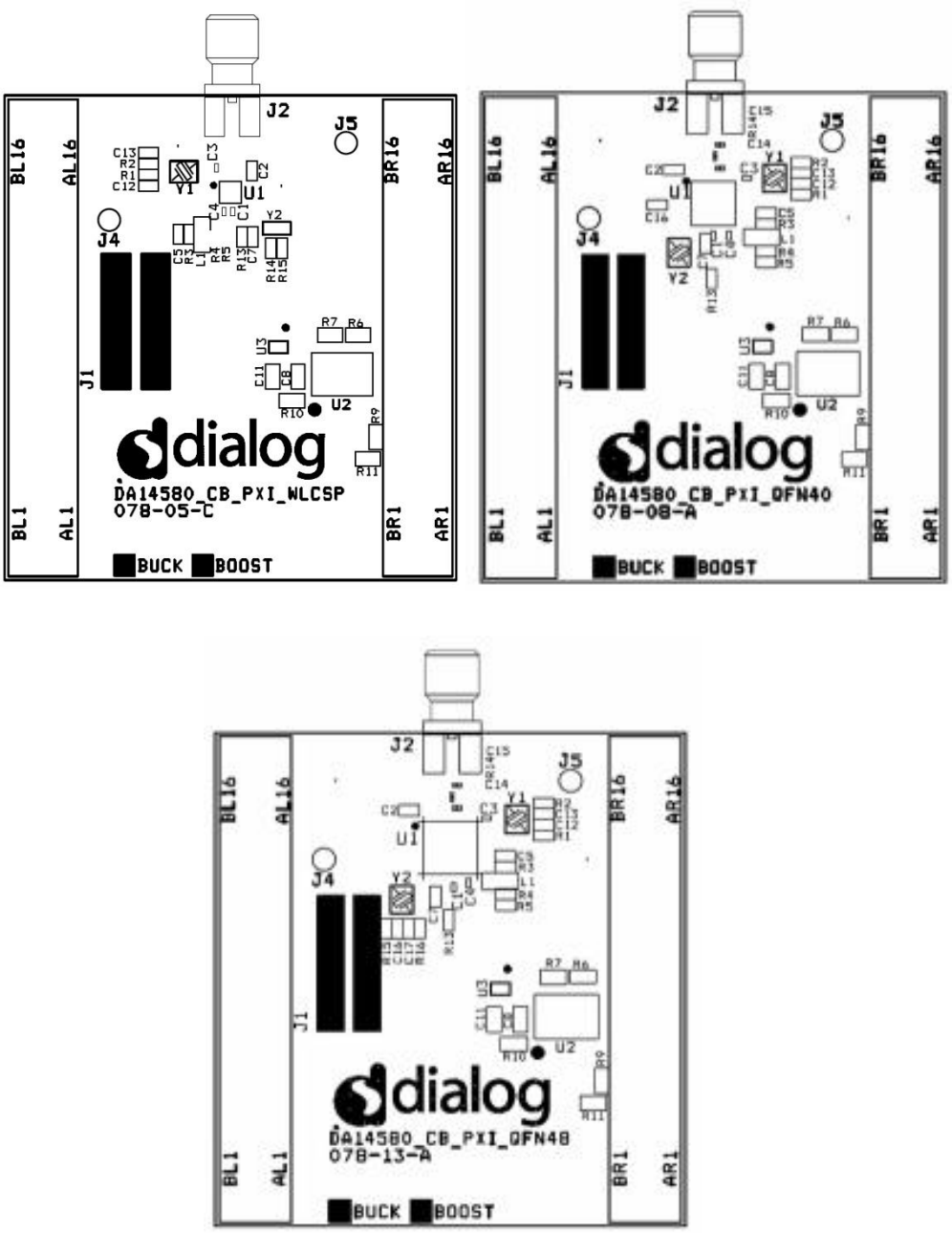



**Figure 1 : Development Kit Daughterboard with WLCSP (top left), QFN40 (top right) and QFN48 (bottom)**

The contents of the three boards are described in the table below:

**Table 3: Development Kit Daughterboard components**

| Name | Description |
|---|---|
| **Connectors** | |
| **J1** | Socket to connect the Daughterboard onto the Motherboard |
| **J2** | SMA connector for the antenna |
| **J4** | Connected to Ground |
| **J5** | Connected to Ground |
| **Units** | |
| **U1** | DA14580 : QFN40 or WLSCP34 or QFN48 package |
| **U2** | CSX-750MB, 32MHz Crystal Oscillator (NOT POPULATED) |
| **U3** | TPS79718DCKR, LDO 10mA, 1.8V (NOT POPULATED) |
| **Test Points** | |
| **TP1** | Connects to P0_0. Used for RF internal measurements |
| **TP2** | Connects to P0_3. Used for RF internal measurements |
| **TP3** | Connects to P1_1. Used for RF internal measurements |
| **TP4** | Connects to P1_2. Used for RF internal measurements |

### 3.3.2 Schematics and layout

The schematics for the PCB are depicted in the following figures:

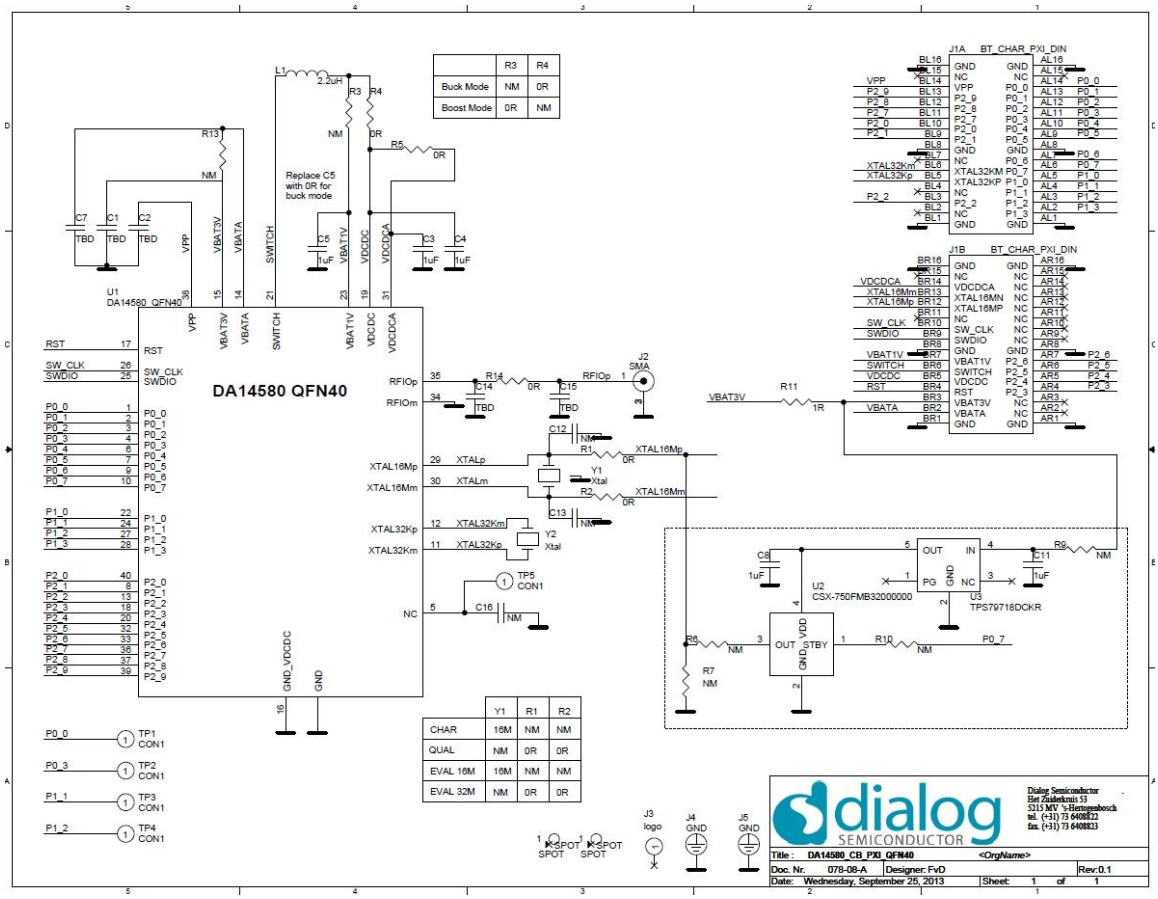


Figure 2 : Schematics of the QFN40 Daughterboard

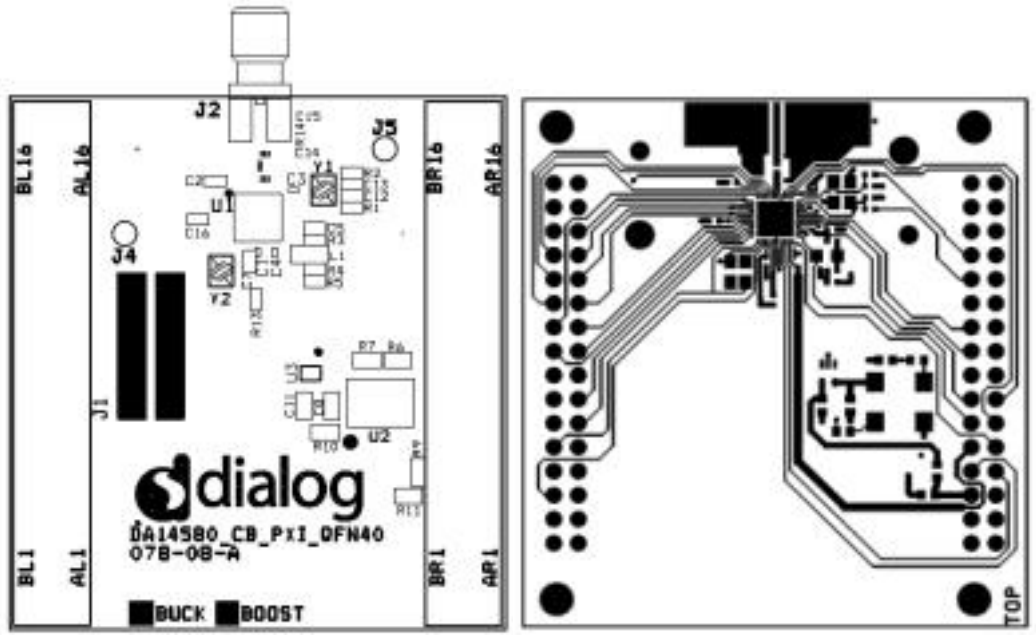



Figure 3: Silkscreen (left) and top (right) layout views of the QFN40 Daughterboard

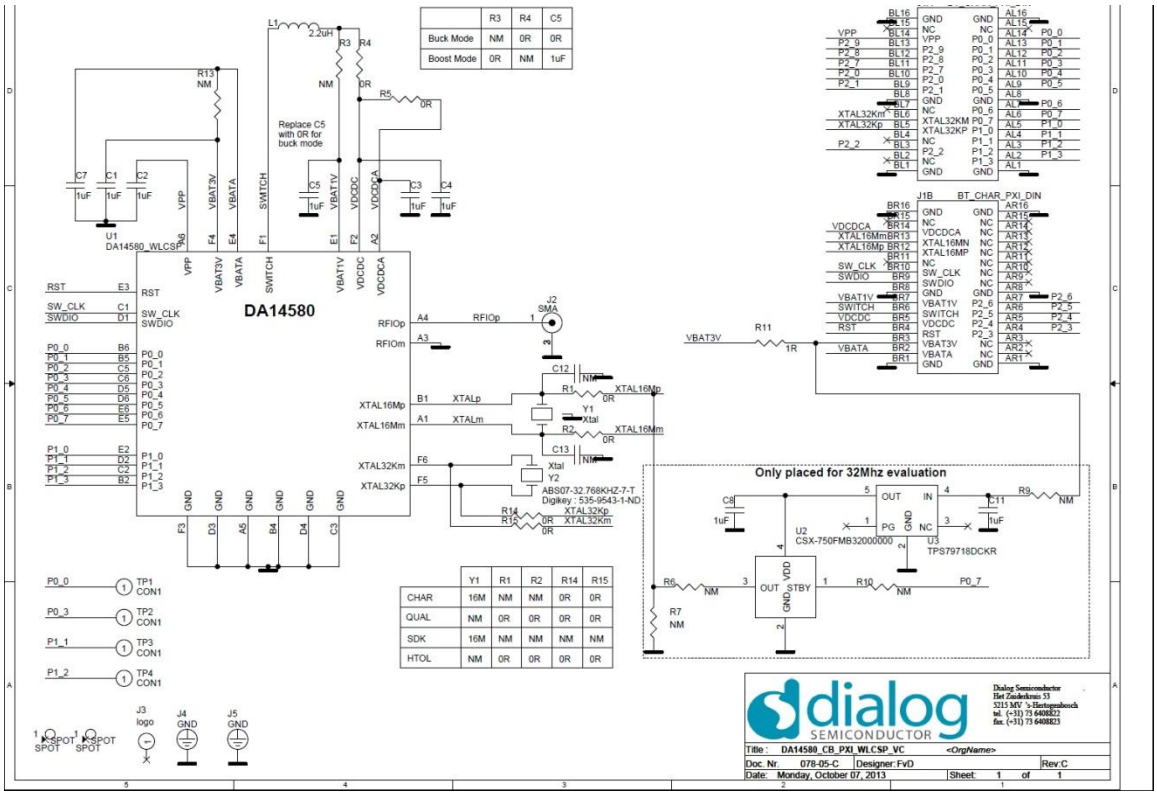



Figure 4: Schematics of the WLCSP Daughterboard

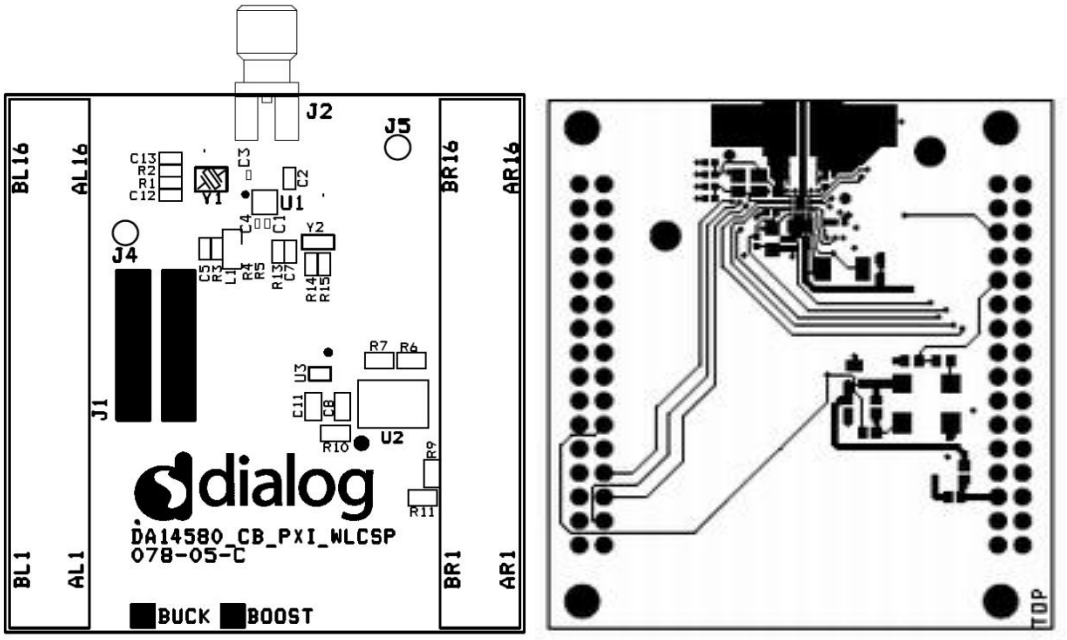



**Figure 5: Silkscreen (left) and top (right) layout views of the WLCSP Daughterboard**

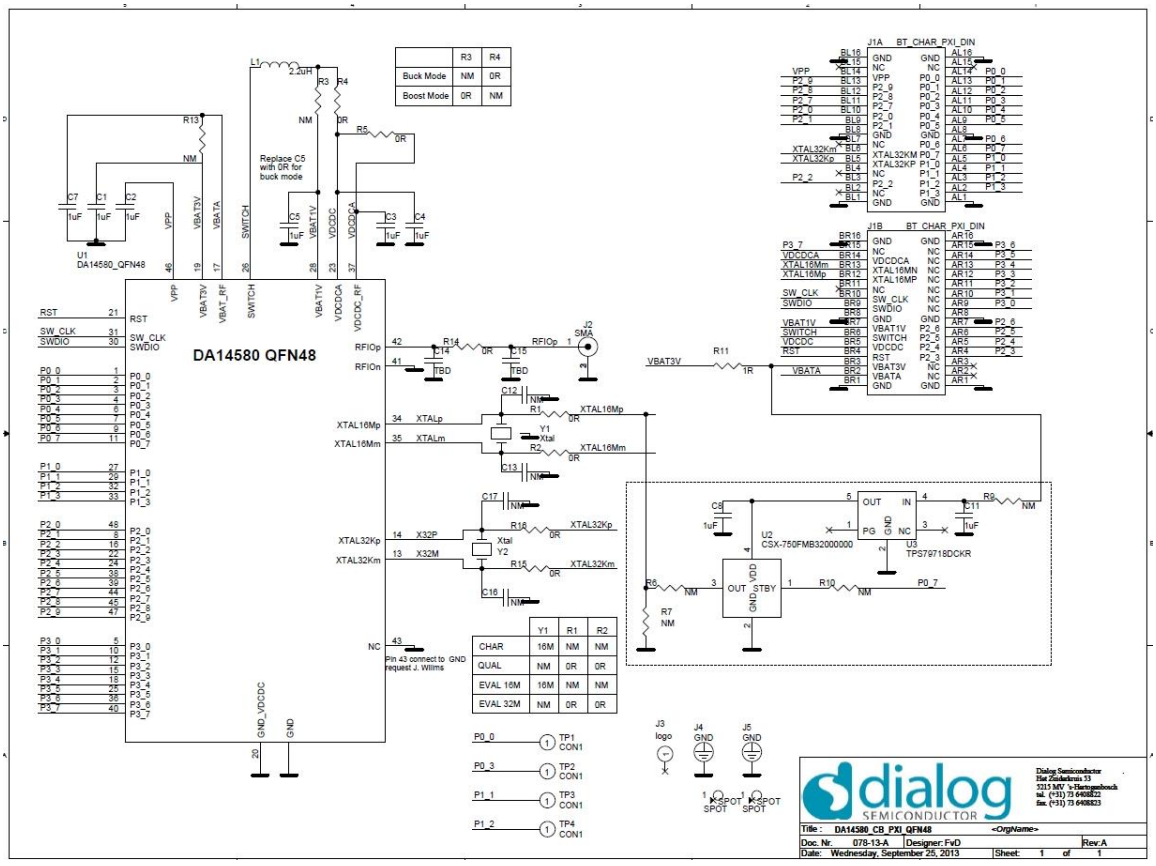



**Figure 6: Schematics of the QFN48 Daughterboard**

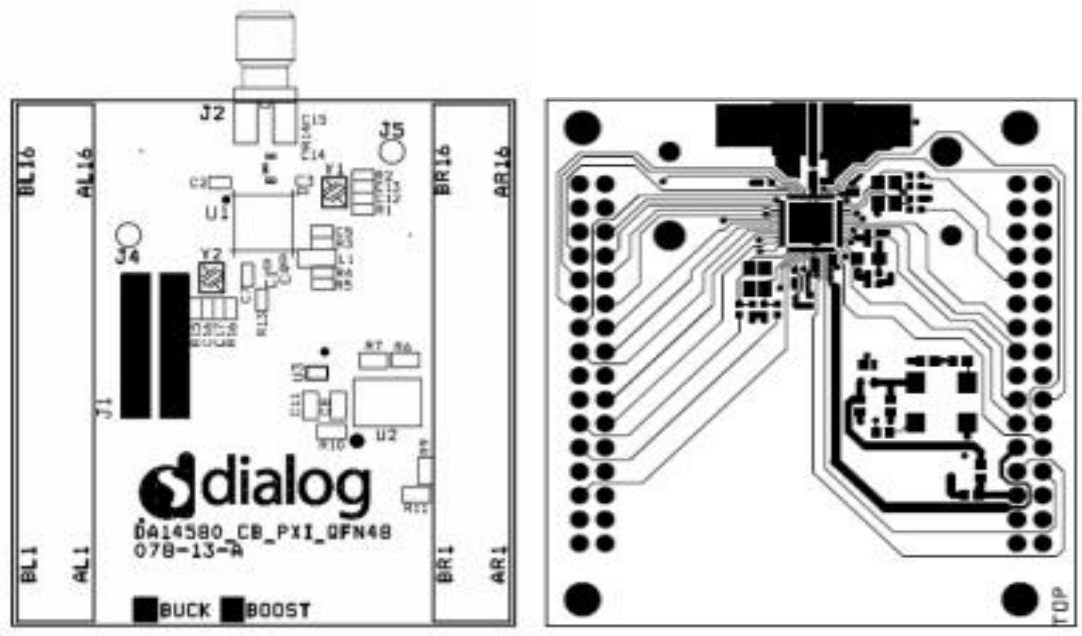



**Figure 7: Silkscreen (left) and top (right) layout views of the QFN48 Daughterboard**

### 3.3.3 Configuring the daughterboard

The daughterboard comes in 3 different flavours according to customer's requirements:

- A daughterboard with a QFN40 package
- A daughterboard with a QFN48 package
- A daughterboard with a WLCSP package

The board is shipped on request pre-configured as either Buck or Boost. Details for modification of the daughterboard to other than the default (shipped) configuration are provided in the *Hardware User Manual*.

## 3.4 Motherboard

### 3.4.1 Block diagram

The block diagram of the motherboard is displayed in the following figure:

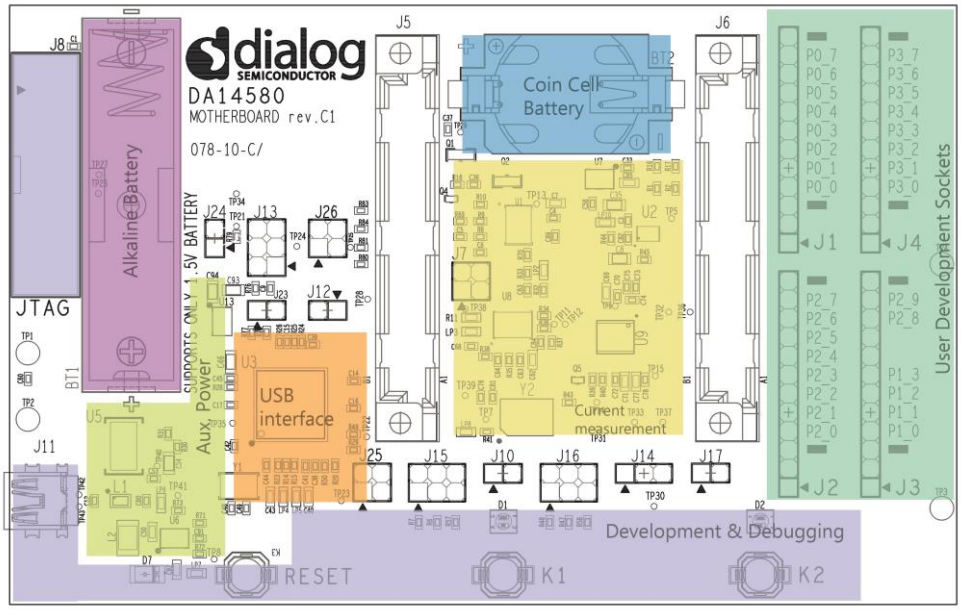
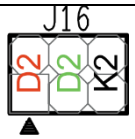



**Figure 8: Block Diagram of the Motherboard (Top View)**

**Table 4: Default Jumper Configuration**

| Name | Color | Description |
|---|---|---|
| **Connectors** | | |
| J1 | Green-Blue | Socket to connect an external board for Bluetooth Smart applications |
| J2 | Green-Blue | Socket to connect an external board for Bluetooth Smart applications |
| J3 | Green-Blue | Socket to connect an external board for Bluetooth Smart applications |
| J4 | Green-Blue | Socket to connect an external board for Bluetooth Smart applications |
| J8 | Lila | JTAG header. Complies to the J-link standard format |
| J5 | White | Socket for the Development Kit, Daughterboard |
| J6 | White | Socket for the Development Kit, Daughterboard |
| J11 | Lila | Mini-USB connector |
| J12 | White | Connects P1_2 pin to the enable gate of the VPP voltage. To be used for programming the OTP. |
| J13 | White | Controls the power source for the power measurement circuit: BT1 (alkaline), BT2(coin cell) or VCC_IN (LDO) |
| J14 | White | Selects power source for the DA14580: either coin or alkaline |
| J15 |  White | Connects:<br>P1_1 with the K1 button.<br>P1_2 with the D1 led (green).<br>P1_3 with the D1 led (red). |
| J16 | White | Connects:<br>P0_6 with the K2 button<br>P0_7 with the D2 led (green). |

| | | |
|---|---|---|
| |  | P1_0 with the D2 led (red). |
| J17 | White | Connects the RESET button to the VBAT power. To be used as a soft reset on the FT2232HL chip. |
| J10 | White | Connects GPIO for measurement trigger |
| J23 | White | Selects 3.0/1.5 Volt output for the LDO. |
| J24 | White | Shorts the 10Ohm resistor at the LDO output |
| J25 | White | Connects UART TX/RX to the default GPIOs |
| J26 | White | Connects UART RTS/CTS to the default GPIOs |
| **Units** | | |
| U1 | Yellow | Current to Voltage Conversion OpAmp |
| U2 | Yellow | OpAmp for the ADC voltage reference |
| U3 | Orange | FT2232HL, Dual Serial to USB bridge |
| U5 | Green | Low Dropout Regulator for the VBUS power |
| U6 | Green | Step-up regulator, generates 6.8 Volts |
| U7 | Yellow | Low noise Low Dropout Regulator to create silent 5V |
| U8 | Yellow | Differential OpAmp for the conversion circuit |
| U9 | Yellow | Analog to Digital Converter |
| **Test Points** | | |
| TP1 | C/V Converter | output suitable for Oscilloscope |
| TP2 | GND | ground point |
| TP3 | GND | ground point |
| TP4 | GND | ground point |
| TP5 | P0_5 | option for crystal calibration |
| TP7 | 3.3V | power supply rail |
| TP8 | +6.8V | power supply rail |
| TP9 | 5.0V | power supply rail |
| TP11 | VINN | output of differential converter |
| TP12 | VINP | output of differential converter |
| TP13 | 5.0VA | power supply rail |
| TP14 | 27MHz | oscillator |
| TP15 | VREF | voltage reference |
| TP16 | Bumpon 3M SJ61A1 | Mechanical Part |
| TP17 | Bumpon 3M SJ61A1 | Mechanical Part |
| TP18 | Bumpon 3M SJ61A1 | Mechanical Part |
| TP19 | Bumpon 3M SJ61A1 | Mechanical Part |
| TP20 | Bumpon 3M SJ61A1 | Mechanical Part |

| | | |
|---|---|---|
| TP21 | VCC_IN | 3.0V/1.5V, LDO or battery |
| TP22 | UTX | UART Transmit |
| TP23 | URX | UART Receive |
| TP24 | UCTS | UART Clear to Send |
| TP25 | URTS | UART Request to Send |
| TP26 | SWDIO | JTAG input/output |
| TP27 | SWCLK | JTAG clock |
| TP28 | VPP_EN | VPP enable |
| TP29 | VPP | VPP (6.8V when VPP_EN is high) |
| TP30 | RST | reset |
| TP31 | AD_DIS | analog trigger control |
| TP32 | GND | ground point |
| TP33 | GND | ground point |
| TP34 | GND | ground point |
| TP35 | VREG | FT2232H regulator (1.8V) |
| TP36 | VBAT1V | voltage input for boost cfg |
| TP37 | VBAT3V | voltage input for buck cfg |
| TP38 | VBAT | DA14580 I/O voltage |
| TP39 | AD_DO | SPI data out (low if AD_DIS is high) |
| TP40 | VBUS | USB VBUS (connect a USB cable) |
| TP41 | GND | ground point |
| TP42 | USBDM | USB signal (neg) |
| TP43 | USBDP | USB signal (pos) |
| **Buttons** | | |
| RESET | Lila | Reset button. Resets all devices on board |
| K1 | Lila | Connects to P1_1 (J15[5,6] should be shorted) |
| K2 | Lila | Connects to P0_6 (J16[5,6] should be shorted) |

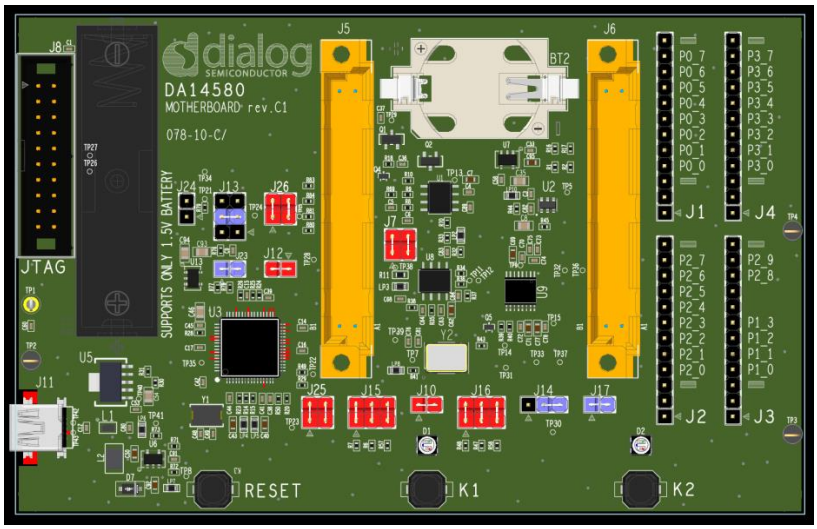



**Figure 9: Jumper Settings for Buck Configuration**

**Note 1** The jumpers depicted with red color are possible sources of leakage

**Table 5: Development Kit Motherboard components**

| Jumper Block | Description | Default State |
|---|---|---|
| J7 | Power measurement input | populated (1-2,3-4) |
| J10 | Power measurement trigger | populated |
| J12 | VPP programming voltage enable | populated |
| J13 | Power input select | populated (3-4) |
| J14 | Daughterboard power input select | populated (2-3) |
| J15 | Key1, LED1 connection | populated |

| | | |
|---|---|---|
| J16 | Key2, LED2 connection | populated |
| J17 | RESET power source | populated |
| J23 | LDO voltage selection | populated |
| J24 | LDO output resistor bypass | not populated |
| J25, J26 | UART connection | populated (1-2,3-4) |

### 3.4.2 Schematics and layout

For the schematics and layout of the motherboard please refer to the respective documents in the portal.

**Table 6: Motherboard configuration overview**

| # | Description | Related Board Items | What to do |
|---|---|---|---|
| 1 | Enable P1_2 to activate the 6.5V on the VPP pin for the OTP programming | J12 | Apply Jumper to enable feature |
| 2 | Power supply of the measurement circuit | J13 | Jumper at<br>*1-3*: power from BT1 *(Boost Configuration*)<br>*3-5*: power from BT2 (*Buck Configuration*)<br>*3-4*: power from U13 (*default – 3.0V*) |
| 3 | DA14580 power supply | J14 | Jumper at<br>*1-2*: power from Alkaline (*Boost Configuration*)<br>*2-3*: power from Coin Cell (*Buck Configuration - default*) |
| 4 | General Purpose LEDs/Buttons | J15 | Jumper at<br>*5-6*: P1_1 connects to K1 button<br>*3-4*: P1_2 connects to D1 LED 1<br>*1-2*: P1_3 connects to D1 LED 2 |
| 5 | General Purpose LEDs/Buttons | J16 | Jumper at<br>*5-6*: P0_6 connects to K2 button<br>*3-4*: P0_7 connects to D2 LED 1<br>*1-2*: P1_0 connects to D2 LED 2 |
| 6 | Reset | J17 | Apply Jumper to allow RESET button to drive a reset |
| 7 | Selects 1.5 / 3 V | J23 | Selects the voltage output of LDO (U13) that provides power to DA14580 daughterboard through J13 (jumper on 3-4)<br>Placed: 3.0V output<br>Not placed: 1.5V output |
| 8 | Shorts the 10 Ohm resistor at the LDO output | J24 | Do not apply Jumper. |
| 9 | Connects UART TX/RX to the default | J25 | Apply Jumpers. |

| | | | |
|---|---|---|---|
| | GPIOs | | |
| 10 | Connects UART RTS/CTS to the default GPIOs | J26 | Apply Jumpers. |
| 11 | Connects GPIO for measurement trigger | J10 | Apply Jumper. |

### 3.4.3 Configuring the motherboard

The motherboard can be configured with use of Jumpers that enable/disable various features. An overview is presented in the Table 4.

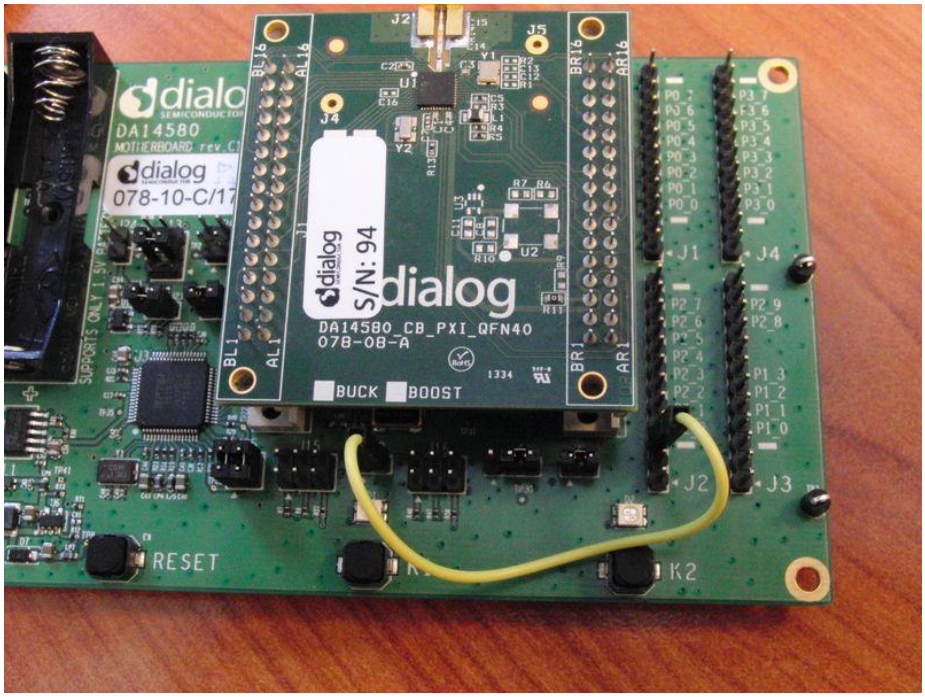



**Figure 10 : Cable connection of measurement trigger to ground**

### 3.4.4 Operation without measurement trigger (J10)

When the board is not required to use the measurement trigger function, jumper J10 can be removed. This jumper when placed connects GPIO P1_3 to the gate of Q5 and by issuing a high level it can create a signal that the PowerProfiler application detects and places a marker on the captured waveform. This can be useful for software debugging. Any other GPIO from 580 can be used by connecting a wire to J10, but if the jumper is not placed at all, it may pick-up noise (from EMI etc.) and produce false triggers.

The issue can be avoided by connecting the gate of Q5 to ground (Figure 5-10) or – a more permanent solution – by adding a pull down resistor (100 kΩ), as shown on Figure 5-11. The resistor keeps the gate low when it is left unconnected.

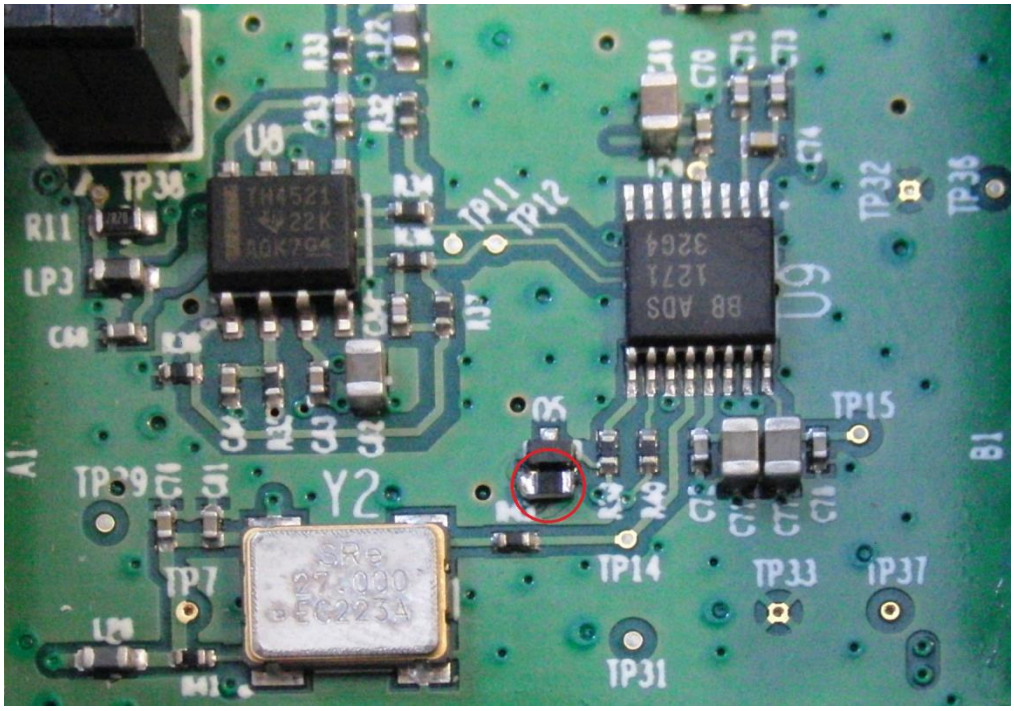



**Figure 11 : Pull down resistor to the gate of Q5 FET**

## 3.5 USB Dongle

### 3.5.1 Usage

The Development kit comes also with a standard USB Bluetooth Smart Adapter (or Dongle). For more info on its usage please refer to the UM-B-010.

### 3.5.2 Schematic

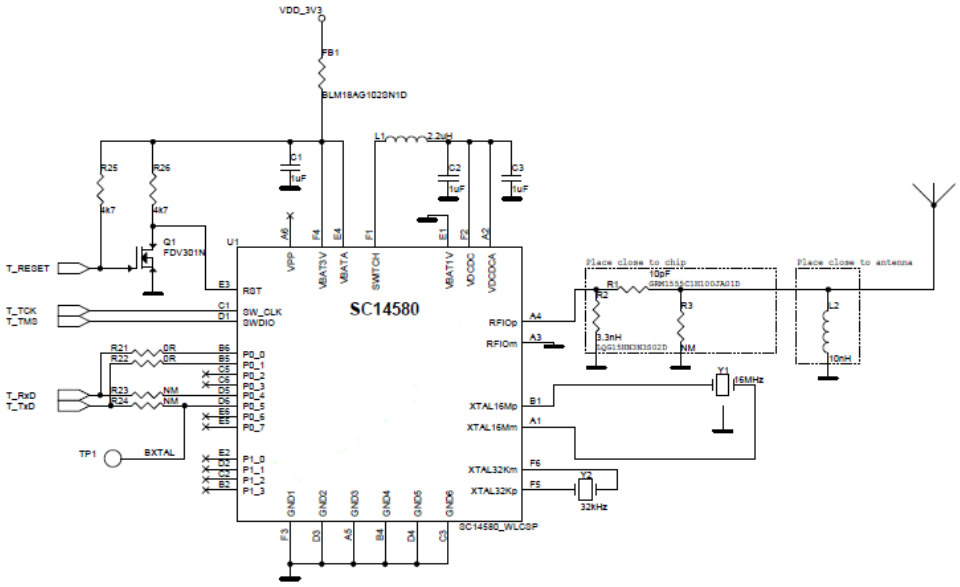
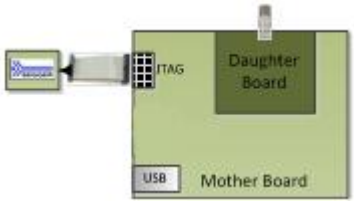
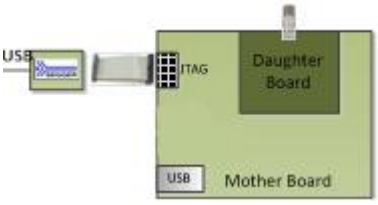
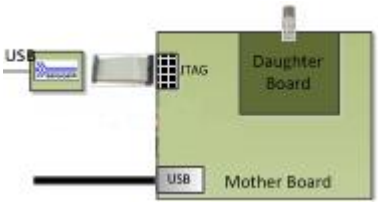



**Figure 12: USB Dongle Schematic**

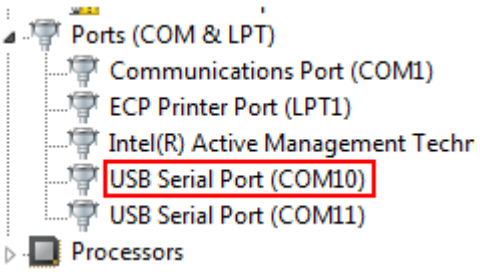
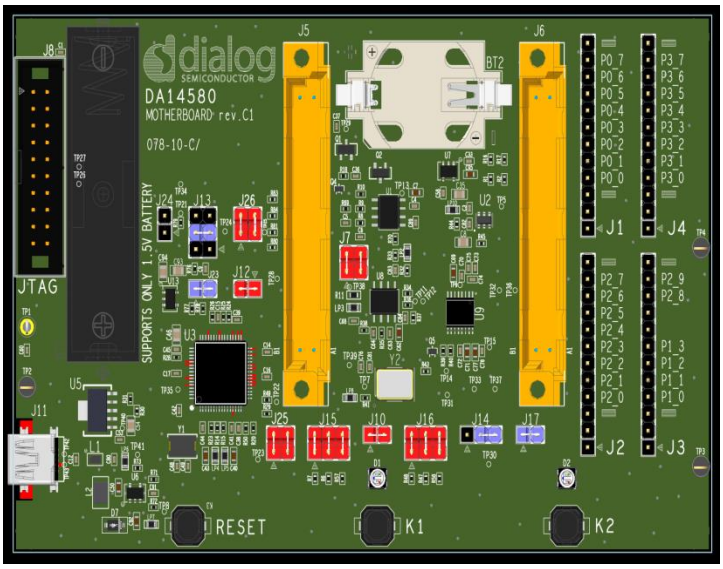
# 4 Quick start guide

## 4.1 Preparations

This chapter shows the user how to quickly set up the software development environment of the Bluetooth LowSmart demo

| | | |
|---|---|---|
| **4.1.1** | Stack the daughter board on top of the mother board. |  |
| **4.1.2** | Connect Jlink debugger to the mother board with flex cable. |  |
| **4.1.3** | Connect Jlink debugger to a PC with a mini-USB-to-USB cable. |  |
| **4.1.4** | Connect the USB cable. |  |

| | | |
|---|---|---|
| **4.1.5** | Normally the UART connection is supplied by the USB cable. If you want to use an external serial adapter, remove the jumpers from headers J25 and J26 and connect the appropriate external cable to either the default pins or whatever GPIOs you want to use.<br>However, the use of other pins is ***optional,*** while the default is to use the virtual COM port over USB.<br>The virtual COM port number assigned is always the first one, as is seen at the example *Windows Device Manager* screenshot (*here COM10*). |  <br>  |
| **4.1.6** | Make sure that the jumper configuration on your motherboard is the same as the side figure (*Buck configuration*).<br>For more configuration options refer to *Table 5* |  |

| DAUGHTERBOARD | | |
|---|---|---|
| **Signal Name** | **FTDI Pin Number** | **DA14580 GPIO** |
| UTX | 17 | P0_4 |
| URX | 16 | P0_5 |
| UCTS | 18 | P0_2 |

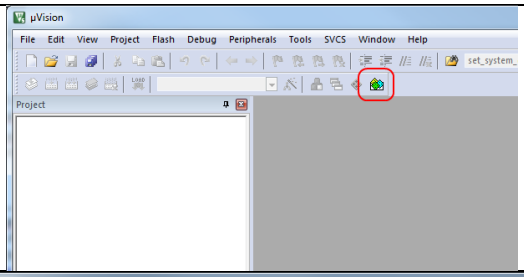
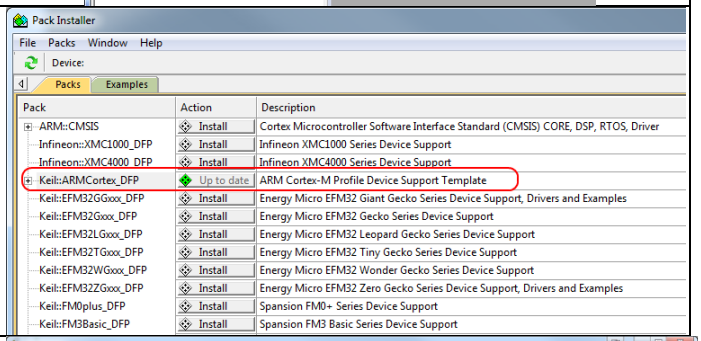
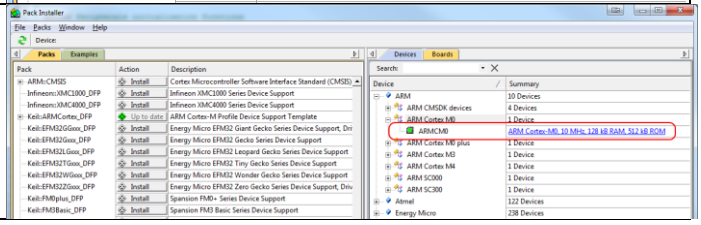
| | | |
|---|---|---|
| URTS | 19 | P0_3 |
| **USB DONGLE** | | |
| **Signal Name** | **ATMEL AVR32 Pin Number** | **DA14580 GPIO** |
| UTX | D10 | P0_4 |
| URX | D9 | P0_5 |

**Table 7: Default UART Connections for D/B & USB Dongle**

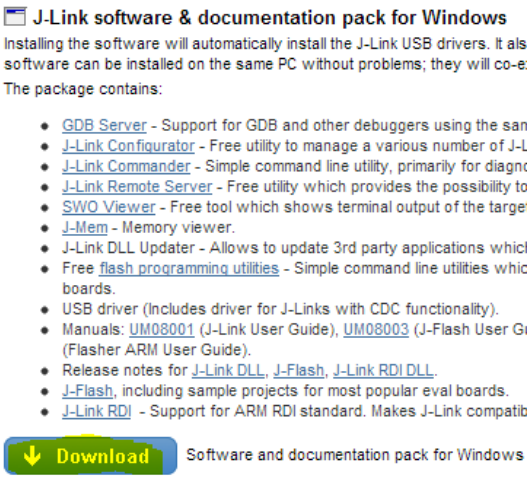
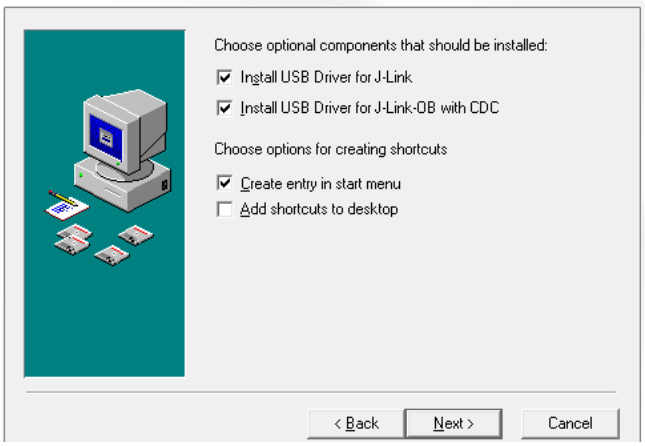
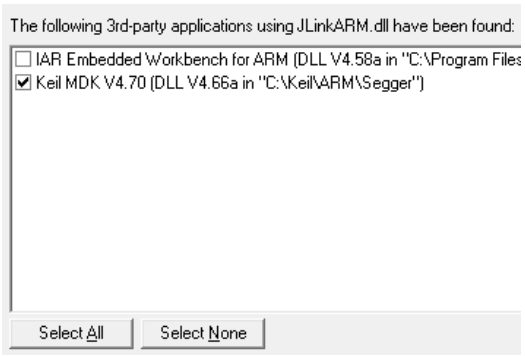
## 4.2 Installation of the tools and drivers

To install the Software development environment please follow the steps below.

### 4.2.1 Keil

| | | |
|---|---|---|
| 4.2.1.1 | Download and install Keil MDK-ARM uVision IDE.<br>Both uVision ver.4.0 and ver. 5.0 are supported. | Keil:<br>https://www.keil.com/download/product/<br><br>Keil MDK-ARM Version 5 – Installation:<br>http://www2.keil.com/mdk5/install<br><br>Starters Guide:<br>http://www.keil.com/uvision/ide_ov_starting.asp |
| 4.2.1.2 | For uVision Version 5 you have to install the ARM Cortex M profile package (see also http://www.keil.com/dd2/Pack/) |  |
| 4.2.1.3 | 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.<br>Click on the “Install” button to the right of “Keil::ARMCortex_DFP” package. |  |
| 4.2.1.4 | If the installation is successful, the pack installer window should look like this. |  |

### 4.2.2 SEGGER Jlink driver

| | | |
|---|---|---|
| 4.2.2.1 | Download and install the Jlink software & documentation pack for Windows.<br>*Please note that your SEGGER Jlink serial number is required for downloading.*<br>*This can be found on the plugin module as shown below*<br>  |  <br>http://www.segger.com/jlink-software.html |
| 4.2.2.2 | In order to have the USB Dongle properly recognised by Windows as a J-Link device, you have to install the driver with the settings shown in the side figure.<br>At the end of the installation, please tick the IDE (Keil MDK Vxx) that you are using. |  <br>  |

### 4.2.3 FTDI driver

| | | |
|---|---|---|
| 4.2.3.1 | The Development Kit uses the D2xx driver.<br>For Windows, this driver is part of the Combined Driver Model (CDM) driver. | USB Drivers:<br>http://www.ftdichip.com/Drivers/D2XX.htm |

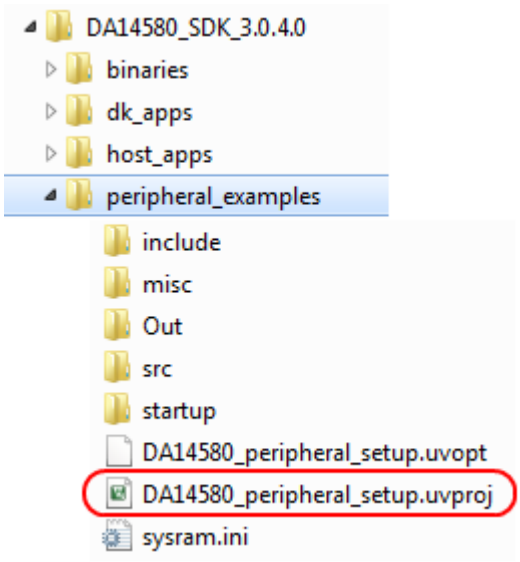
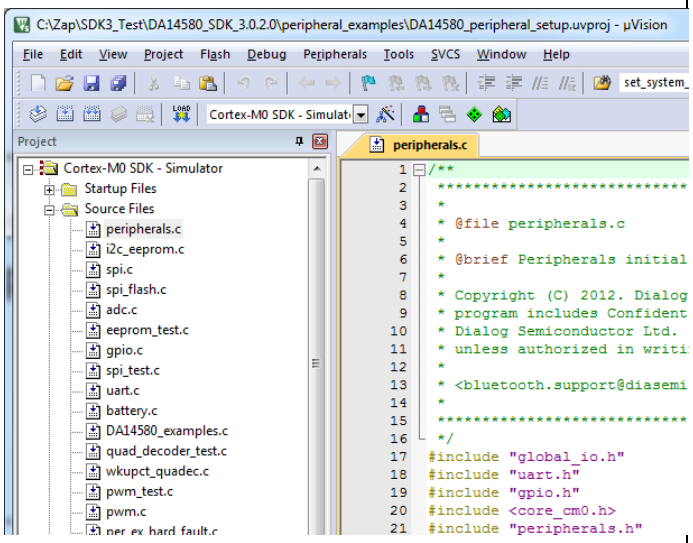
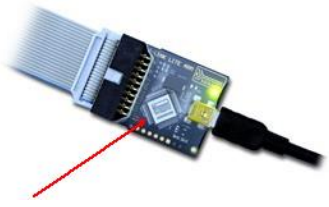
| | | |
|---|---|---|
| | (It is recommended that the latest driver available from the page below is used.) | Installation Guide: http://www.ftdichip.com/Support/Documents/InstallGuides.htm |
| **4.3 Tera Term** | | |
| 4.3.1 | Download and install Tera Term on your PC. | Tera Term: http://en.sourceforge.jp/projects/ttssh2/releases/ |

## 4.4 Using the demo kit

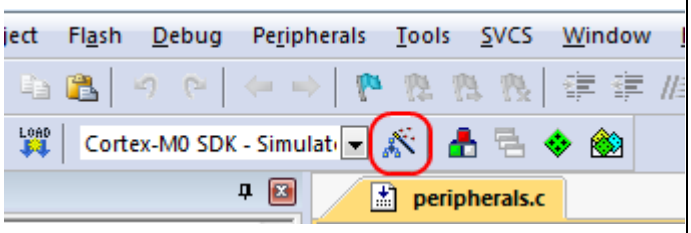
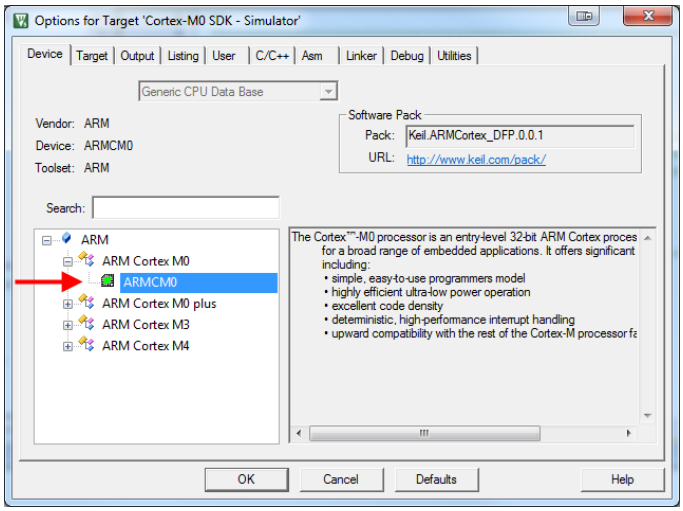
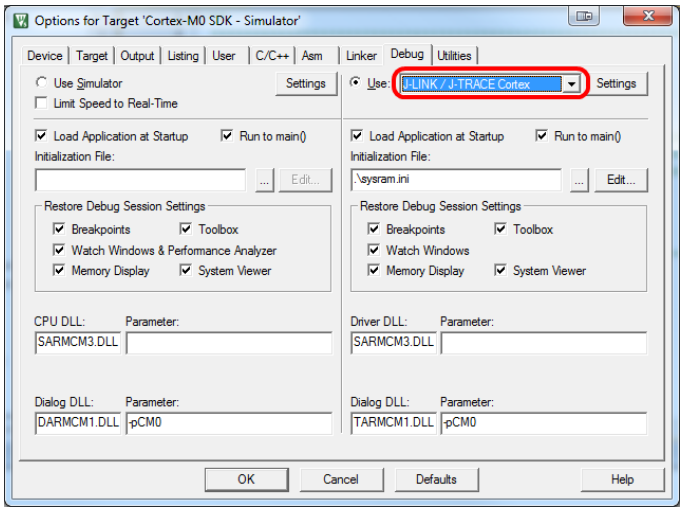
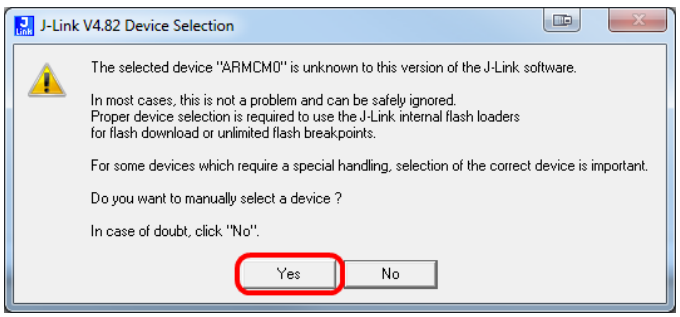
Follow these steps to easily create a working demo kit.

**NOTE: The following instructions may apply to all projects in the SDK**

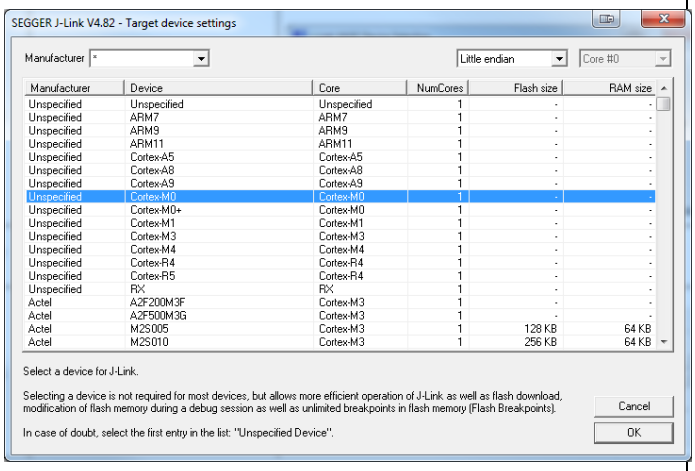
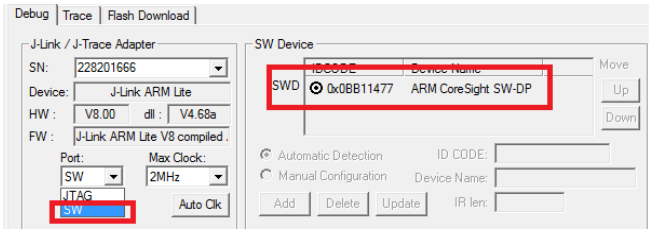
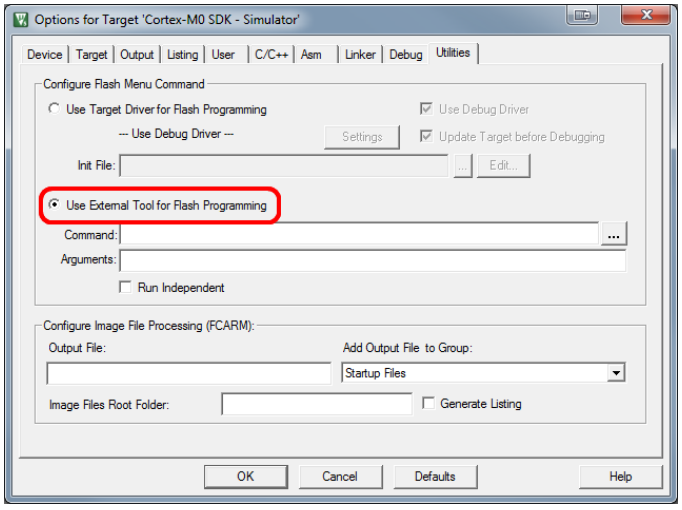
### 4.4.1 Run an example on DA14580

| | | |
|---|---|---|
| 4.4.1.1 | After you download SDK at http://support.dialog-semiconductor.com/ you can find a source code example project in example directory called "*peripheral_examples* Double click "DA580_peripheral_setup.uvproj", as shown in the image to the right. |  |
| 4.4.1.2 | The development environment should look like this when the project is opened with Keil. |  |
| 4.4.1.3 | Connect the J-Link debugger hardware. |  |

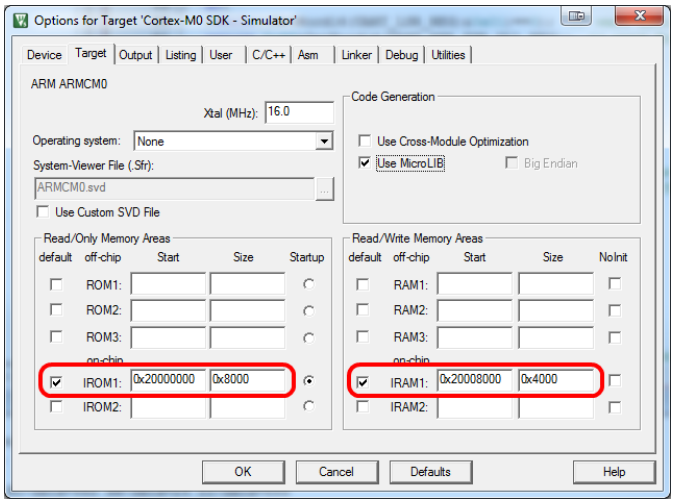
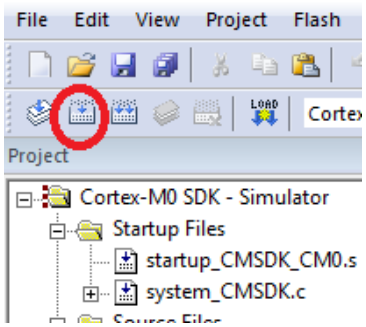
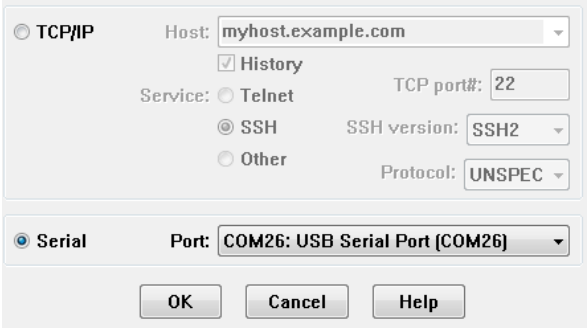
 

| | | |
|---|---|---|
| 4.4.1.4 | Click on the “Target Options” button (or ALT+F7) |  |
| 4.4.1.5 | Open the “Device” tab and select the ARMCM0 option as shown |  |
| 4.4.1.6 | Open the “Debug” tab and select “J-LINK/J-TRACE Cortex” from the drop down options.<br>Make sure that the the initialization file field is set correctly to “.\sysram.ini”.<br><br>Click on ‘Settings’. A pop-up window may be shown to inform you that a new firmware version is available.<br>Click ‘Yes’ to proceed<br>Note: If at this stage the Keil IDE environment crashes, look at Appendix A for a possible solution. |  |
| 4.4.1.7 | A pop-up screen may appear. Select ‘Yes’ to proceed |  |

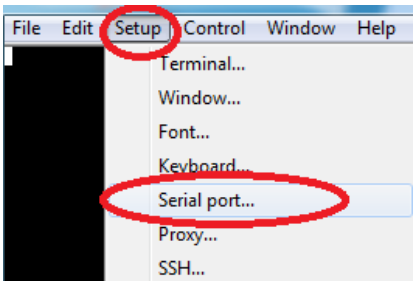
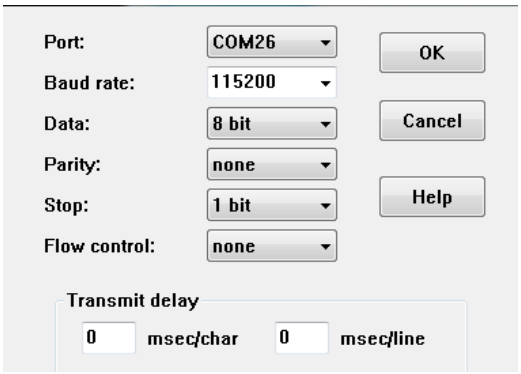
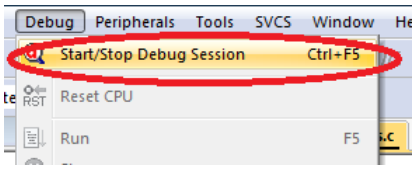
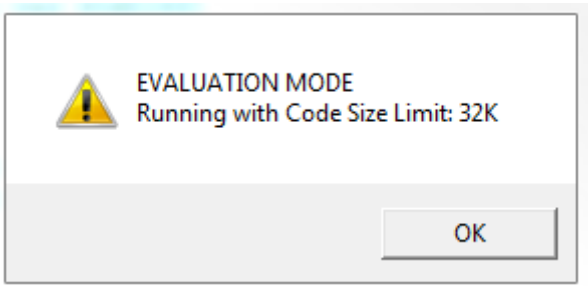
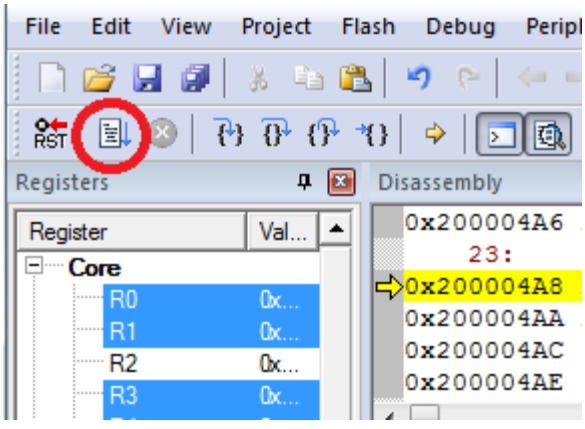
 

| | | |
|---|---|---|
| 4.4.1.8 | Select the ‘Cortex M0’ from the list and click ‘OK’ to proceed. |  |
| 4.4.1.9 | Change the Port setting from JTAG to SW.<br>Make sure the SW Device has been detected correctly. |  |
| 4.4.1.10 | Close the ‘Debug’ tab and open the ‘Utilities’ tab.<br>Since there is no Flash on board, select the option:<br>‘Use External Tool for Flash Programming’.<br>Click ‘OK’ |  |

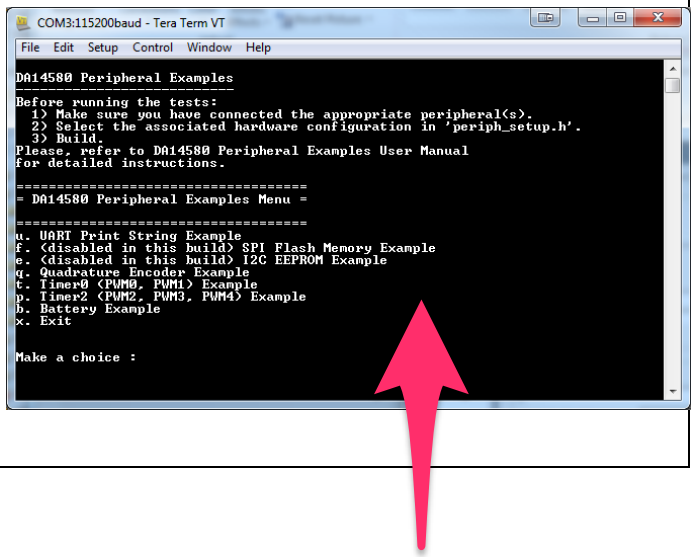
 

| | | |
|---|---|---|
| 4.4.1.11 | Open the ‘Target’ tab and change the ‘IROM1’ values to<br>0x20000000 / 0x8000<br>And the ‘IRAM1’ values to<br>0x20008000 / 0x4000<br>And click ‘OK’ |  |
| 4.4.1.12 | Click “OK” to save the settings. | All settings have been saved properly now, and you can continue to build the example. |
| 4.4.1.13 | Build the project by pressing “F7” key, or click the build button as shown in following picture |  |
| 4.4.1.14 | 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) |
| 4.4.1.15 | Open the Tera Term serial terminal on you PC. | |
| 4.4.1.16 | Open Tera Term and choose a COM port, which you have found in step 3, and click OK |  |

| | | |
|---|---|---|
| 4.4.1.17 | Choose Setup->Serial port to configure the Baud rate etc. |  |
| 4.4.1.18 | 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. |  |
| 4.4.1.19 | Go back to Keil Project. In the menu bar, select Debug->Start/Stop Debug Session. |  |
| 4.4.1.20 | A dialog window pops up, like the one on the right. Please click “OK” |  |
| 4.4.1.21 | Press F5 key or click execution button as shown in following picture, to start code execution. |  |

| | | |
|---|---|---|
| 4.4.1.16 | 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.<br>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. |  |

# Appendix A Opening your project for the first time

## A.1 Issue description

When, on a Keil uVision project, some entries in file .uvopt is missing or the file is missing, then, when the user clicks on the button 'settings' (options{debug tag}) with the{J-LINK/J-TRACE Cortex} selected, uVision crashes.

## A.2 Possible causes

Some important information concerning the j-link driver is missing. Calling the driver's dll probably causes the crash.

## A.3 Versions of Keil uVision found to be affected

At least versions 5.11.1.0 and 5.10.0.2 are affected.

## A.4 Under which circumstances user will encounter this error

When a local GIT repository is first created, this file (.uvopt) does not exist, since it is not included in the remote repository. When the user opens the project for the first time, this file is created, but some keys/values are missing.

## A.5 A proposed solution

1) Ensure that the .uvopt file does not exist in the folder of your project. If it exists and crash has been identified to happen, delete the .uvopt file.
2) Open the Keil project and close it. The .uvopt file is created automatically in the project folder (where the .uvproj is located).
3) Open the .uvopt file, using your favourite text editor.
4) Under the key <TargetOption> add the flowing lines:

```
<TargetDriverDllRegistry>
<SetRegEntry>
<Number>0</Number>
<Key>JL2CM3</Key>
<Name>-U228202424 -O78 -S0 -A0 -C0 -JU1 -JI127.0.0.1 -JP0 -RST0 -N00("ARM
CoreSight SW-DP") -D00(0BB11477) -L00(0) -TO18 -TC10000000 -TP21 -TDS8007 -
TDT0 -TDC1F -TIEFFFFFFFF -TIP8 -TB1 -TFE0 -FO7 -FD20000000 -FC800 -
FN0</Name>
</SetRegEntry>
</TargetDriverDllRegistry>
```

5) Save the .uvopt file and close the text editor.
6) Open the Keil project in uVision.
7) Click on *Project→Options for Project 'XXX'.*
8) On the 'Debug' Tab, select J-Link / J-TRACE Cortex debugger and click on the 'Settings' button for the debugger (not the simulator). This is the instance where the crash would happen.
9) The 'Cortex JLink/JTrace Target Driver Setup' Dialog opens. *Select your debugger as normally.*
10) Close the dialog windows clicking ok.
11) *Now, normal operation of j-link debugger is resumed.* After you have finished your work, close the Keil uVision IDE to allow for updates to the *.uvopt file to be saved.*

# Revision history

| Revision | Date | Description |
|---|---|---|
| 1.0 | 18-Mar-2014 | Initial version for DA14580-01 |
| 1.1 | 9-May-2014 | Added more Keil installation instructions |
| 1.2 | 5-Aug-2014 | Added Appendix A |
| 1.3 | 8-Aug-2014 | Updated title and textual changes |

### 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