PAN1740 Design Guide

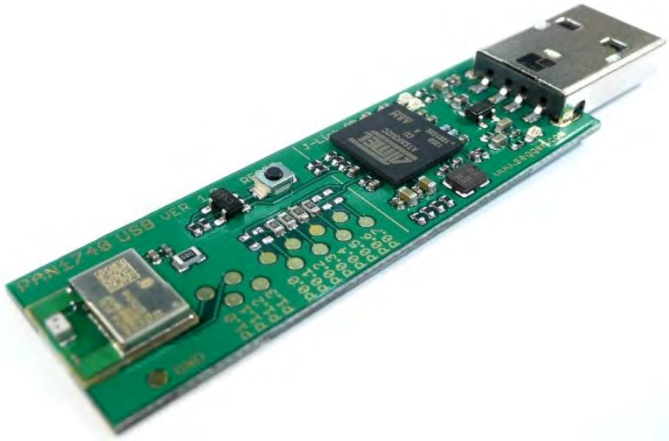


TABLE OF CONTENTS

1. Scope of this Document 4
2. Key benefits when using PAN1740 4
3. Bluetooth Low Energy 4
4. Description PAN1740 5
5. Block Diagram PAN1740 Module 6
6. PAN1740 USB-Dongle 7
   6.1. Functionality 7
   6.2. Schematic 8
7. Mother Board 9
   7.1. Functionality 9
   7.2. Placement 10
8. PAN1740 Adapter Board 11
   8.1. Schematic 11
9. Beacon 12
   9.1. Schematic 12
10. Beacon Kit 13
11. Layout Recommendation 14
12. Development of Applications 15
   12.1. Basic Platform Tools 18
   12.2. Dialog Specific Tools 18
   12.3. Bluetooth Address and Crystal Trim Values 18
   12.4. Example for Connection Manager 19
      12.4.1. USB Driver 19
      12.4.2. Start the Connection Manager 20
      12.4.3. Make PAN1026 Peripheral visible 20
      12.4.4. Open a Connection 21
      12.4.5. IPhone Demo 22
      12.4.6. Next Steps 22
13. Run the Keil project example 23
14. Example Proximity Profile with two PAN1740-USB dongles 26
15. Smart Snippets 29
   15.1. Program Structure 29
      15.1.1. Project and Port Selection 29
      15.1.2. Board Setup 29
      15.1.3. UART Booter 30
      15.1.4. Power Profiler 30
      15.1.5. Sleep Mode Advisor 31
      15.1.6. OTP Programmer, SPI Flash Programmer and EEPROM Programmer 33
      15.1.7. SPotA (Software Patch over the Air) 33
   15.2. Application Demo: Proximity Profile 34

16. Dialog Serial Port Service (DSPS)....35
16.1. Pro Kit with PAN1740 Adapter Board - IPod....35
16.2. Two Pro Kits with PAN1740 Adapter Board....37
16.3. Two PAN1740 USB sticks....37
17. Production Tools....39
18. History for this Document....40
19. Related Documents....40
20. General Information....41
21. FCC Warning....41
22. Life Support Policy....41

1. SCOPE OF THIS DOCUMENT

This Design Guide applies to the Bluetooth development modules PAN1740 Experimenter Kit (PAN1740 EXP). The intention is to enable our customers to easily and quickly integrate Panasonic's PAN1740 module in their product.

This guide describes the Hardware and gives useful hints.

2. KEY BENEFITS WHEN USING PAN1740

- Single-mode Bluetooth Smart System-on-Chip
- ARM CORTEX M0 CPU
- Small 9.0 x 9.5 x 1.8 mm SMD package with antenna
- Includes 16 MHz and 32.768 kHz crystal
- Temperature Range from -40°C to +85°C
- Pre-programmed calibration data and BT-Address
- Peak Power consumption 4.9mA Rx and Tx
- Less than a few µA in low power modes
- Link budget 93dBm (Rx Sensitivity -93, Tx 0 dBm)

3. BLUETOOTH LOW ENERGY

Bluetooth Low Energy (BLE), part of Bluetooth Ver. 4.0, specifies two types of implementation: Single mode and dual mode. Single mode chips implement the low energy specification and consume just a fraction of the power of classic Bluetooth, allowing the short-range wireless standard to extend to coin cell battery applications for the first time. Dual mode chips combine low energy with the power of classic Bluetooth and are likely to become a de facto feature in almost all new Bluetooth enabled cellular phones and computers.

Bluetooth marks

According to the new Bluetooth SIG marks "Bluetooth Smart" (single mode → mainly sensors) and "Bluetooth Smart Ready" (dual mode → gateway and hub devices) the PAN1740 fulfills criteria to label a product as a Bluetooth Smart device.

4. DESCRIPTION PAN1740

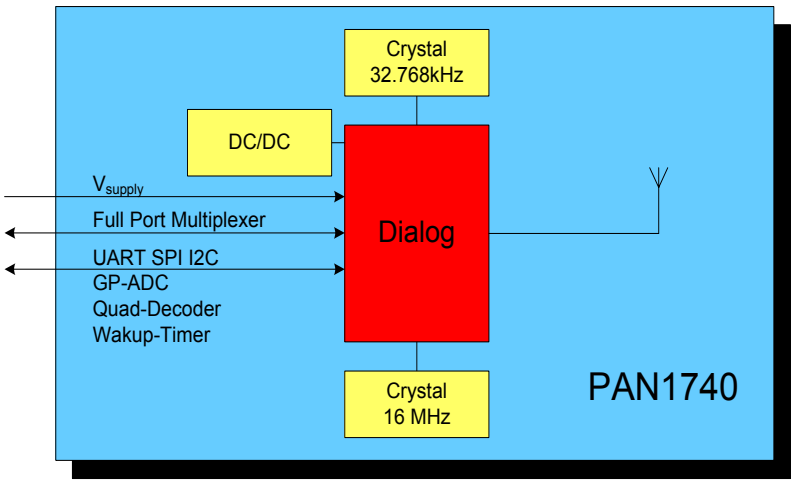
PAN1740 is the next generation Bluetooth Low Energy Module from Panasonic with reduced form factor, significantly lower power consumption and embedded Software Stack. The Single mode Bluetooth Smart System-on-Chip module is optimized for low power, small size and low system cost products. It reduces external component count, development effort and time to market.

The Module is manufactured in a very small 9.0* 9.5* 1.8 mm SMD package with shielded case and chip antenna. The power consumption of only 4.9mA in Tx and Rx mode makes the use of coin cell batteries possible or reduces the needed battery capacity and cost of existing solution by at least 50%. It is qualified according to Bluetooth 4.0 standard. FCC, IC and CE approvals are under preparation.

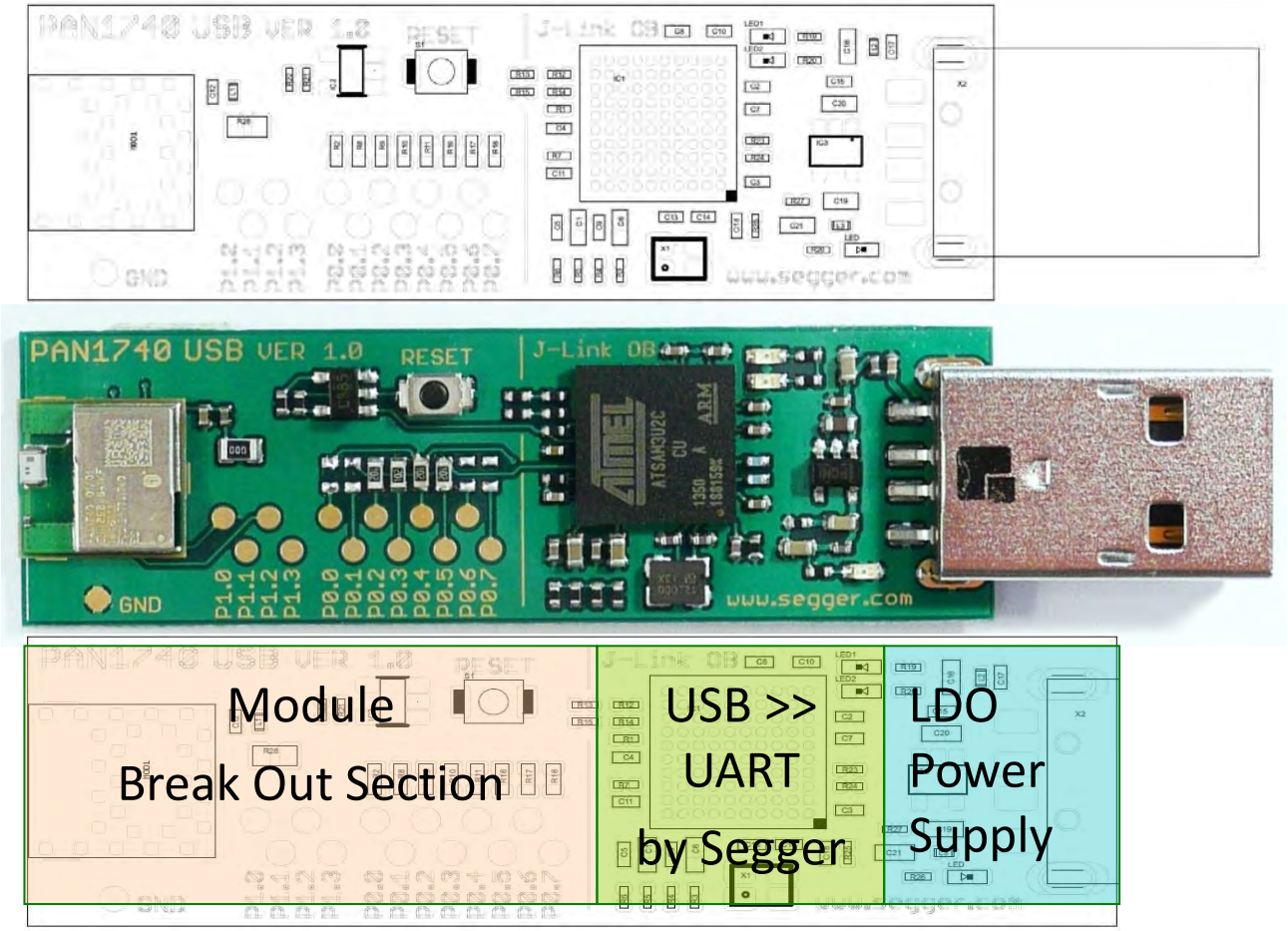
The PAN1740 comes with a complete software development platform, which includes a qualified Bluetooth Smart single-mode stack that can be compiled with a number of available BLE profiles, custom application and programmed on the module. Multiple Bluetooth Smart profiles for consumer wellness, sport, fitness, security and proximity applications are supplied as standard, while additional customer profiles can be developed and added as needed.

Please contact your local sales office for further details on additional options and services, by visiting www.panasonic.com/rfmodules for U.S. and http://industrial.panasonic.com/eu/ for Europe or write an e-mail to wireless@eu.panasonic.com

5. BLOCK DIAGRAM PAN1740 MODULE



6. PAN1740 USB-DONGLE



6.1. Functionality

Atmel µC includes Segger USB-to-UART programmer (serial number on the backside)

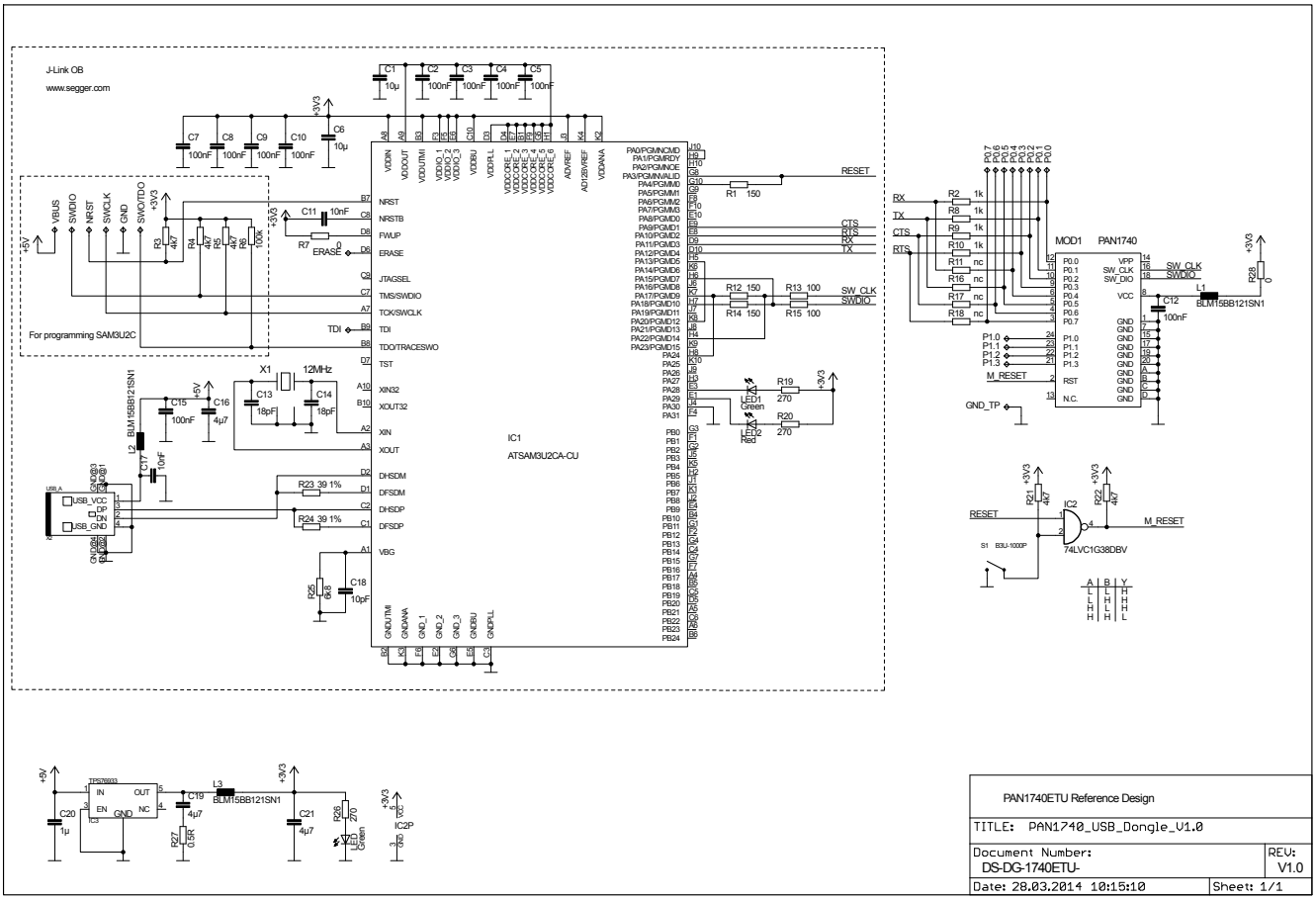
OTP cannot be damaged - Failsafe development

Runs with Dialog's Keil compiler projects

Runs with "Connection Manager"

Can be used for SW development "on the fly"

6.2. Schematic



7. MOTHER BOARD

7.1. Functionality

The details of the mother board are described in the Dialog documentation

Can be used with PAN1740 Adapter Board

OTP can be programmed

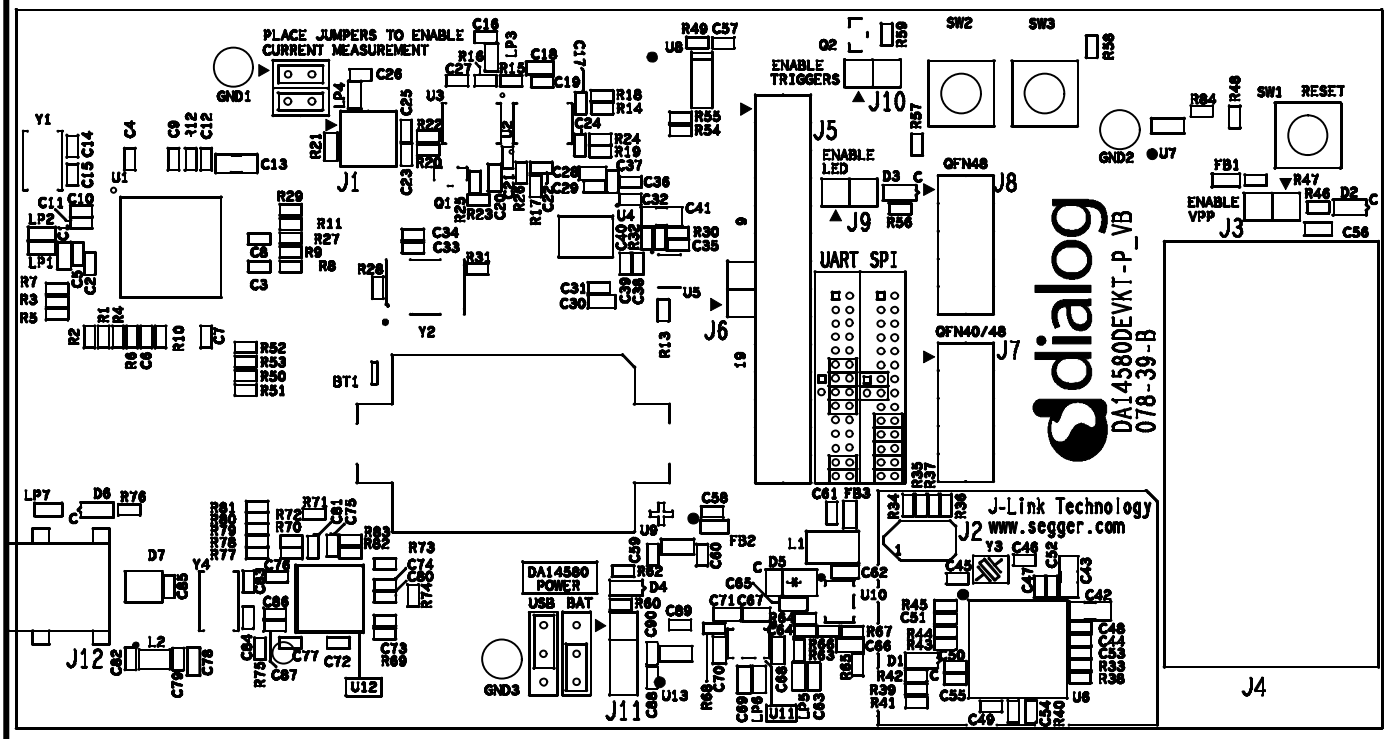
Runs with Dialog's Keil compiler projects

Runs with "Connection Manager"

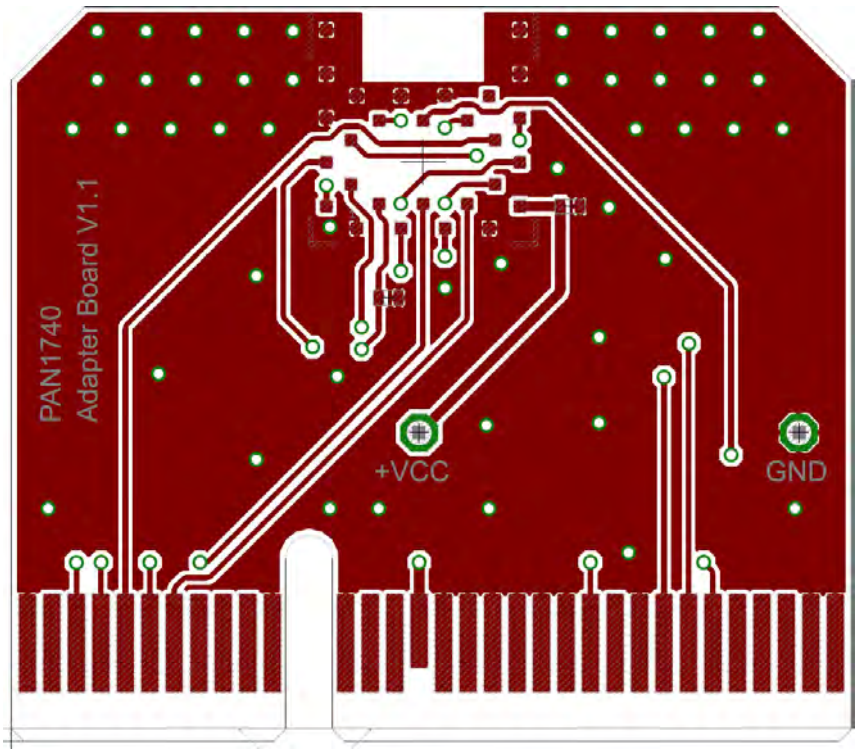
Can be used for SW development "on the fly"

Runs with "Smart Snippets" including Power Profiler

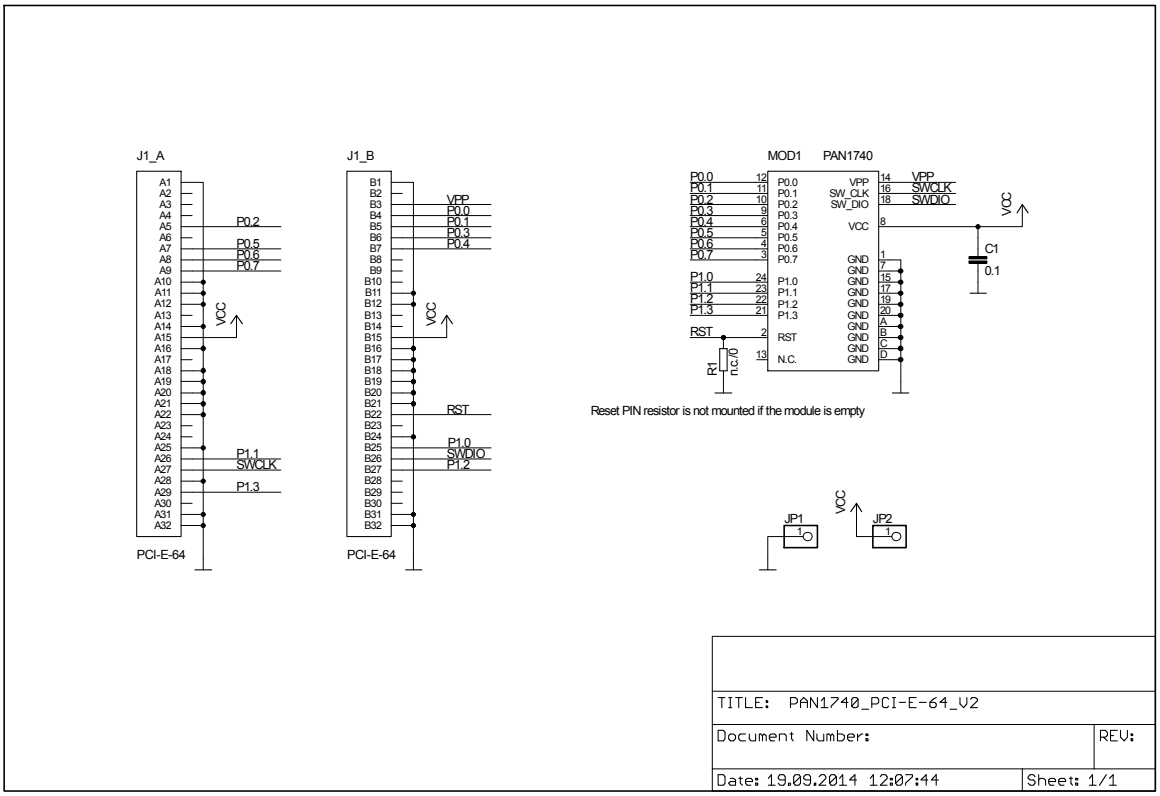
7.2. Placement



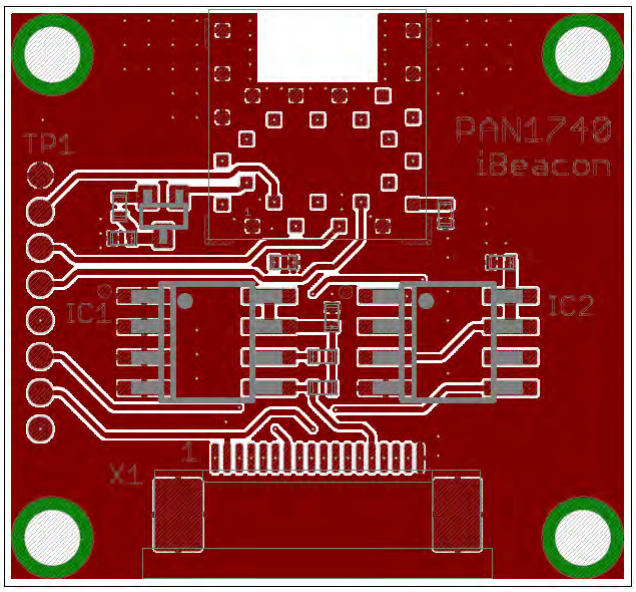
8. PAN1740 ADAPTER BOARD



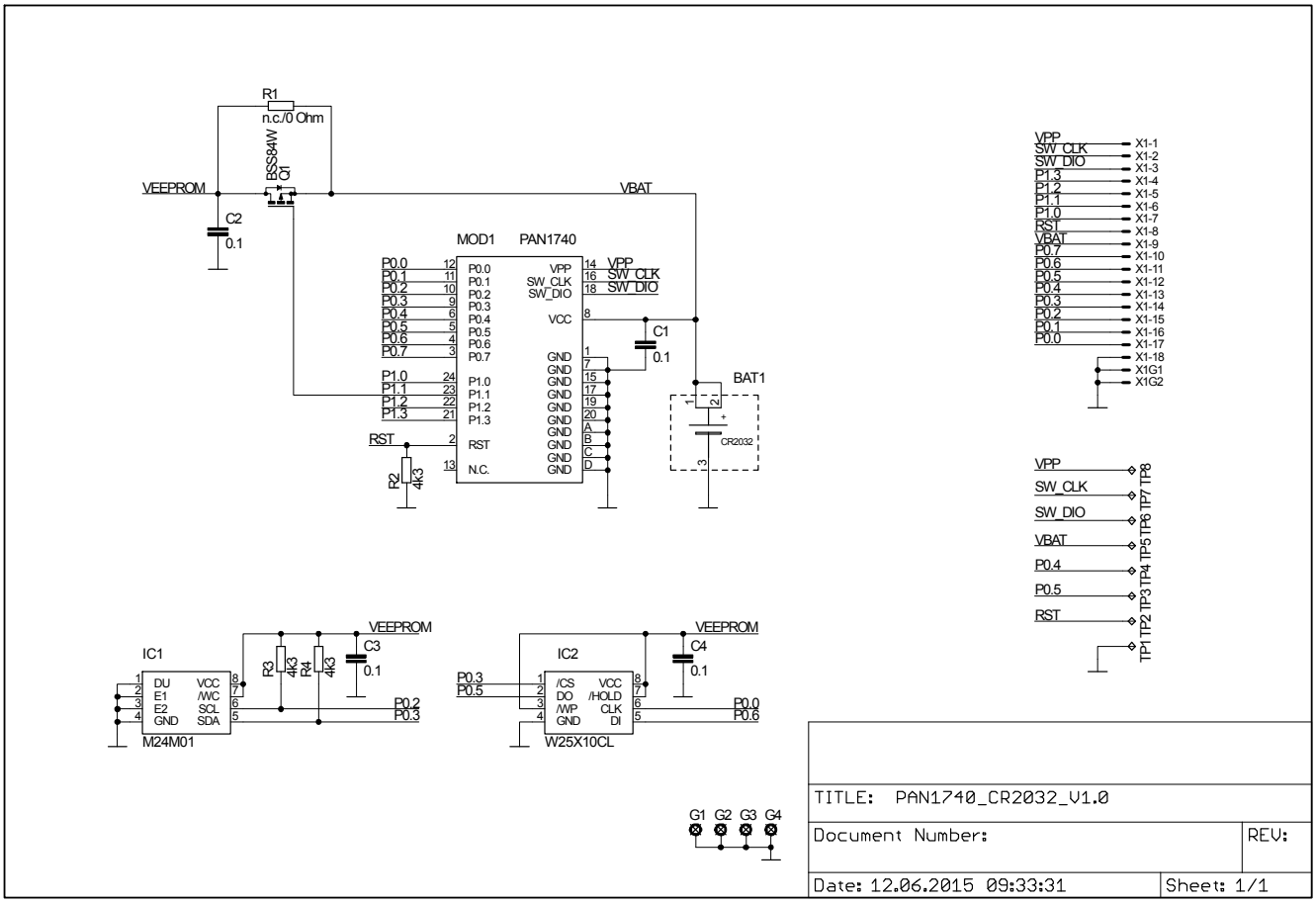
8.1. Schematic



9. BEACON



9.1. Schematic



10. BEACON KIT

The Beacon-Kit consists of:

- 1 pc. Dialog Motherboard
- 1 pc. Flexible Flat Cable – 0.50 mm pitch – 18 conductors – 50 mm length – Type 2
- 1 pc. Beacon Adapter Board
- 5 pcs. Beacon

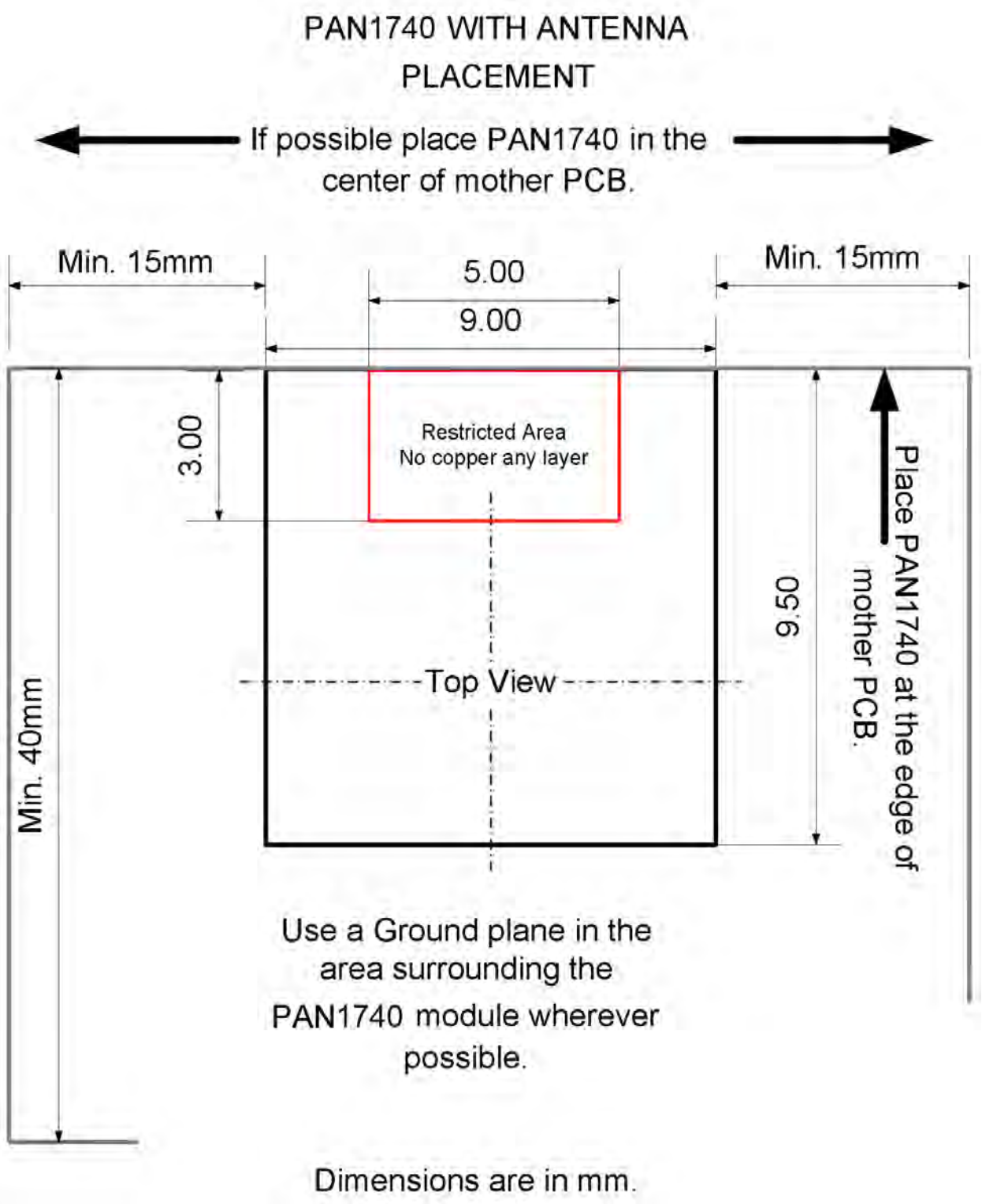
The Beacons come along pre-configured with Dialog's Beacon Software which makes it easy to start evaluating. Simply power them up by plugging a CR2032 battery and watch out for them using the *Locate Beacon* App which can be downloaded in the Google Play Store. This scenario is depicted below.

The Beacon software is stored in the SPI Flash (IC2) on the PCB and can be changed by establishing a connection between Dialog's Motherboard and a Beacon via the Beacon Adapter Board and the FFC as shown below.

Note: An FFC longer than 5cm can cause transmission errors when using JTAG. Hence, longer cables should use the UART interface to download software to the Beacon.

Simply change Dialog's Beacon code available on their website to your needs with Keil uVision, compile it and download it to the device using Smart Snippets for instance. Please read the following chapters and Dialog's Documentation to be able to download your code into the Beacon.

11. LAYOUT RECOMMENDATION



Note: The above recommendation for the Ground plane is based on a double layer PCB. If additional ground planes are implemented in other layers and connected by sufficient vias the minimum xy dimensions can be reduced.

12. DEVELOPMENT OF APPLICATIONS

In this and the following chapters, the basic tools and some examples are shown to get the USB-dongles running. All examples use Windows7 OS and iPhone 5G, but also higher versions should work.

Installing and running HW drivers on any system requires time and each step may require re-booting your system. Contact your system administrator if any system related problems appear during the installation phase. Install the drivers step-by-step in the below order and use the latest software releases.

There are two basic approaches for implementation:

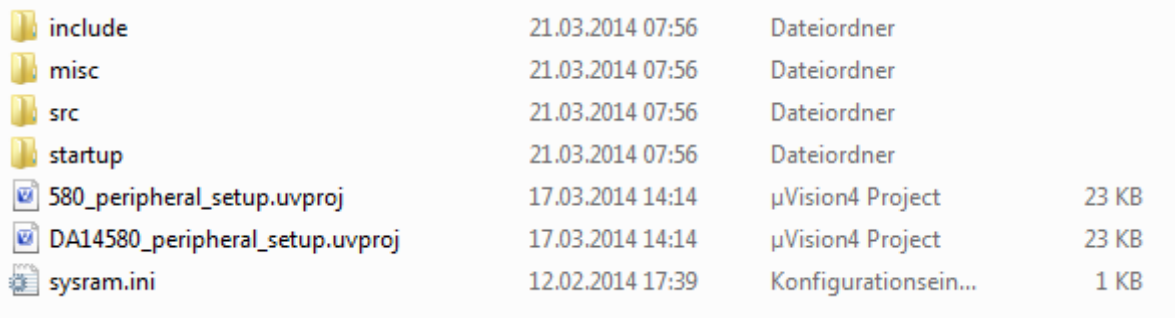
1. Full embedded HCI with external host controller

If BLE functionality is being added to a running application -- E.g. for porting the setup GUI from your application into a smart phone -- developing a smart phone "App" and installing either a proprietary BLE profile on the host controller and or a BT-Sig certified profile should be considered. The BT-Developers portal provides guidance for this process - http://developer.bluetooth.org/

2. Standalone sensor application

Applications requiring less than 32KB can reside and execute on the PAN1740 module. There are several examples in the Keil projects. The download link is in chapter 12.1. The profiles are located in the SDK folder (please use the latest release from Dialog website):

\DA14580_SDK_3.0.4.0\dk_apps\src\ip\ble\hl\src\profiles



Smart Snippets is needed to burn the application into the module. If you are working with the PAN1740ETU USB-Dongle applications can be developed and executed in RAM, but for security reasons it is not possible burn the OTP. Panasonic's Experimenters' kit is required to access OTP is accessible. Please contact your local sales distributor for support. The Experimenters' kit may be mandatory for production. For items beyond the scope of this design guide, refer to the Smart Snippets guide on the Dialog's support website. Following is a list of supported profiles.

dialog SEMICONDUCTOR smart{snippets}

	Software Features List (note 1)	DA14580-00		DA14580-01	
		2.0.4		3.0.2	3.0.4
	"X" = Supported feature "NS" = Not Supported Feature	release date: 23/12/2014		release date: 28/03/2014	release date: 30/06/2014
1	**Embedded Development Tools**				
1.1	Keil project environment	X		X	X
1.2	JTAG debugger	X		X	X
1.3	GNU / GCC toolset	NS		NS	X
2	**Stack**				
2. 1	**Protocol features**				
2.1.1	Bluetooth Smart 4.1 core stack	X		X	X
2.1.2	MTU size	23		23	23
2.1.3	Large packet support (L2CAP fragmentation)	NS		X	X
2.1.4	Master mode	X		X	X
2.1.5	Slave mode	X		X	X
2.1.6	Master and Slave sequentially	X		X	X
2.1.7	Multilink support (maximum links)	4		6	6
2.1.8	UUID 128bits	X		X	X
2. 2	**Low Energy features**				
2.2.1	BLE timer wakeup	X		X	X
2.2.2	GPIO wakeup	X		X	X
2.2.3	Quadrature wakeup	X		X	X
2.2.4	Active mode	X		X	X
2.2.5	Extended Sleep Mode	X		X	X
2.2.6	Deep Sleep	X		X	X
3	**Profiles**				
3. 1	**Health Profile**				
3.1.1	Blood Pressure Profile	X		X	X
3.1.2	Blood Pressure Service	X		X	X
3.1.3	Glucose Profile	X		X	X
3.1.4	Glucose Service	X		X	X
3.1.5	Health Thermometer Profile	X		X	X
3.1.6	Health Thermometer Service	X		X	X
3. 2	**Sports and Fitness Profile**				
3.2.1	Cycling Power Profile	NS		NS	X
3.2.2	Cycling Power Service	NS		NS	X
3.2.3	Cycling Speed and Cadence Profile	X		X	X
3.2.4	Cycling Speed and Cadence Service	X		X	X
3.2.5	Location and Navigation Profile	NS		NS	X
3.2.6	Location and Navigation Service	NS		NS	X
3.2.7	Heart Rate Profile	X		X	X
3.2.8	Heart Rate Service	X		X	X
3.2.9	Running Speed and Cadence Profile	X		X	X
3.2.10	Running Speed and Cadence Service	X		X	X

..\DA14580_SDK_3.0.4.0\dk_apps\src\ip\ble\hl\src\profiles

(Please use the latest release from Dialog website)

dialog SEMICONDUCTOR smart{snippets}

	Software Features List (note 1)	DA14580-00		DA14580-01	
		2.0.4		3.0.2	3.0.4
	"X" = Supported feature "NS" = Not Supported Feature	release date: 23/12/2014		release date: 28/03/2014	release date: 30/06/2014
3. 3	**Proximity Profile**				
3.3.1	Proximity Profile	X		X	X
3.3.2	Find Me Profile	X		X	X
3. 4	**Alerts and time Profile**				
3.4.1	Time Profile	X		X	X
3.4.2	Current Time Service	X		X	X
3.4.3	Reference Time Update Service	X		X	X
3.4.4	Next DST Change Service	X		X	X
3.4.5	Phone Alert Status Profile	X		X	X
3.4.6	Phone Alert Status Service	X		X	X
3.4.7	Alert Notification Profile	X		X	X
3.4.8	Alert Notification Service	X		X	X
3.4.9	Immediate Alert Service	X		X	X
3. 5	**Peripherals Profile**				
3.5.1	HID over GATT Profile	X		X	X
3.5.2	HID Service	X		X	X
3. 6	**Generic Profile**				
3.6.1	Scan Parameters Profile	X		X	X
3.6.2	Scan Parameters Service	X		X	X
3.6.3	Battery Service	X		X	X
3.6.4	Device Information Service	X		X	X
3.6.5	Link Loss Service	X		X	X
3.6.6	Tx Power Service	X		X	X
3.6.7	Accelerometer	X		X	X
4	**Peripheral Device Drivers/HAL**				
4.1	UART driver	NS		X	X
4.2	GPIO driver	NS		X	X
4.3	SPI driver	NS		X	X
4.4	SPI Flash driver	NS		X	X
4.5	I2C EEPROM driver	NS		X	X
4.6	ADC driver	NS		X	X
4.7	Battery Level driver	NS		X	X
4.8	PWM driver	NS		X	X
4.9	Quadrature driver	NS		X	X
4.10	Wakeup timer driver	NS		X	X

12.1. Basic Platform Tools

1. Windows 7 or higher - www.microsoft.com
2. Keil Compiler 32K free license – www.keil.com/arm/mdk.asp
3. Segger J-Link software - www.segger.com/jlink-software.html

Use the serial number from the bottom side of the USB-Dongle

4. Bluetooth 4.1 - www.bluetooth.org

12.2. Dialog Specific Tools

Download in the support forum http://support.dialog-semiconductor.com under the top menu bar "Software".

1. Connection Manager

Enables basic GATT connections between two BLE devices.

2. Dialogs Projects

Keil Projects with BLE Profiles and examples. These project files are used to implement the customers application. The BLE profiles are already implemented and you can setup the full feature set and I/O capability of the module.

3. Smart Snippets

Smart Snippets is the Tool for reading and writing the OTP. With this tool, the last step in the development chain can be performed by burning the OTP fuses. This tool does not work with the USB-Dongle for security reasons. The programming voltage of 6.8V needed to enable the programming is not supported.

12.3. Bluetooth Address and Crystal Trim Values

1. Main Frequency Calibration up to 1ppm @ 2.4GHz

Panasonic calibrates the 16 MHz crystal and writes this calibration data in the OTP header. This provides best performance and a stable frequency. Customers do not need to take care of this step in their production.

2. Pre-programmed Bluetooth/MAC Address

Each Bluetooth device must have a unique MAC address which is provided from the IEEE. Since this may lead to additional costs and registration effort for customers Panasonic burns a unique address into the OTP header from our database. Customers do not need to take care of this step in their production.

3. Precise High Performance Crystal Sleep clock

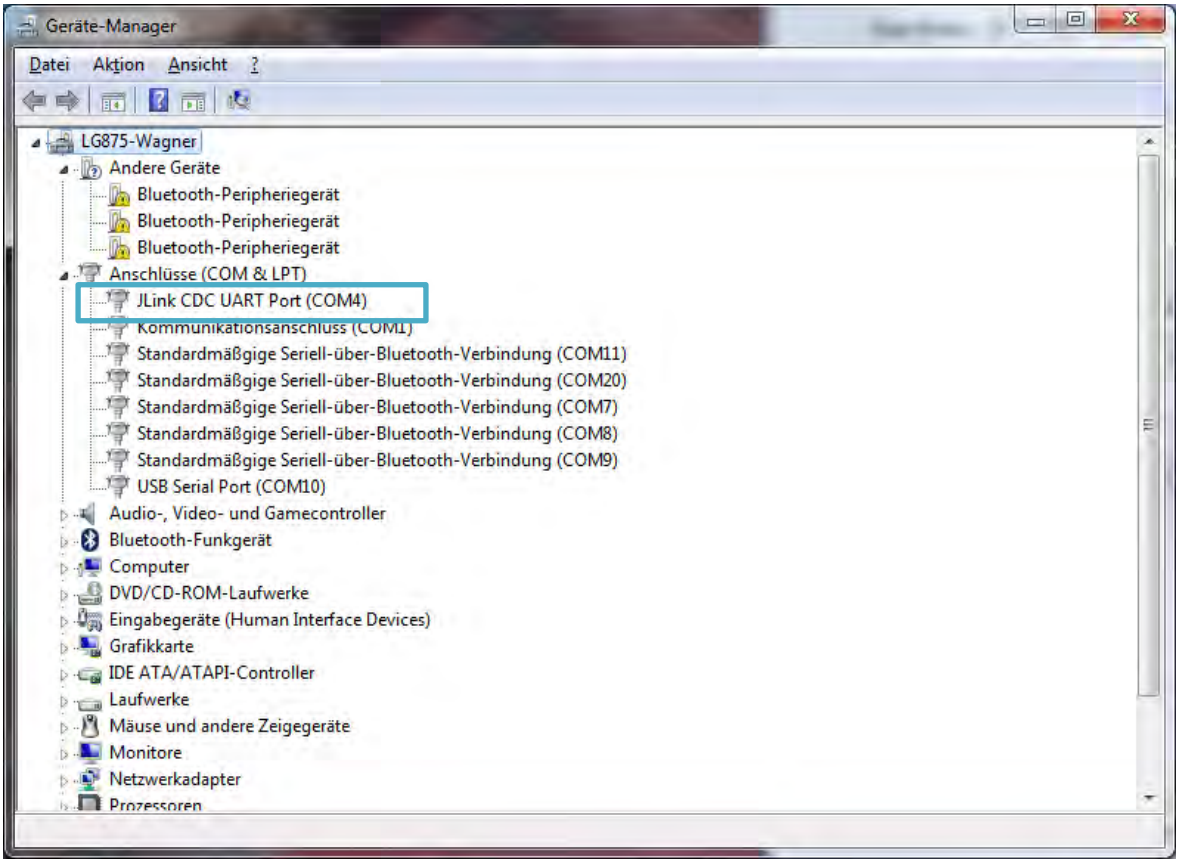
Since this is a low energy device and the key functionalities are the sleep functions with just a few µA current draw Panasonic has integrated a 32.768 kHz crystal clock into the PAN1740 module. Therefore no external components are needed and there are no hidden costs.

12.4. Example for Connection Manager

In the following example two USB-dongles are used. PAN1026-USB works as a peripheral and the PAN1740-USB as central device.

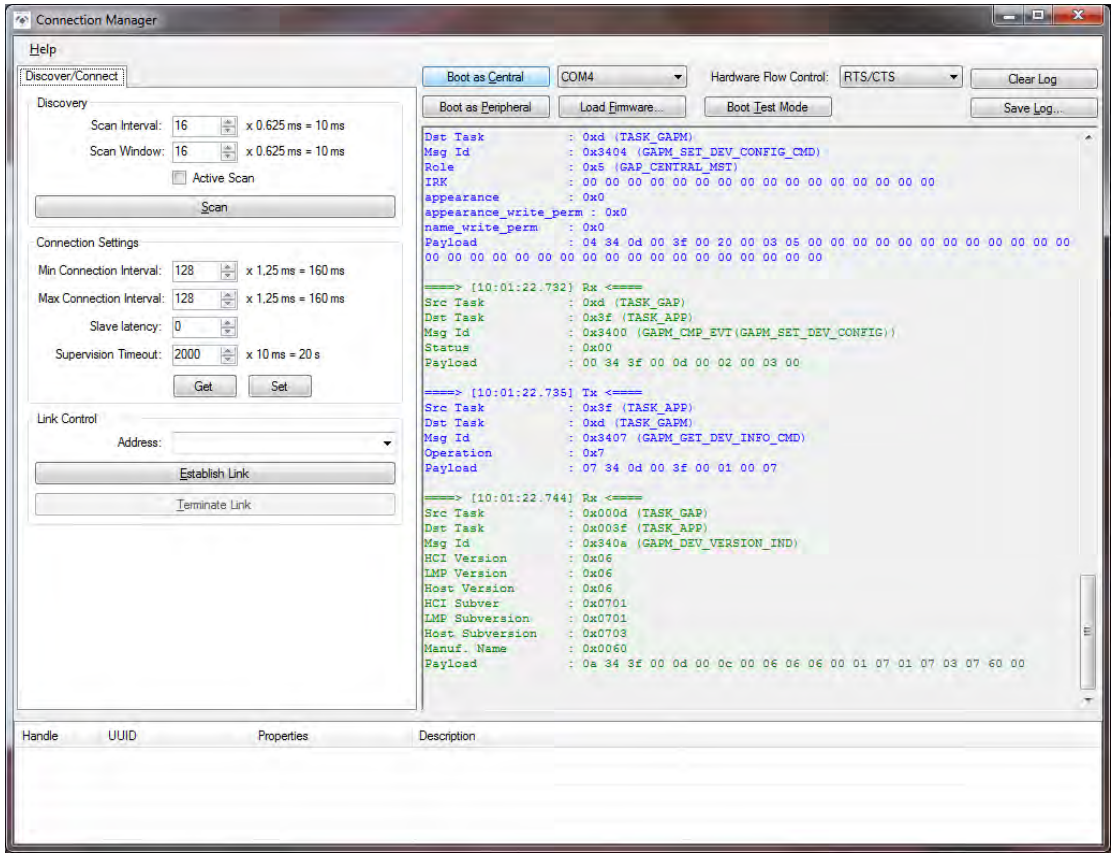
12.4.1. USB Driver

Download and install USB Drivers from Segger
www.segger.com/jlink-software.html



12.4.2. Start the Connection Manager

1. Make sure you have admin rights and access to the program folder
2. Wait until the init phase is finished
3. Press load Firmware and select "full_emb.hex"
4. Wait until the init phase is finished
5. Press either "Boot as Central" or "Boot as Peripheral" (Example shows Central)

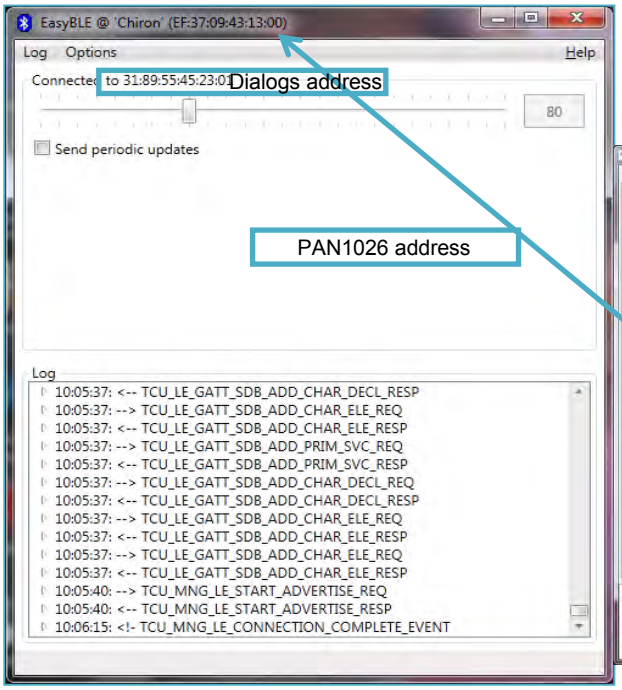


12.4.3. Make PAN1026 Peripheral visible

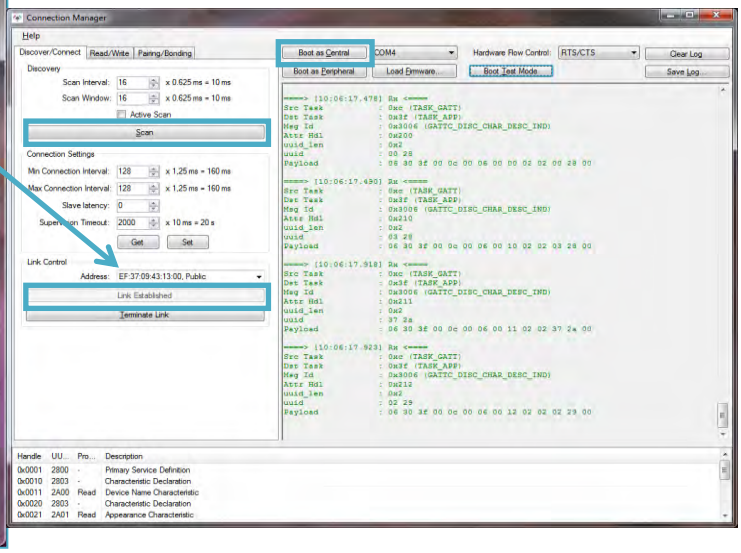
EasyBLE Tool and a PAN1026-USB dongle may be used to make a peripheral visible with the heart rate profile. Use this link to download EasyBLE:

http://pideu.panasonic.de/files/Documents/WM%20Documents/PAN1026/EasyBLE.zip

12.4.4. Open a Connection

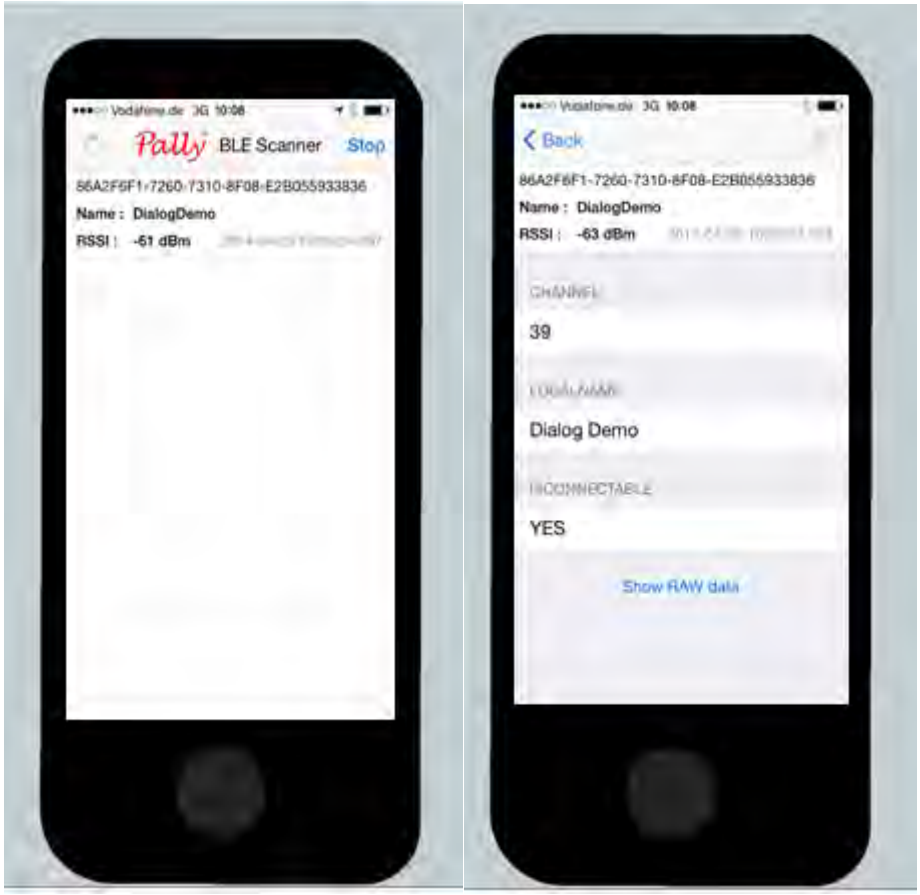


1. Press "Scan"
2. Select PAN1026 address and establish link



12.4.5. IPhone Demo

The PAN1026 used in the example above may be substituted with an iPhone (model 4S and up) and e.g. "Bluetooth Smart Scanner" App.



"Bluetooth Smart Scanner" may be replaced by other BLE Apps for Android or Windows based smart phones. Depending on the application, a proprietary app may be written or existing certified BLE profile used.

12.4.6. Next Steps

The PAN1026 was chosen to demonstrate sending a heart rate payload data, as Connection Manager cannot send data, since it installs only the central or peripheral functionality on the PAN1740, but not the profile. Two PAN1740 USB dongles can be connected by setting one side to Central and the other to Peripheral. To send data you need to run the Keil project in debug mode together with a profile e.g. peripheral example, chapter 13 describes this process.

13. RUN THE KEIL PROJECT EXAMPLE

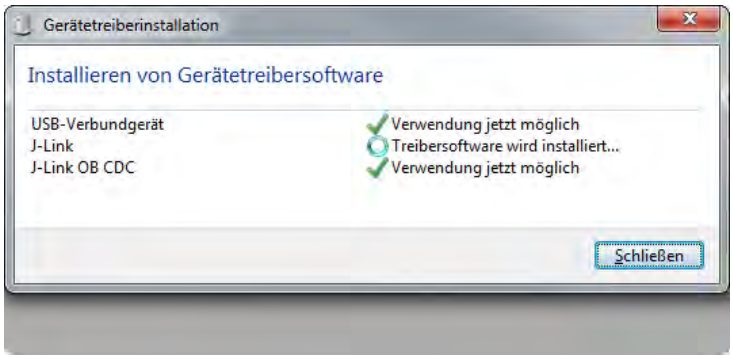
For more detailed information on project examples, refer to the Projects in Dialog's software download page. Recommended is the proximity example as this is the most common profile.

Download and install the SDK using the following URL:

http://support.dialog-semiconductor.com/software-downloads/index

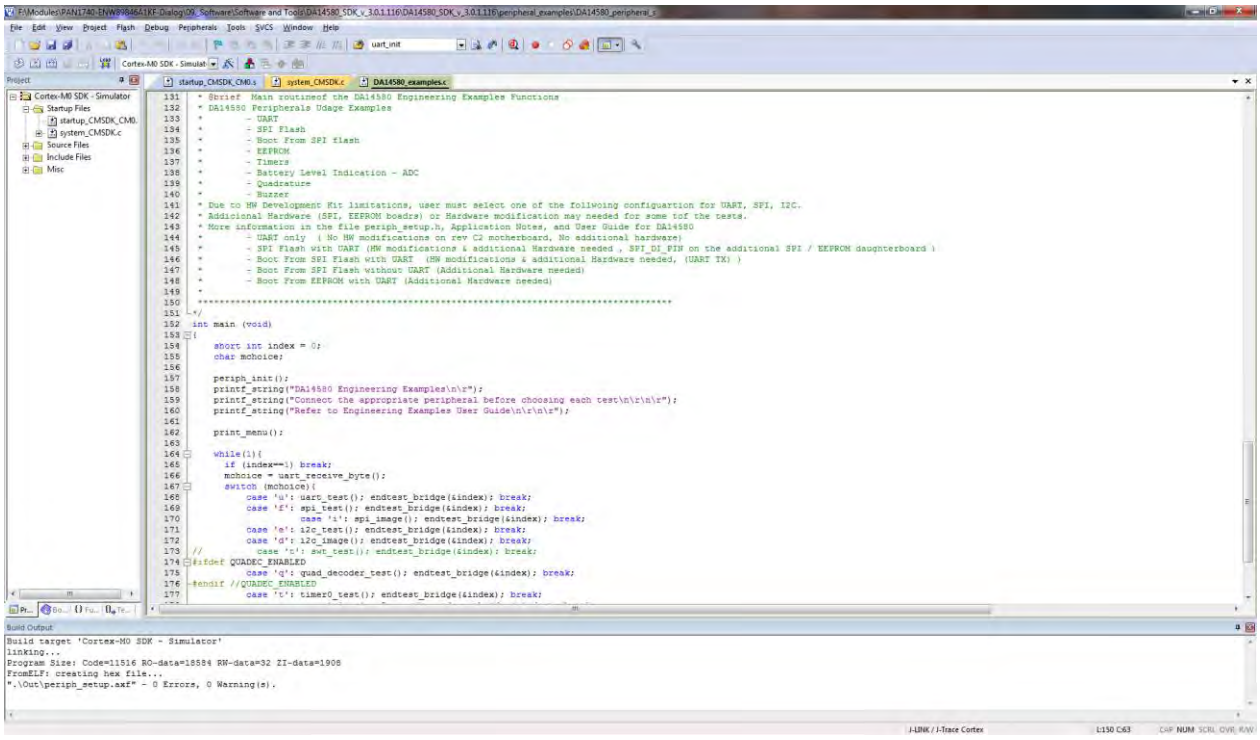
Install the dongle after the Segger J-Link software driver installation.

Refer to chapter 12.1 Basic Platform Tools to download them.

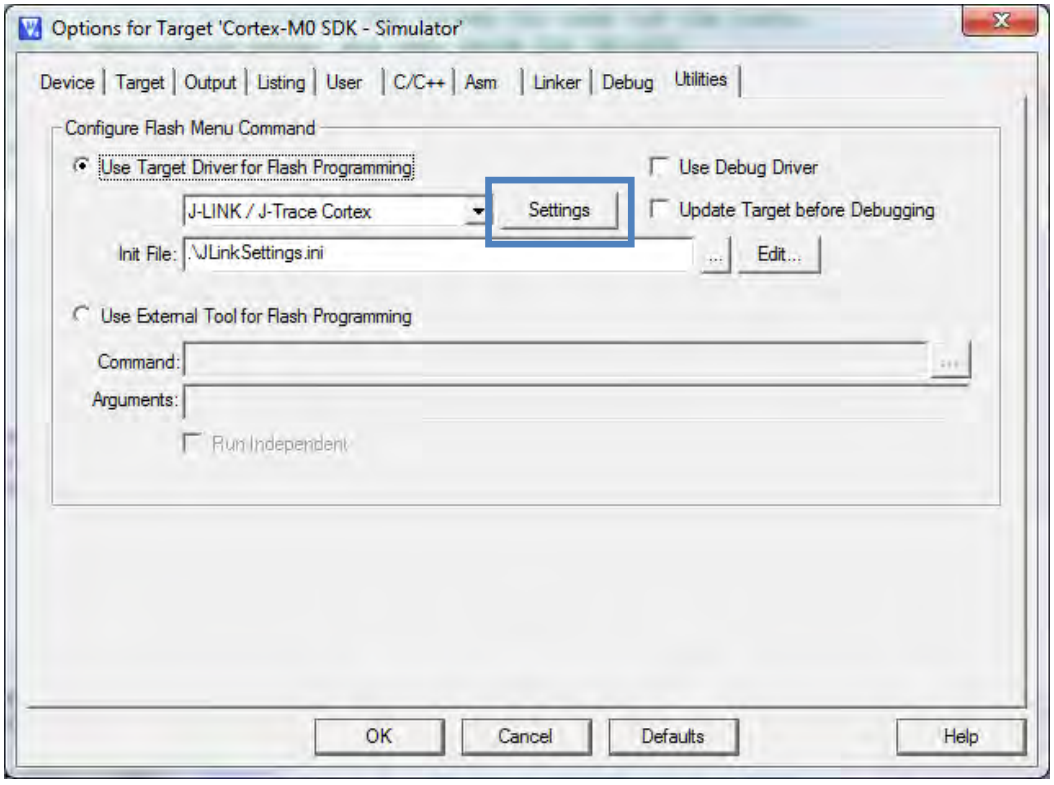


Open the proximity project example found in the Dialog SDK. The project is located in the following SDK folder (please use the latest release from Dialog website):

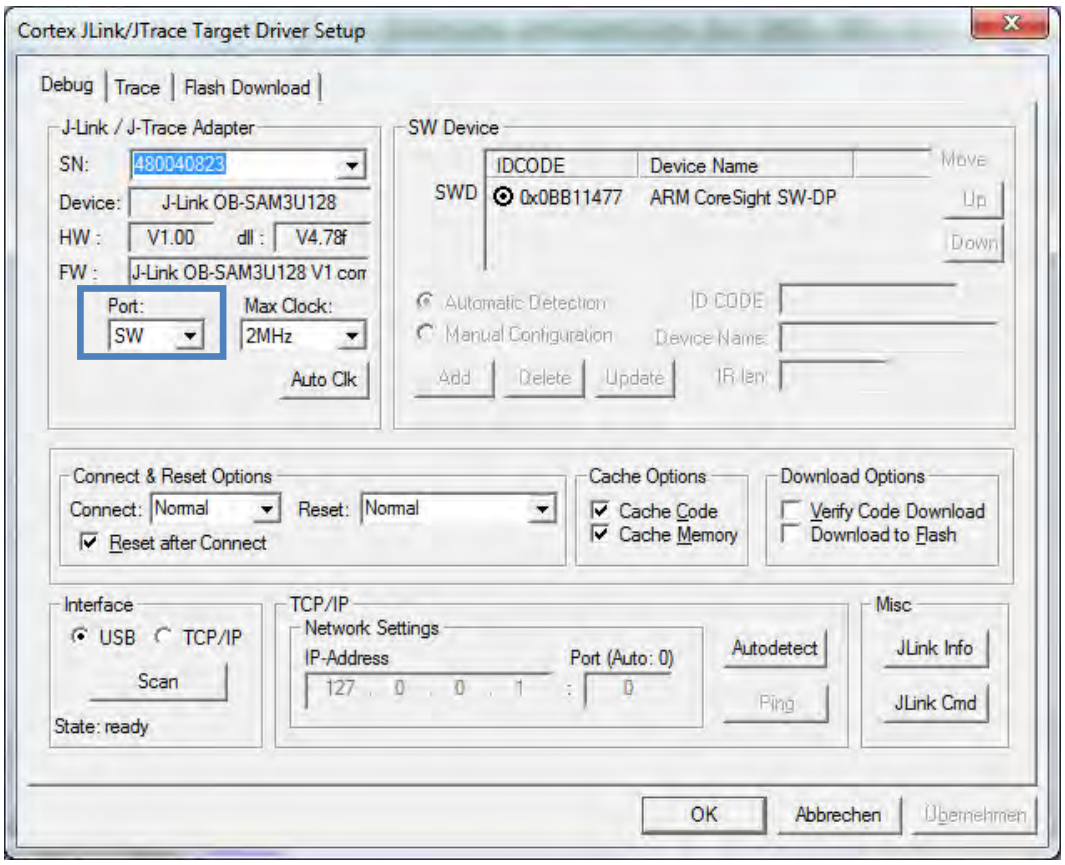
DA14580_SDK_3.0.2.1\dk_apps\keil_projects\proximity\monitor_fe_usb



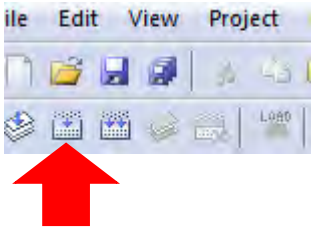
In the Keil compiler Configure the Flash Target (Flash >> Configure Flash Tools). Choose "Settings".



Click the Debug tab. Setup the Port in the J-Flash Settings to "SW". Press "OK".

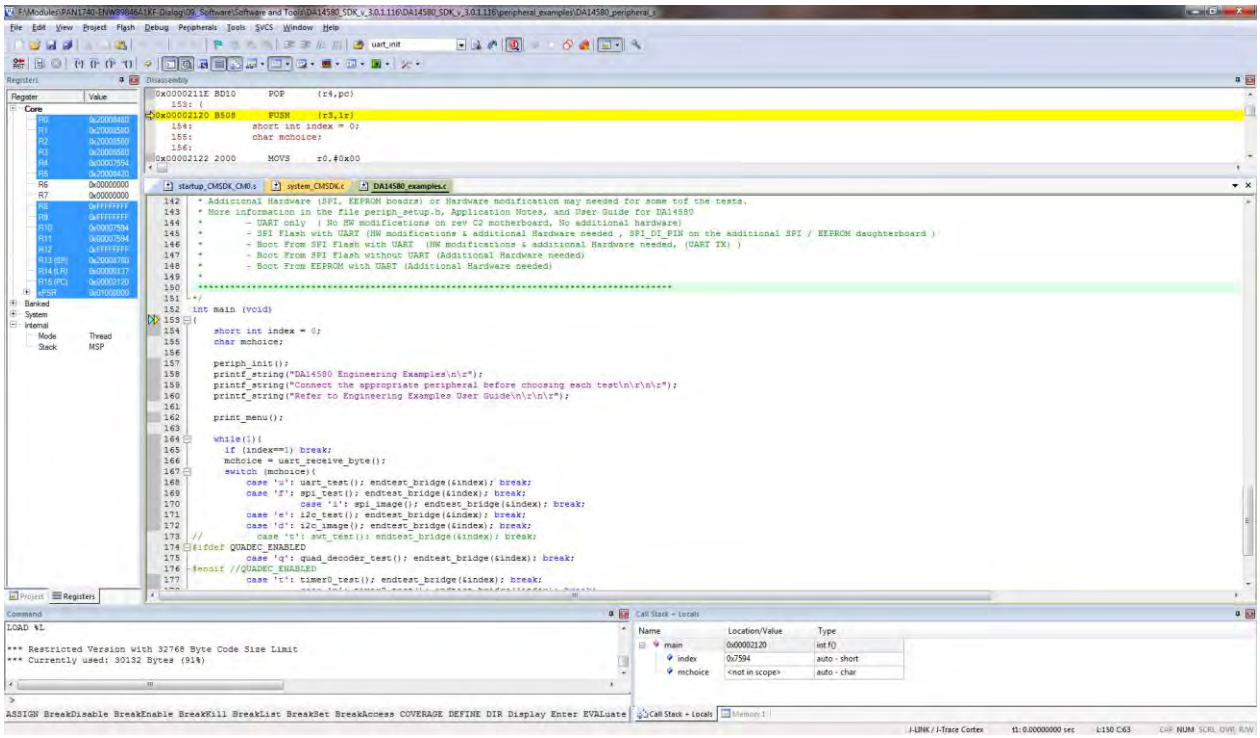


Build Target files (Press F7 or click on the build button) and run the debug session (Press "Ctrl"+F5 or click on the "Debug" button).



Build

Debug



The proximity project has now been compiled and downloaded into the RAM of the PAN1740ETU.

14. EXAMPLE PROXIMITY PROFILE WITH TWO PAN1740-USB DONGLES

Refer to the Basic Development Kit User Manual.

http://support.dialog-semiconductor.com/resources

Additional Tools:

Microsoft Visual C++ 2010 Express – Freeware Compiler

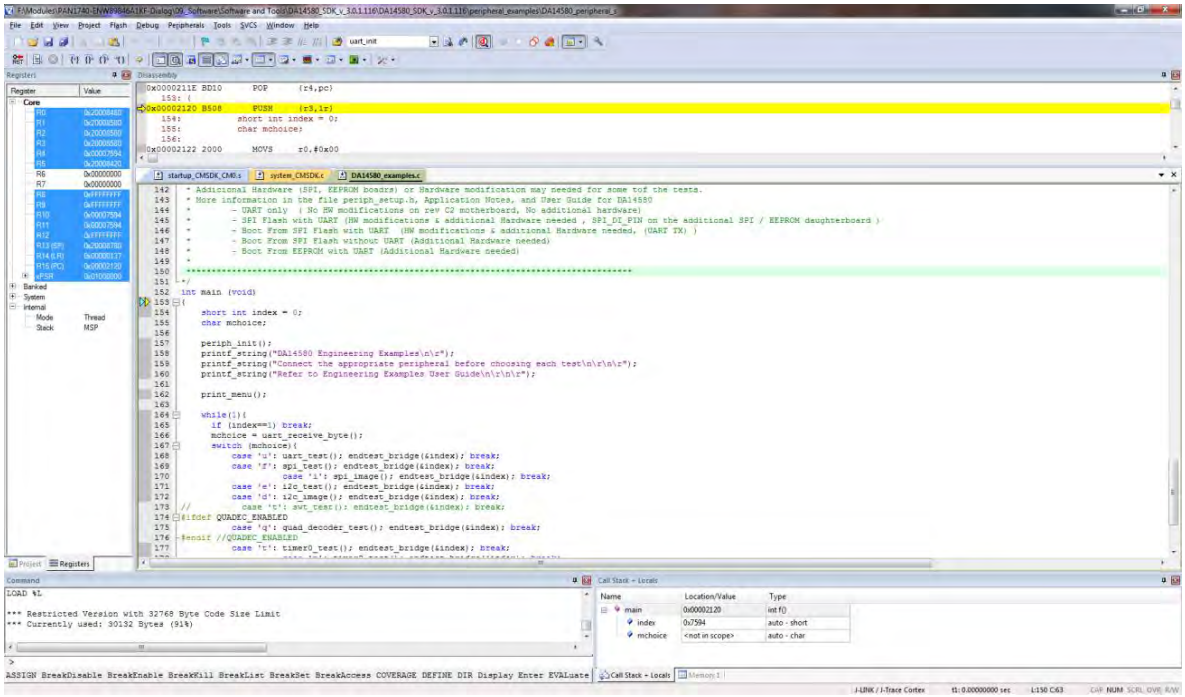
This example uses Dialog's SDK version 3.0.2.1

Receiver Configuration:

Download the proximity/monitor_fe_usb image into the dongle and refer to Section 13 for details (please use the latest release from Dialog website):

DA14580_SDK_3.0.2.1\dk_apps\keil_projects\proximity\monitor_fe_usb

Open Keil, compile this project and download the hex file. To download, either start and stop the debug mode or load the hex file with Connection Manager. For debug mode using the Keil compiler be sure to check the Configure Flash Tool setting described in section 13.

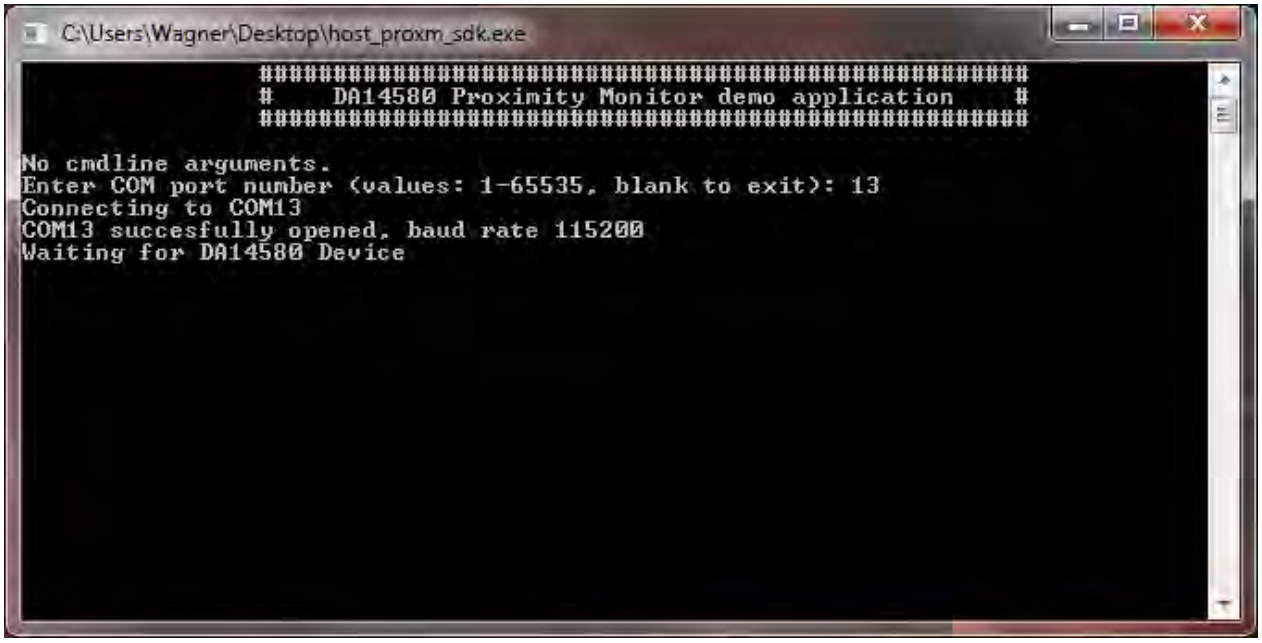


Note: Debug session must be stopped.

Open the Monitor Host Application folder in the SDK (please use the latest release from Dialog website).

DA14580_SDK_3.0.2.1\DA14580_SDK_3.0.2.1\host_apps\windows\proximity\monitor

Open the project file "host_proxm.sln" with Microsoft C++ compiler. Compile (Press F7") and run this SW (F5) and determine the correct COM port using Windows Device Manager and enter this port number in the Proximity Host application (DOS window).



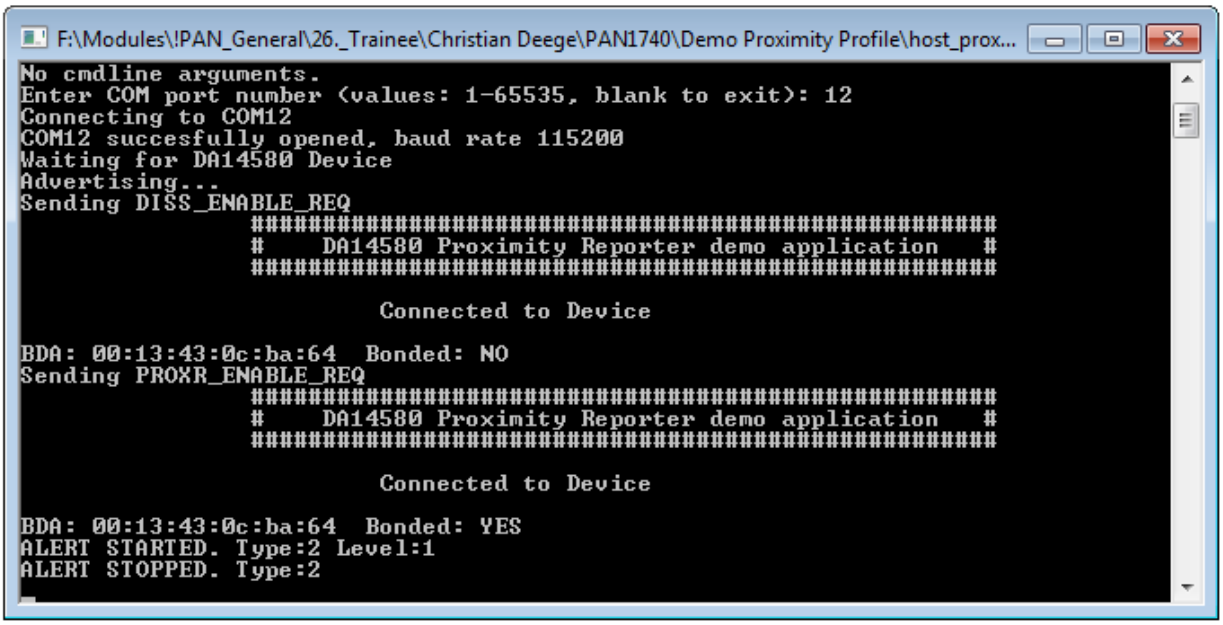
Transmitter Configuration:

Download the proximity/reporter_fe_usb image into the dongle: Open Keil, compile this project and download the hex file. To download, either start and stop the debug mode or load the hex file with Connection Manager.

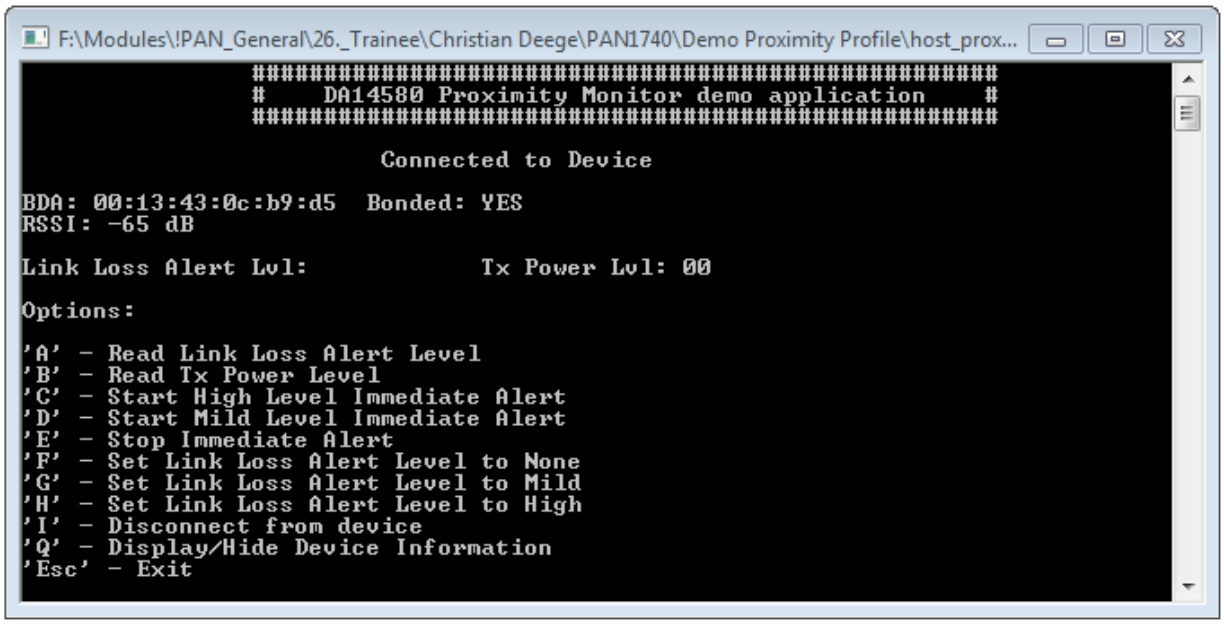
Open the reporter host application (please use the latest release from Dialog website)

DA14580_SDK_3.0.2.1\DA14580_SDK_3.0.2.1\host_apps\windows\proximity\reporter

Open the project file "host_proxr.sln" with Microsoft C++ compiler. Compile (Press F7") and run this SW (F5) and determine the correct COM port using Windows Device Manager and enter this port number in the proximity host application (DOS window).



Receiver Side will show connection status



The dongles are now connected with the proximity profile.

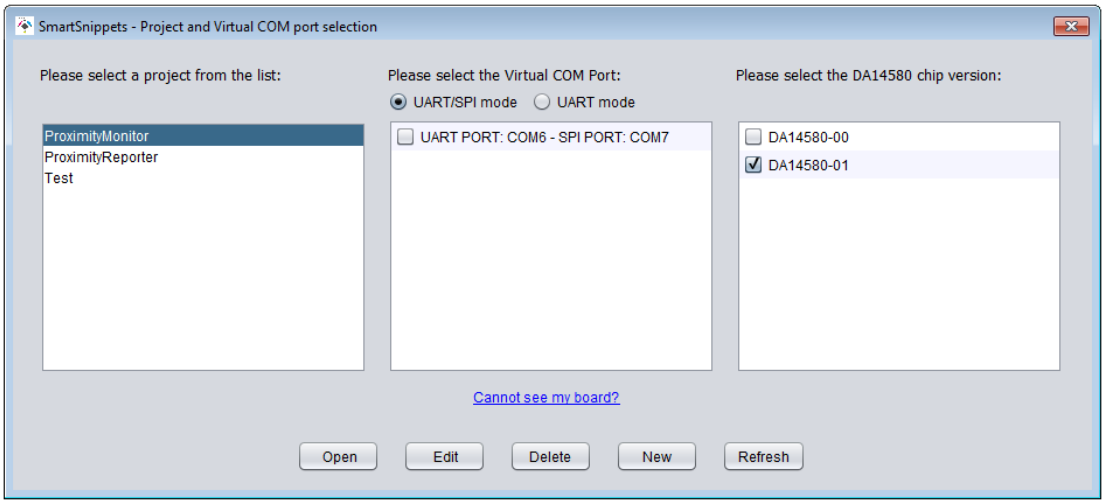
15. SMART SNIPPETS

The following description describes the structure and the usage of Smart Snippets in a nutshell. For additional information please select *Help → User Guide* in Smart Snippets after the Software has been fully launched.

15.1. Program Structure

15.1.1. Project and Port Selection

The first task after launching the application *Smart Snippets* is to select a project, the virtual COM port and the chip version to be able to control the development kit.



When the application launches the first time, there will be no projects to select. The user will have to create one by pressing the "New" button. The name should not contain any spaces or special characters.

After selecting a project, the user will have to select the DA14580 chip version and a virtual COM port which is assigned to the connected dialog development kit.

Finally, the user needs to press *Open* to establish a connection to the development kit. The software will show its default layout with a toolbar (Board Setup, UART Booter, Power Profiler, Sleep Mode Advisor, OTP Programmer, SPI Flash Programmer, EEPROM Programmer and SPotA) and a few of these tools in the center of the display. Every single tool can be enlarged to full screen if necessary by double clicking its header or the maximize button. The previously listed tools will be explained in the following sections.

15.1.2. Board Setup

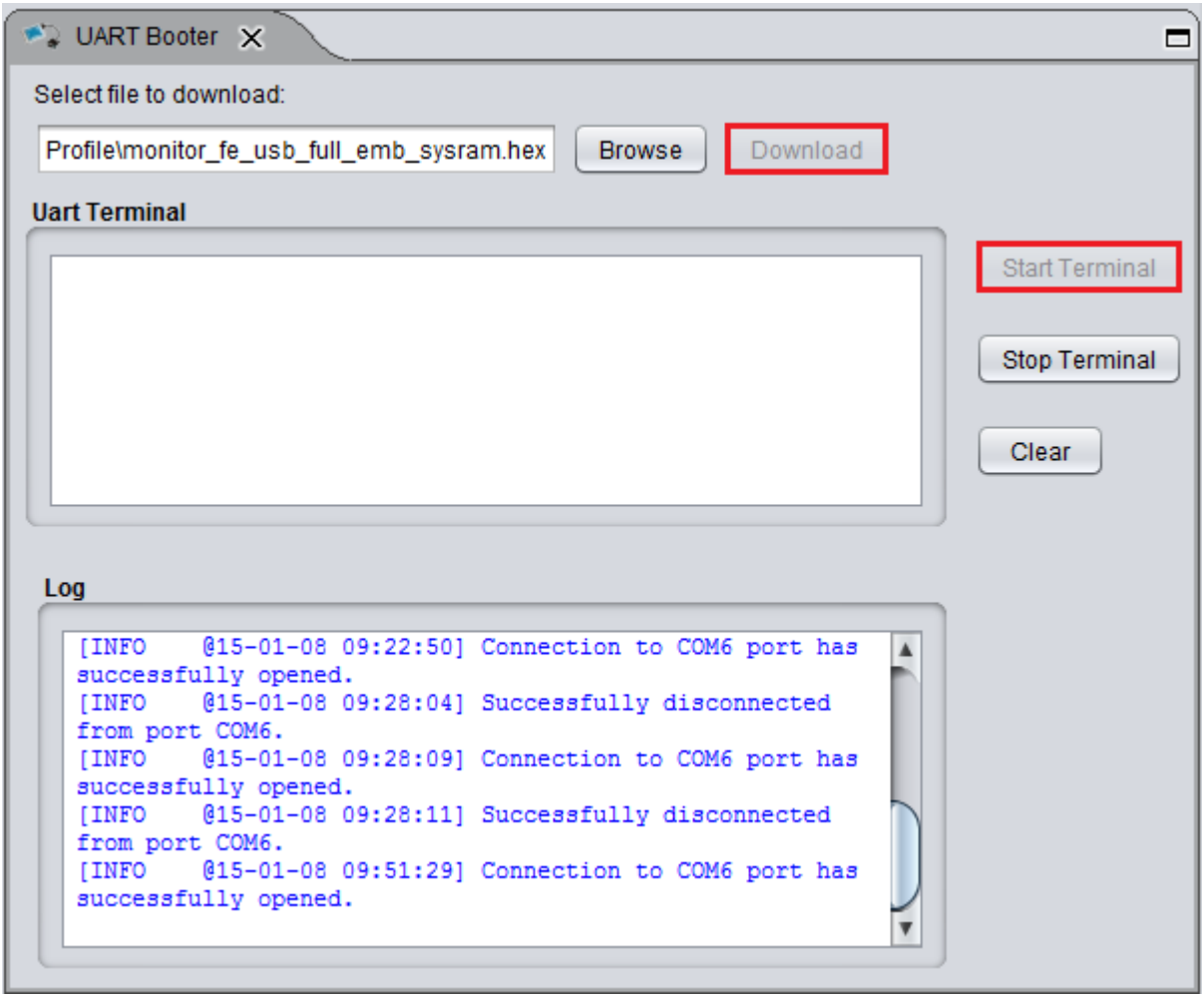
The tab *Board Setup* in the toolbox has to be used before any other toll as it establishes a communication with the development kit during the boot sequence and comes along with two lists.

The upper list contains UART ports with its baud rate, which connect the FTDI chip with the DA14580. The lower one selects the GPIO pin which enables 6.8V for OTP programming.

15.1.3. UART Booter

The UART Booter enables to download application code directly into the RAM of the DA14580 to test its behavior in terms of power consumption for instance.

The user has to select the desired code (.bin, hex. or .ihex), press *Download* and observe the log to handle the requested hardware reset by the user.

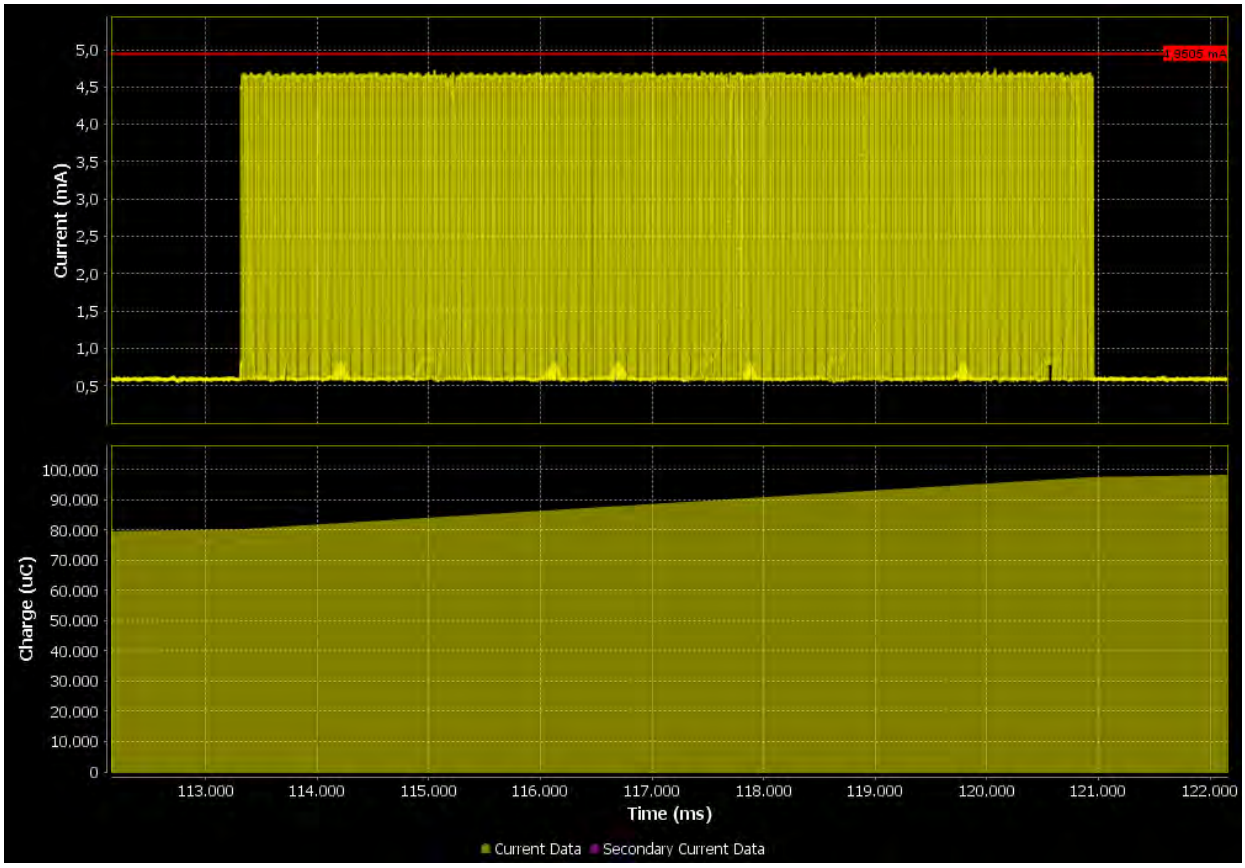


Furthermore, the UART Booter enables the opportunity to receive debugging information via UART. To activate this functionality the user will have to press *Start Terminal.* Please note that an activated UART connection disables the OTP connection with the result that the UART connection has to be closed to enable the OTP connection and vice versa.

15.1.4. Power Profiler

The tool Power Profiler enables the user to measure the power consumption of the desired application with all its functionality.

To start the measurement the user will have to press *Initialize* and *Start* after the initialization has successfully opened the COM-Port connection. A measurement example of the Bluetooth scanning process of the PAN1740 is depicted below.



On the right hand side, the tool provides the measurement information about *Peak Current (mA), Average Current (mA), Charge (µC)* and the *Sleep Mode* of the current measurement.

In addition, the tool provides the following control functionalities *Auto Trigger Mode* and *Auto Stop Mode.*

In *Auto Trigger Mode* the measurement process starts as soon as the current (mA) exceeds a user-definable threshold.

In *Auto Stop Mode* the measurement process stops automatically when the user-definable time (ms) is elapsed.

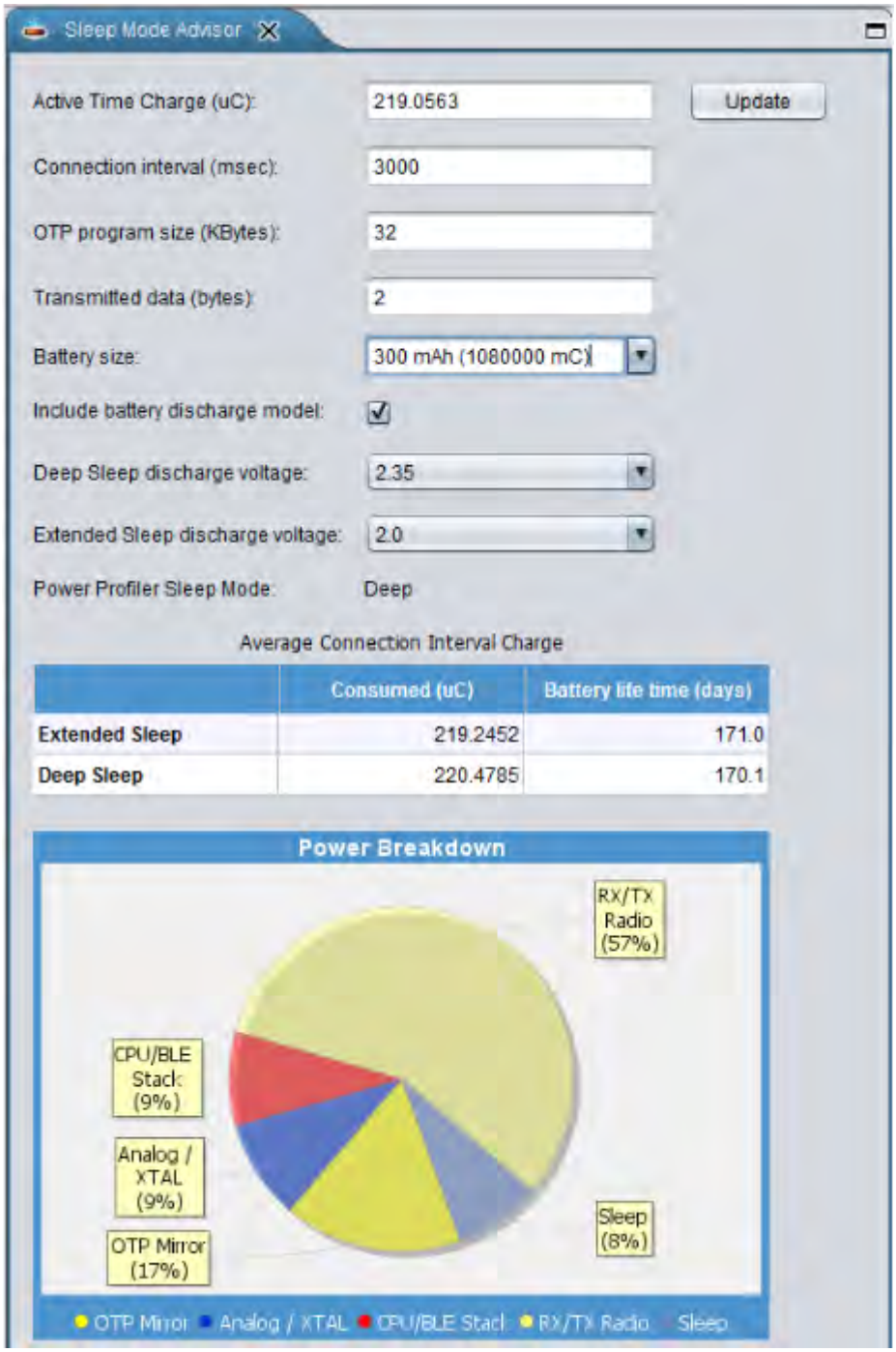
The previously depicted toolbar can be found in the top of Smart Snippets and enables the user to add measurements and markers, export/import data to/from csv files, clear secondary current data and to take snapshots of the Power Profiler chart (.png), which can be found in the path Dialog/Smart Snippets/Projects/*UserProject*.

For additional information about the Power Profiler Configuration Dialog, please refer to the Smart Snippet help (Help/User Guide/Power Profiler).

15.1.5. Sleep Mode Advisor

This tool uses the gathered consumption data from the Power Profiler and depicts the power consumption in the sleep modes *Deep Sleep* and *Extended Sleep* in a circle diagram. The

user will have to configure parameters such as battery size etc., to obtain reliable calculation results.



This previous figure from the Smart Snippets help shows an example that prefers the Extended Sleep Mode as the battery lasts 171 days compared to 170 days in deep sleep.

For detailed information, please refer to the Smart Snippets help (Help/User Guide/Sleep Mode Advisor).

15.1.6. OTP Programmer, SPI Flash Programmer and EEPROM Programmer

To burn the OTP Memory and the OTP Header on the DA14580 with a user-definable .hex/.ihex or .bin file, the tool OTP Programmer is used.

The OTP Image tab serves the purpose to read and burn the OTP Memory while the OTP Header tab is used to validate and burn the OTP header. The tab OTP NVDS works similar to the OTP Header and is used to burn the OTP NVDS memory block.

The SPI Flash Programmer enables the user to download an image file to the SPI flash memory of the DA14580. The functionality is similar to the OTP Programmer functionality, but the used firmware is different.

The EEPROM Programmer is used for downloading an image file to the DA14580 EEPROM Memory similar to the OTP and SPI Programmer functionality.

For detailed information of how to use the OTP Programmer, the SPI Flash Programmer and/or the EEPROM Programmer, please refer to the Smart Snippets help (Help/User Guide/OTP Programmer or SPI Flash Programmer or EEPROM Programmer).

15.1.7. SPotA (Software Patch over the Air)

The tab SPotA can be used to execute Software patches from changing a single variable in the code which resides in the SRAM to changing an instruction or data value read from the ROM used for protocol realization. Please note that a SPotA is only possible with an SPotA capable counterpart.

Please refer to the Smart Snippets help (Help/User Guide/SPotA) to understand how to establish a connection and patch the software.

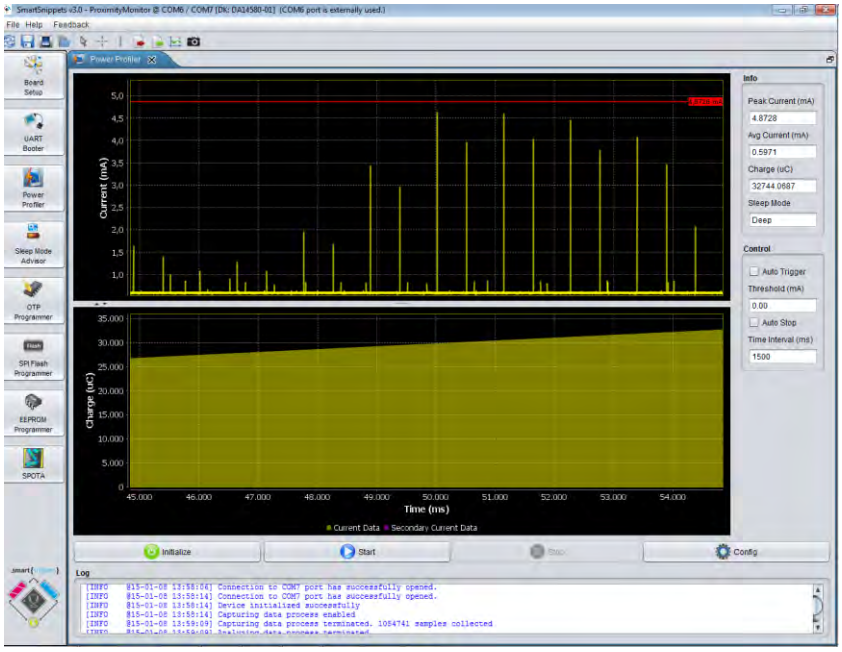
15.2. Application Demo: Proximity Profile

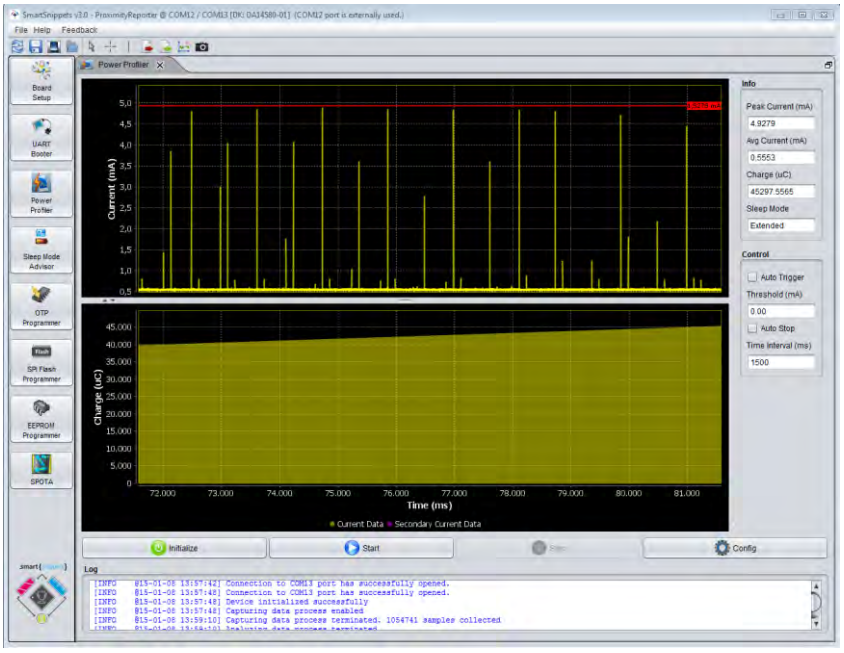
This application example demonstrates the usage of Smart Snippets based on the previously used proximity profile demo. Two DA14580 evaluation boards with PAN1740 adapter boards have been used which have both been connected to Smart Snippets.

One kit is loaded with the application code for the monitor (monitor_fe_usb_full_emb_sys-ram.hex) by the UART Booter, the other one with the application code for the reporter (reporter_fe_usb_full_emb_sysram.hex).

After that, the applications *host_proxm_sdk* and *host_proxr_sdk* have been launched and the respective COM ports have been entered and opened.

Once the devices have been connected, Smart Snippets enables live evaluation of the application code for instance by measuring the power consumption as depicted below.





16. DIALOG SERIAL PORT SERVICE (DSPS)

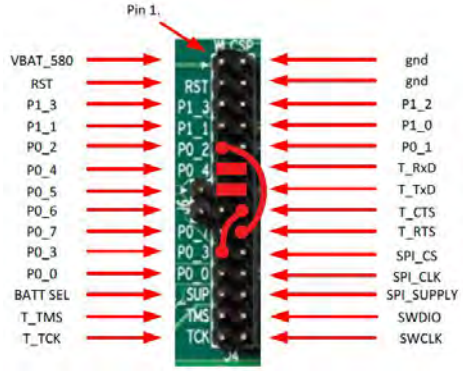
Dialog provides with DSPS its own BLE communication profile. Software has been developed for the Development Kit Pro and tablets resp. phones allowing a serial port to be emulated between development kits, handheld devices and PAN1740 USB Sticks. This software can be downloaded from dialogs support website. Following are a few different examples.

http://support.dialog-semiconductor.com/software-downloads/index

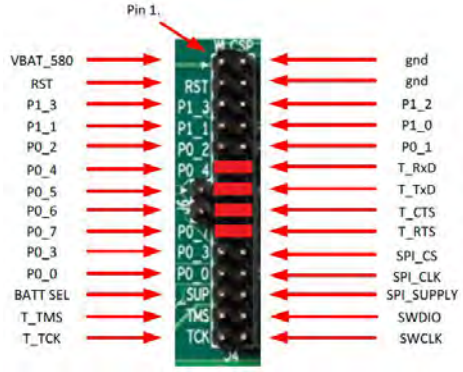
16.1. Pro Kit with PAN1740 Adapter Board - IPod

The following example shows an easy application of the SPS with Dialog's Pro Kit with a PAN1740 Adapter Board as device and an IPod running the DSPS application as host to exchange data. The kit needs to be set up as device using Dialogs SPS application example "*sps_device*". For that to happen, either the Connection Manager or Smart Snippets can be used.

The default pin assignment for the DSPS application software with hardware flow control on the Pro Kit is as follows.



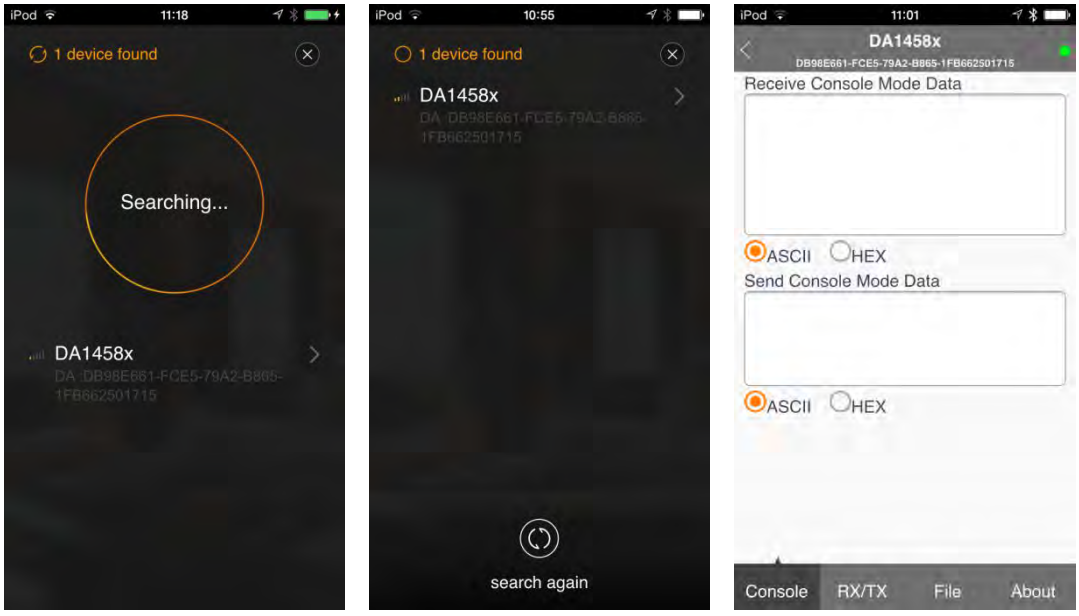
This pin assignment can be changed to any desired pin assignment in the application source code file *periph_setup.h*. To simplify the connection by using jumpers the following pin assignment is suitable.



This assignment does only require a simple amendment in the code.

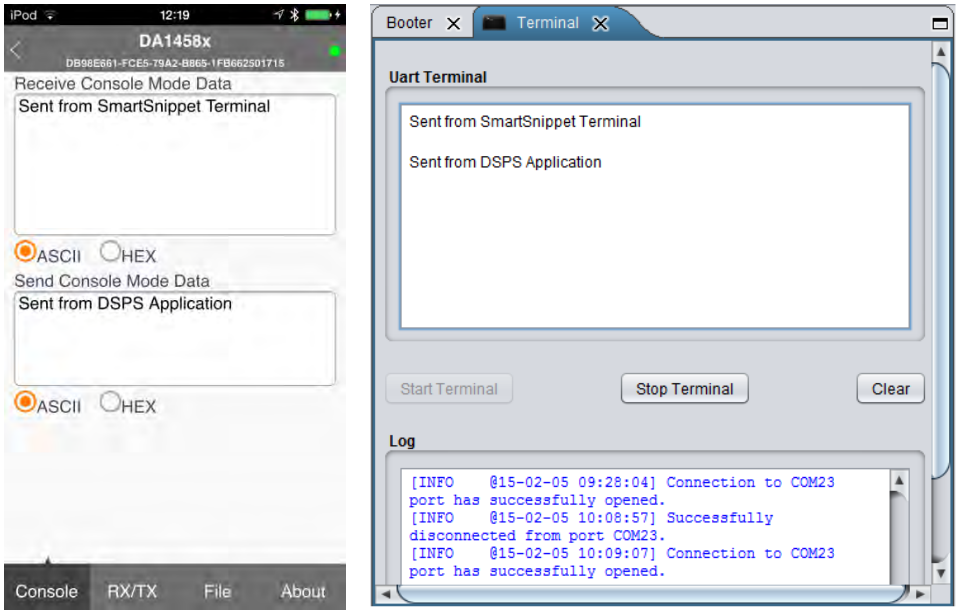
```
#define UART1_RTS_PORT  GPIO_PORT_0
//#define UART1_RTS_PIN   GPIO_PIN_3  // Previous configuration
#define UART1_RTS_PIN   GPIO_PIN_6    // New configuration

#define UART1_CTS_PORT  GPIO_PORT_0
//#define UART1_CTS_PIN   GPIO_PIN_2  // Previous configuration
#define UART1_CTS_PIN   GPIO_PIN_7    // New configuration
```

As soon as the kit has been set up as described in previous chapters, the user can launch the DSPS iOS application which starts scanning automatically for discoverable devices supporting the Serial Port Service. The user can connect to the desired device by selection. This procedure is depicted below.



After the connection has been established the user should see a tabbed view with a connection status bar at the top and the tabs *Console*, *RX/TX*, *File* and *About* at the bottom.

The *Console* tab enables to send data (ASCII or HEX) immediately after typing the data into the field *Send Console Mode Data* and receives data from the device, e.g. Smart Snippet UART Terminal, as shown below.



The tab *RX/TX* allows similar functionality as the *Console* tab, but enables to send data character by character manually or based on a cyclic sending interval which can be determined by the user.

The tab *File* enables to send data files instead of character strings. After the definition of a connection interval, the file directory of the device can be browsed to select the desired file.

For detailed information, please refer to the Dialog document UM-B-038.

16.2. Two Pro Kits with PAN1740 Adapter Board

Similar to the previous example it is also possible to use two of Dialog's Pro Kits with PAN1740 Adapter Boards to emulate a serial port. Note the HW flow control changes (jumper or source code) required as explained in chapter 16.1.

The first step is to set up one kit with the "*sps_device*" application software and the other one with the *"sps_host"* application software by downloading the .hex files onto the kits.

After that, the host should have discovered and connected to the device enabling the user to launch a terminal (for instance the Smart Snippet Terminal) and exchange data.

16.3. Two PAN1740 USB sticks

To establish a serial port connection with two PAN1740 USB sticks it is necessary to amend the *da14580_config.h* of the host and the device source code as follows.

Initially, it is mandatory to change the flow control mode from the default hardware flow control to software flow control.

```
/* Previous configuration */
/*
#define CFG_UART_HW_FLOW_CTRL
#undef CFG_UART_SW_FLOW_CTRL

#ifdef CFG_UART_HW_FLOW_CTRL
    #undef CFG_UART_SW_FLOW_CTRL
#endif
*/

/* New configuration */
#define CFG_UART_SW_FLOW_CTRL
#undef CFG_UART_HW_FLOW_CTRL
```

The next step is to disable the sleep mode, as this is not fully supported for software flow control.

```
/*Sleep modes*/
/* Previous configuration
//#define CFG_EXT_SLEEP
*/

// New configuration
#undef CFG_EXT_SLEEP
```

After that, the amended source codes have to be compiled and the output .hex files have to be downloaded onto the sticks by using Dialogs *Connection Manager* or *Keil uVision*. Please note that it is not possible to run the sticks with Dialogs Smart Snippets.

After the application software has been successfully downloaded to the sticks, please launch two terminals (e.g. Tera Term) with the corresponding COM port and the following settings:

Baud Rate: 115200

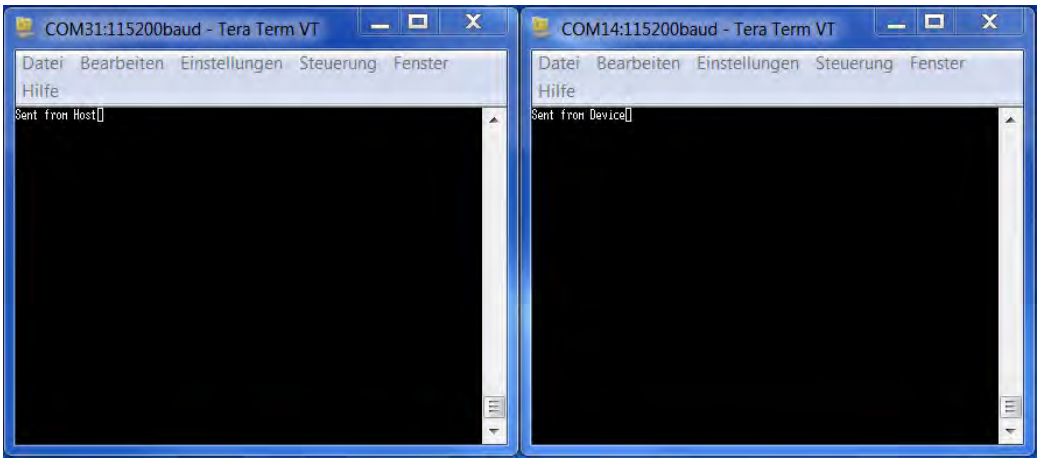
Data Bits: 8

Stop Bits: 1

Parity: None

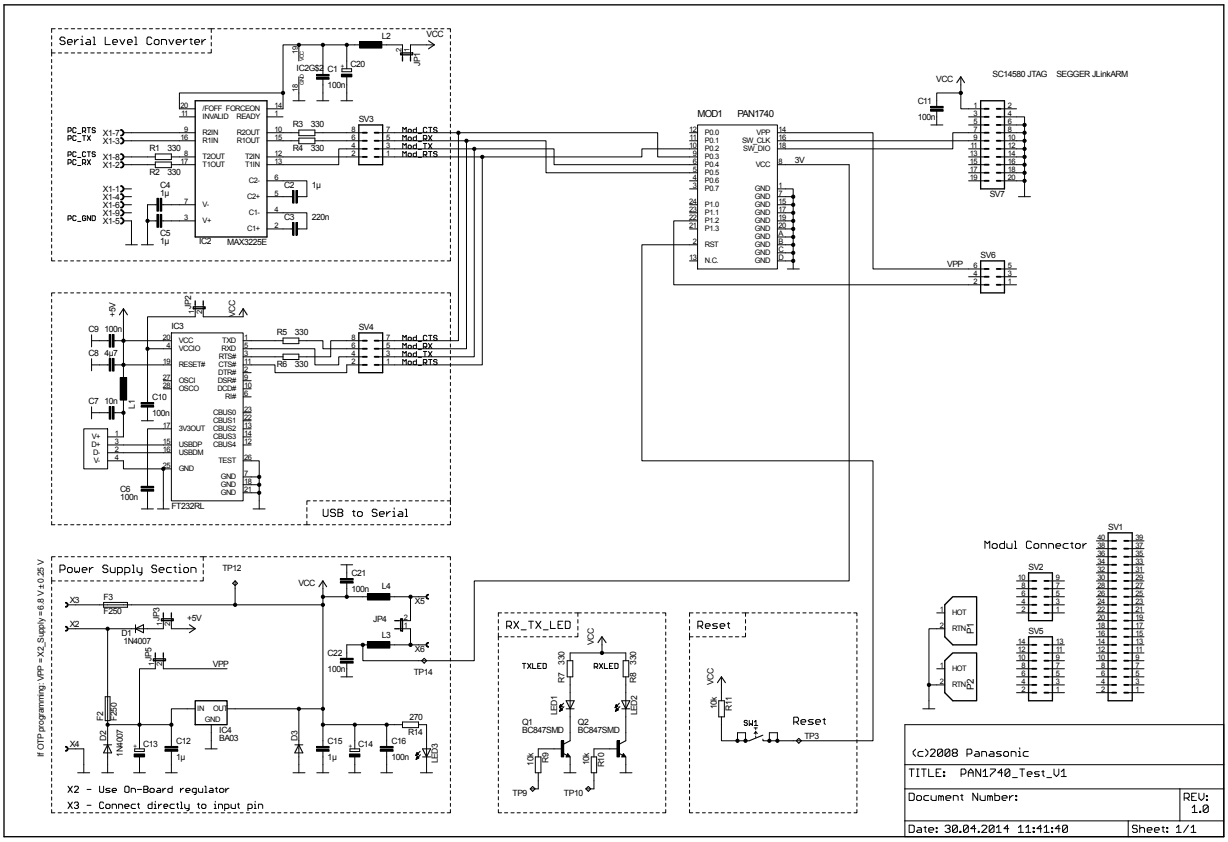
Flow Control: Xon/Xoff

Expecting the previous steps to be successful, the data exchange can be started. The data transmission happens immediately, so that one terminal window will stay empty for unidirectional data transfer as depicted below.



17. PRODUCTION TOOLS

To program the PAN1740 in production a J-Link programmer and the 6.8V programming voltage on the VPP input pin is required. Here is an example schematic for a programming jig.



For more details on programming the OTP refer to the Smart Snippets documention and Dialog's programming guide located on Dialog's website.

Note: The crystal frequency register and flag as well as the Bluetooth MAC address is already burned.

18. HISTORY FOR THIS DOCUMENT

Revision	Date	Modification / Remarks
0.1	02.04.2014	Initial Preliminary Release.
1.0	04.06.2014	Added USB dongle and Connection Manager description. Release Version.
1.1	25.06.2014	Added Keil Project example.
1.2	17.07.2014	Added Proximity example for connecting two PAN1740 USB dongles.
1.3	05.09.2014	Added information about production tools.
1.4	26.09.2014	Editoral changes.
1.5	13.01.2015	Added chapter about Mother Board, Adapter Board and Smart Snippets
1.6	06.02.2015	Added chapter about SPS
1.7	16.02.2015	Added software flow control source code amendments in SPS chapter
1.8	30.07.2015	Added chapter Beacon
1.9	17.08.2015	Added a note about beacon programming with FFC
2.0	01.10.2015	Added the schematic for the Beacon

19. RELATED DOCUMENTS

[1] PAN1740 Datasheet
http://pideu.panasonic.de/files/Documents/WM%20Documents/PAN1740/PAN1740_Datasheet.pdf

20. GENERAL INFORMATION

© Panasonic Electronic Devices Europe GmbH 2010.
All rights reserved.

Panasonic does not warranty and accepts no liability for the information contained herein. The information contained in this document is subject to change without notice. Modules containing "ES" in the series number are Engineering Samples -- i.e. PANxxxxES. This means, the design of this product is not yet concluded. Engineering Samples may be partially or fully functional, and there may be differences to be published Data Sheet.
Engineering Samples are not qualified and are not to be used for reliability testing or series production.

Disclaimer:
Customer acknowledges that samples may deviate from the Data Sheet and may bear defects due to their status of development and the lack of qualification mentioned above.
Panasonic rejects any liability or product warranty for Engineering Samples. In particular, Panasonic disclaims liability for damages caused by

- the use of the Engineering Sample other than for Evaluation Purposes, particularly the installation or integration in another product to be sold by Customer,
- deviation or lapse in function of Engineering Sample,
- improper use of Engineering Samples.

Panasonic disclaimes any liability for consequential and incidental damages.
In case of any questions, please contact your local sales partner or the related product manager.

21. FCC WARNING

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.
The FCC and other regulatory certifications for the PAN1740 will be published in the PAN1740 Datasheet.

22. LIFE SUPPORT POLICY

This Panasonic product is not designed for use in life support appliances, devices, or systems where malfunction can reasonably be expected to result in a significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Panasonic customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Panasonic for any damages resulting.