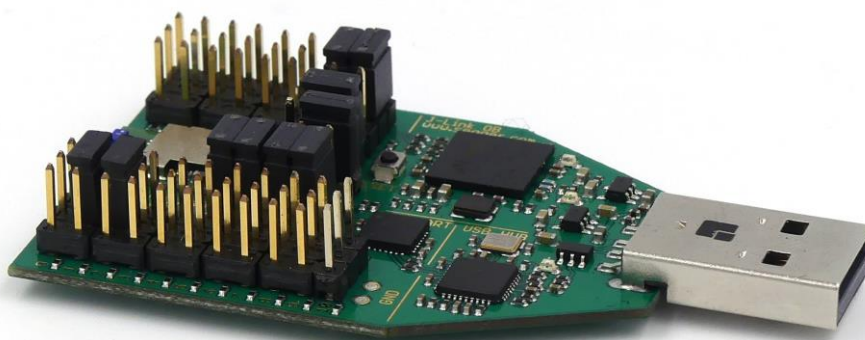


PAN1762

Bluetooth Low Energy Module

Software Guide

Rev. 1.4



By purchase of any of the products described in this document the customer accepts the document's validity and declares their agreement and understanding of its contents and recommendations. Panasonic Industrial Devices Europe GmbH (Panasonic) reserves the right to make changes as required at any time without notification.

© Panasonic Industrial Devices Europe GmbH 2019.

This document is copyrighted. Reproduction of this document is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Do not disclose it to a third party.

All rights reserved.

This Software Guide does not lodge the claim to be complete and free of mistakes.

The information contained herein is presented only as guidance for Product use. No responsibility is assumed by Panasonic for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.

Description of hardware, software, and other information in this document is only intended to illustrate the functionality of the referred Panasonic product. It should not be construed as guaranteeing specific functionality of the product as described or suitable for a particular application.

Any provided (source) code shall not be used or incorporated into any products or systems whose manufacture, use or sale is prohibited under any applicable laws or regulations.

Any outlined or referenced (source) code within this document is provided on an "as is" basis without any right to technical support or updates and without warranty of any kind on a free of charge basis according to § 516 German Civil Law (BGB) including without limitation, any warranties or conditions of title, non-infringement, merchantability, or fitness for a particular purpose. Customer acknowledges that (source) code may bear defects and errors.

The third-party tools mentioned in this document are offered by independent third-party providers who are solely responsible for these products. Panasonic has no responsibility whatsoever for the performance, product descriptions, specifications, referenced content, or any and all claims or representations of these third-party providers. Panasonic makes no warranty whatsoever, neither express nor implied, with respect to the goods, the referenced contents, or any and all claims or representations of the third-party providers.

To the maximum extent allowable by Law Panasonic assumes no liability whatsoever including without limitation, indirect, consequential, special, or incidental damages or loss, including without limitation loss of profits, loss of opportunities, business interruption, and loss of data.

Table of Contents

1	About This Document.....	4
1.1	Purpose and Audience	4
1.2	Revision History	4
1.3	Use of Symbols	4
1.4	Related Documents	5
2	Introduction	6
2.1	Architecture	6
2.2	Device Drivers	7
3	Preparations.....	8
3.1	Unzipper Tool	8
3.2	Directory Layout	10
3.3	Manual Installation.....	10
3.4	Further Readings.....	11
4	Operating Modes.....	12
4.1	Host Mode	12
4.2	AT Command Mode.....	12
4.3	Standalone Mode	13
4.4	Mode Selection.....	13
5	SPP over Bluetooth LE Profile.....	15
6	Bluetooth Low Energy Basics	16
7	Host Mode	17
7.1	PC Application	17
7.2	COM Port Handling.....	20
8	AT Command Mode	21
8.1	Installing EasyStandalone	21
8.2	Programming the Firmware	22
8.3	PC Application	24
9	Standalone Mode	27
9.1	Application	27
9.2	Installing EasyStandalone	28
9.3	Programming the Firmware	28
9.4	Testing the Setup	30
10	Development	32
10.1	Host Mode	32
10.2	AT Command Mode.....	33
10.3	Standalone Mode	35
11	Appendix	42
11.1	Changes to the Toshiba Bluetooth SDK	42
11.2	Contact Details	43

1 About This Document

1.1 Purpose and Audience




This Software Guide is intended as a quick start guide and explains how to setup the PAN1762 USB stick, describes the basic usage modes and gives an introduction to the software that is provided.

The document is intended for software engineers.

1.2 Revision History

Revision	Date	Modifications/Remarks
1.0	2018-11-22	Initial release
1.1	2018-12-10	Updated to Toshiba Bluetooth SDK 4.1.1
1.2	2019-01-14	Added chapter "Pack-File Installation". Fixed connections problems with Android and iOS.
1.3	2019-03-07	Updated to Toshiba Bluetooth SDK 4.2.0. Added support for AT Command Firmware. Updated disclaimer.
1.4	2019-04-01	Updated formatting and structure changes. Editorial updates.

1.3 Use of Symbols

Symbol	Description
	Note Indicates important information for the proper use of the product. Non-observance can lead to errors.
	Attention Indicates important notes that, if not observed, can put the product's functionality at risk.
	Tip Indicates useful information designed to facilitate working with the PAN1762.
⇒ [chapter number] [chapter title]	Cross reference Indicates cross references within the document. Example: Description of the symbols used in this document ⇒ 1.3 Use of Symbols.
✓	Requirement Indicates a requirement that must be met before the corresponding tasks can be completed.

Symbol	Description
→	Result Indicates the result of a task or the result of a series of tasks.
This font	GUI text Indicates fixed terms and text of the graphical user interface. Example: Click Save .
<code>This font</code>	File names, messages, user input Indicates file names or messages and information displayed on the screen or to be selected or entered by the user. Examples: <code>pan1762.c</code> contains the actual module initialization. The message <code>Failed to save your data</code> is displayed. Enter the value <code>Product 123</code> .
Key	Key Indicates a key on the keyboard, e.g. F10 .

1.4 Related Documents

[1] PAN1762 Design Guide

Please refer to the Panasonic website for more information as well as related documents

⇒ [11.2.2 Product Information](#).

2 Introduction

The PAN1762 USB stick is a development platform for the Bluetooth® Low Energy (LE) PAN1762 module. It is based on the Bluetooth Low Energy chipset TC35680 from Toshiba.

For further information please visit <https://toshiba.semicon-storage.com/eu/product/wireless-communication/bluetooth/tc35680.html>.

Toshiba provides a Bluetooth software development kit (SDK) that is available after registration. For further information please visit <https://apps.toshiba.de/web/SDKRegistration/>.

The Toshiba Bluetooth SDK includes a straight-forward serial cable replacement protocol called “SPP over Bluetooth LE profile”.

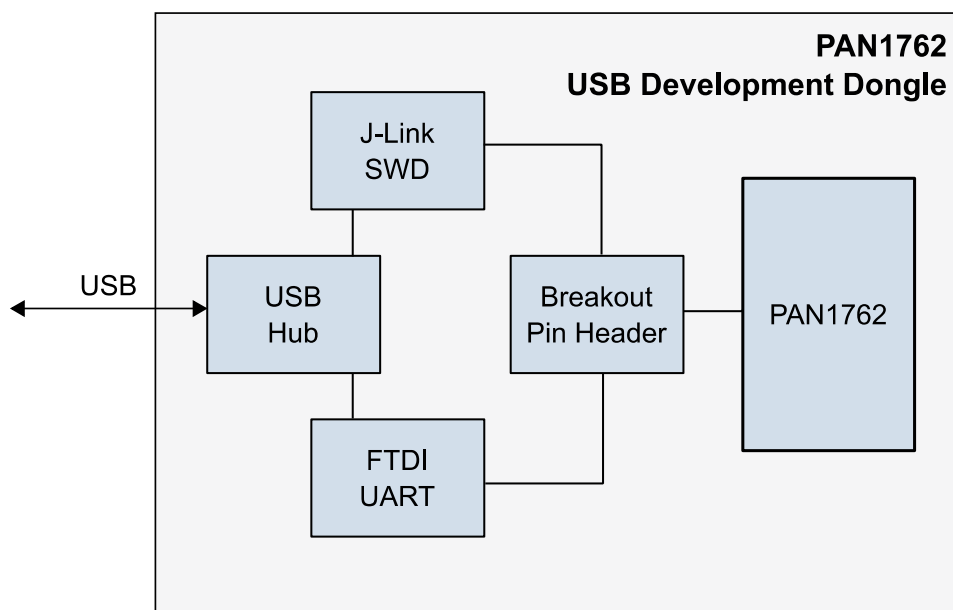
This Software Guide explains the usage of the SPP over Bluetooth LE profile in the system configurations that are possible with the PAN1762 module in order to give customers a quick start for their designs.

The guide refers to the software package release version 1.3.

To download the latest version please visit

<https://pideu.panasonic.de/produkte/bluetooth/pan1762-bluetooth-low-energy.html>

2.1 Architecture

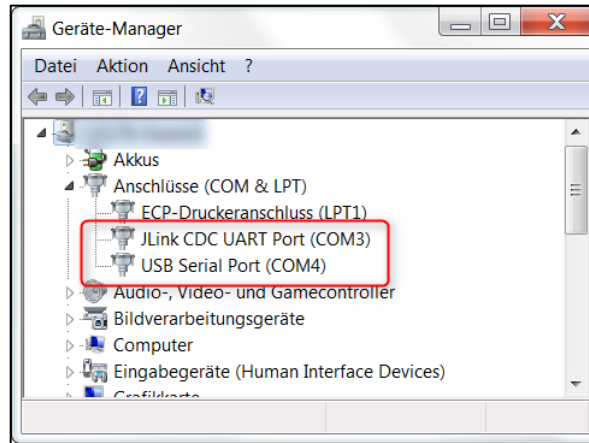


The PAN1762 USB stick consists of the following components:

- PAN1762 module
- PIN header breakout section
- Segger J-Link SWD debugger
- FTDI USB UART

It might be necessary to install drivers for some of the components.

Please note that both the “FTDI USB UART” and the “Segger J-Link” SWD debugger will provide a COM port to the system.



Only the COM port labeled as “USB Serial Port (COM4)” can be used to interface directly with the PAN1762 module.

2.2 Device Drivers

2.2.1 FTDI USB UART

Depending on the operating system that is used, drivers for the “FTDI USB UART” might not be installed automatically.

Having the drivers installed correctly is mandatory for all the examples mentioned in this guide.

If in doubt, please check the FTDI website and install the drivers manually.

For further information please visit <http://www.ftdichip.com/Drivers/VCP.htm>.

2.2.2 “Segger J-Link” SWD Debugger

Depending on the operating system that is used, drivers for “Segger J-Link” SWD debugger might not be installed automatically.

Having the drivers installed correctly is not strictly mandatory for the basic examples mentioned in this guide.

If in doubt, please check the “Segger” website and install the drivers manually.

For further information please visit <https://www.segger.com/downloads/jlink/>.

3 Preparations

All examples shown in this document are based on the Bluetooth SDK version 4.2.0. It is mandatory to have that version downloaded before proceeding.

Toshiba provides a Bluetooth SDK that is available after registration.

For further information please visit <https://apps.toshiba.de/web/SDKRegistration/>.

After registration the Toshiba Bluetooth SDK can be downloaded from the “Developer Zone”: <https://apps.toshiba.de/extranet/display/TBTSW/>.

The sample applications refer to certain files from the Toshiba Bluetooth SDK, so the SDK must be present in a specific location.

When the PAN1762 Software Guide ZIP file is unpacked, the following directories are present in the current directory:

- **Binaries:** contains pre-compiled applications for immediate evaluation
- **Configurations:** contains the source code for all applications

Furthermore the following files are present in the current directory:

- `PAN1762 Software Guide.pdf`: this document
- `pan1762_unzipper.exe`: unzipper tool
- `bluetooth_sdk_differences.zip`: differences to Toshiba Bluetooth SDK version 4.2.0

3.1 Unzipper Tool

Setting up the correct directory layout is tedious and error prone. In order to facilitate this process, the tool **unzipper** is available that will execute all necessary steps in the correct order.



For the rest of this document it is assumed that the installation of the software package was done to `c:\pan`.

If a different path was used, please make sure to adopt the path names in the examples accordingly.

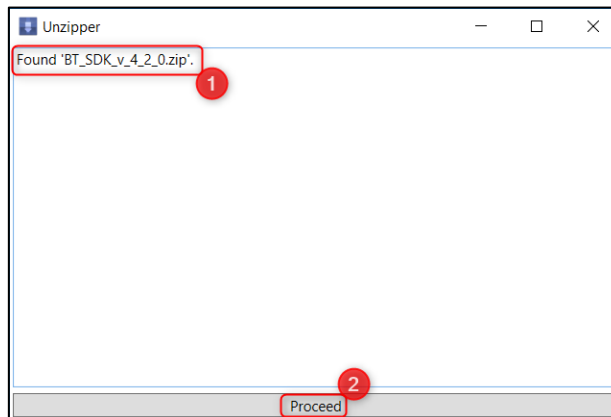
The following requirements must be met to work correctly with **unzipper**:

- ✓ `BT_SDK_v_4_2_0.zip` (Toshiba Bluetooth SDK version 4.2.0) is present in the same directory additionally

1. Execute `pan1762_unzipper.exe`.

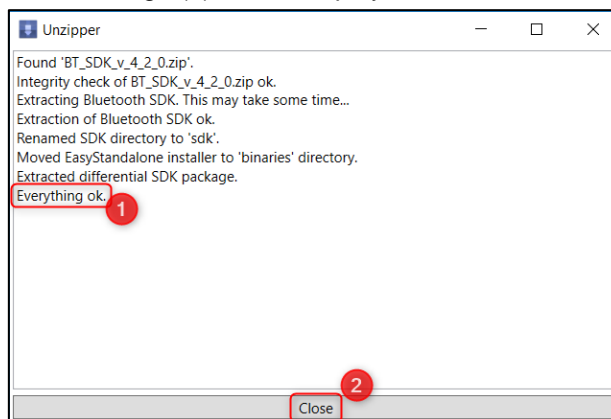
If any of the necessary files is not present, the tool will immediately display an error message.

→ This message (1) will be displayed.



2. Click **Proceed** (2).

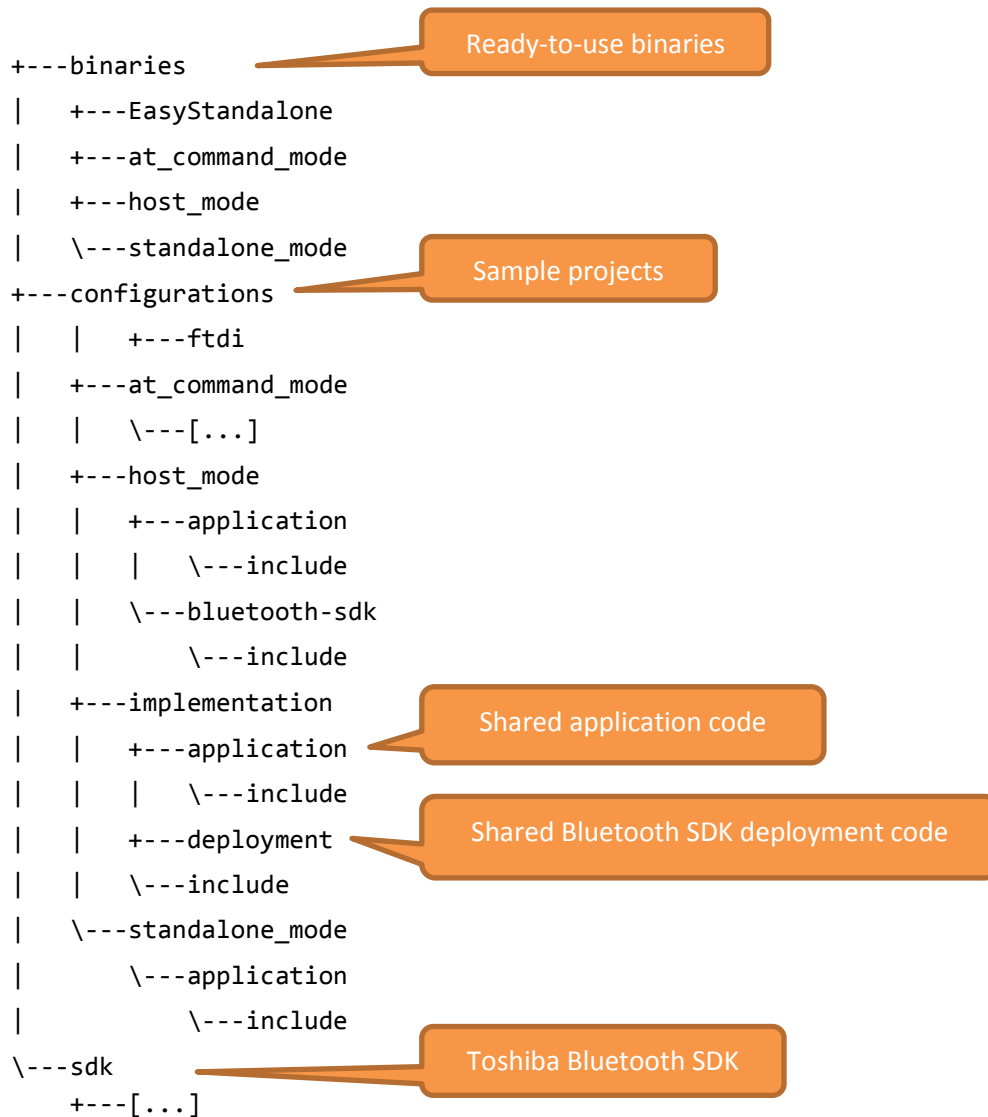
→ This message (1) will be displayed.



3. Click **Close** (2).

3.2 Directory Layout

After **unzipper** has finished the following directory layout is present:



In the installation process the Bluetooth SDK is slightly modified for some of the use cases. The changes are explained in [⇒ 11.1 Changes to the Toshiba Bluetooth SDK](#).

3.3 Manual Installation

If the tool **unzipper** cannot be used, the following steps have to be done manually to create the necessary directory structure.

1. Create a destination directory and change into it.
2. Unzip the ZIP file `PAN1762 Software Guide` into the destination directory.
3. Unzip the Toshiba Bluetooth SDK `BT_SDK_v_4_2_0.zip` in that directory.

4. Rename the directory `BT_SDK_v_4_2_0` to `sdk`.
5. Copy the directory `sdk/at_command/easy_standalone` to `binaries/EasyStandalone`.
6. Change into the subdirectory. `sdk`
7. Unzip `bluetooth_sdk_differences.zip` from parent directory over existing files.

3.4 Further Readings

This document mostly explains the specific features of the PAN1762 module.

If you need more details concerning the underlying chipset, please refer to the documentation from the Toshiba Bluetooth SDK.

The Bluetooth SDK contains extensive documentation, for example the “Bluetooth SDK developers guide”.

All documentation from Toshiba can be found in the Bluetooth SDK directory `BT_SDK_v_4_2_0\documentation`.

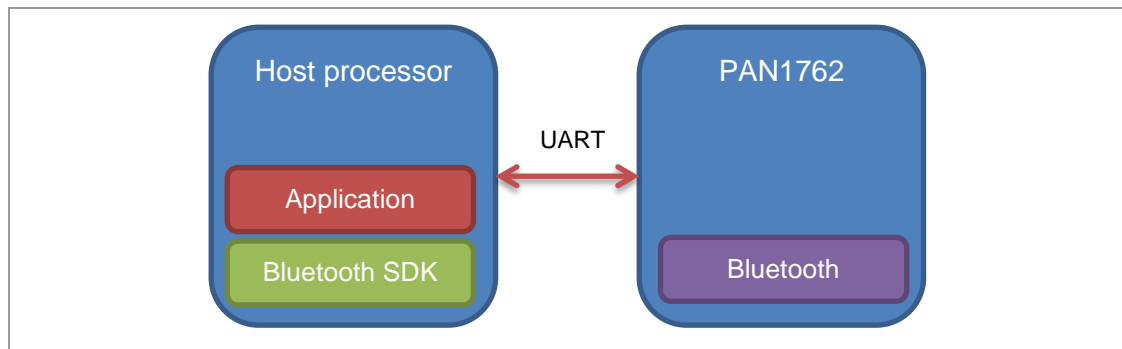
4 Operating Modes

The PAN1762 module allows different operating modes which make different product configurations possible.

The main operating modes are:

- Host mode
- Standalone mode

4.1 Host Mode

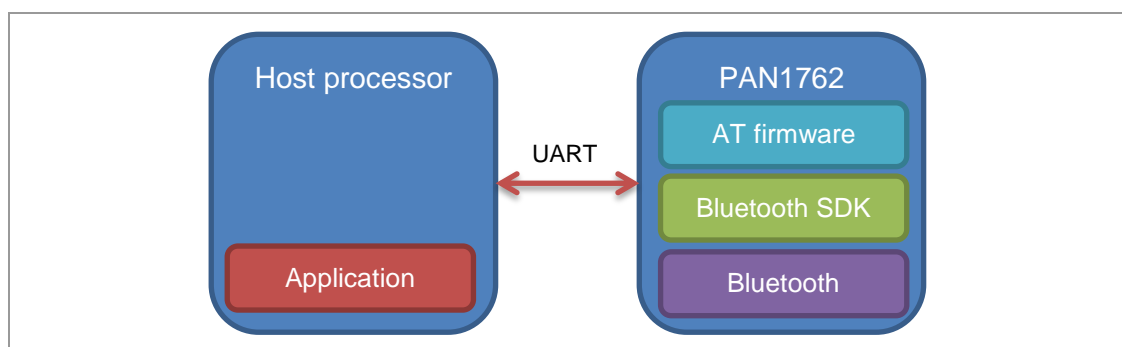


The host mode operating mode can be used in a dual chip product configuration, where the chips are connected via UART.

The PAN1762 module provides the necessary Bluetooth functionality to a host processor, which needs to run the Toshiba Bluetooth SDK. No software needs to be run on the PAN1762 module itself.

This setup gives the most flexibility to the system designer, but the host processor needs the necessary resources to run the Toshiba Bluetooth SDK.

4.2 AT Command Mode



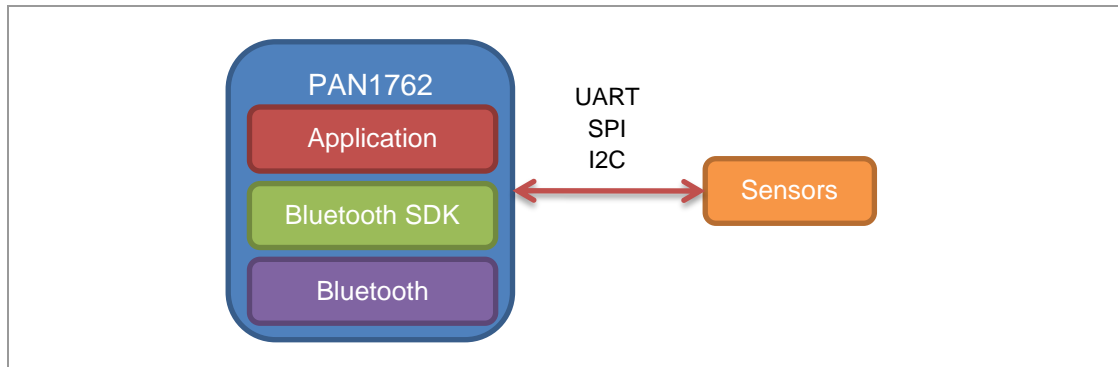
The AT command mode operating mode can be used in a dual chip product configuration as well and again the chips are connected via UART.

The PAN1762 module provides the necessary Bluetooth functionality to a host processor as well, but in a simplified form by using an “AT command” like interface.

The Toshiba Bluetooth SDK is run on the PAN1762 module inside a special AT command firmware.

This setup gives good flexibility to the system designer and the host processor does not need many resources.

4.3 Standalone Mode



The standalone operating mode is a single chip configuration, where all processing is done by the PAN1762 module itself.

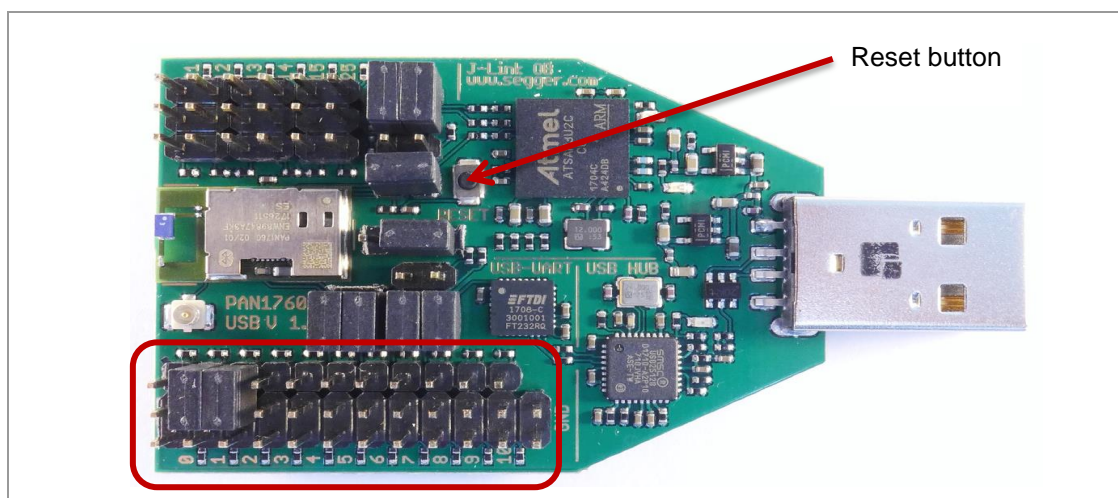
The Toshiba Bluetooth SDK as well as the customer application is run in the context of the Bluetooth processing.

The application has full access to the built-in peripherals and can use them for data input or output.

This setup provides the most cost-efficient solution for a new Bluetooth enabled product.

4.4 Mode Selection

The operating mode of the PAN1762 module is determined by two GPIO jumpers in the PIN header breakout section.



Regardless of the mode to be used GPIO jumper 2 must always be set in the upward position, connecting the GPIO to GND.



When the mode has been changed the PAN1762 module needs to be reset: Press the “reset” button.

4.4.1 Host Mode

0	1	2	3	4	5	6	7	8	9	10	

- Both GPIO jumpers must be set to the upward position, connecting the GPIOs to GND

4.4.2 Standalone Mode

0	1	2	3	4	5	6	7	8	9	10	

- GPIO jumper 1 must be set to the downward position, connecting GPIO jumper 1 to VCC
- GPIO jumper 2 must be set to the upward position, connecting GPIO jumper 2 to GND

4.4.3 AT Command Mode

0	1	2	3	4	5	6	7	8	9	10	

- GPIO jumper 1 must be set to the downward position, connecting GPIO jumper 1 to VCC
- GPIO jumper 2 must be set to the upward position, connecting GPIO jumper 2 to GND

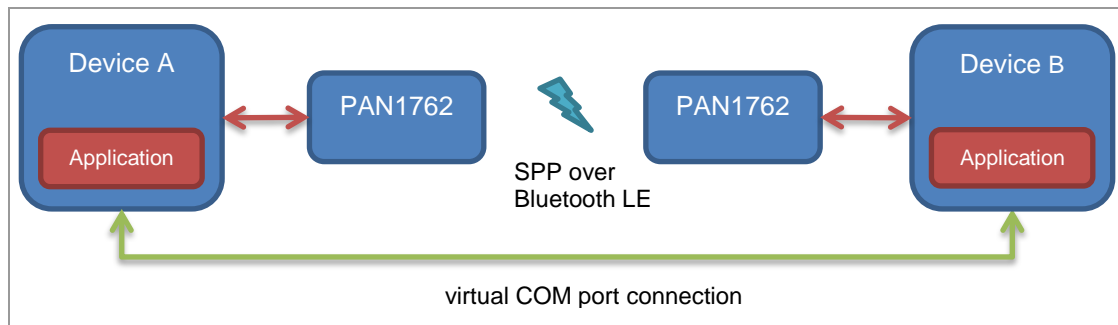


Please note that the AT command firmware must be written to the in-device flash as explained in [⇒ 8.2 Programming the Firmware](#).

5 SPP over Bluetooth LE Profile

The concept of Bluetooth Low Energy encourages developers to write custom and specific Bluetooth Low Energy profiles for their special hardware.

But even if the Bluetooth SDK in conjunction with the PAN1762 module makes this very easy, sometimes all you want to have is some sort of wireless serial cable replacement and transfer some data easily between two devices.



The Bluetooth SDK already contains such a wireless serial cable replacement profile which is called “SPP over Bluetooth LE” profile.

If used on two devices, device A and device B, the SPP over Bluetooth LE profile basically provides a virtual COM port connection between these two devices.



For more information please check out the SPP over Bluetooth LE profile client and server specifications which are part of the Bluetooth SDK:

- `BT_SDK_v_4_2_0\documentation\client_profiles\spp_over_ble_profile.pdf`
- `BT_SDK_v_4_2_0\documentation\server_profiles\spp_over_ble_profile.pdf`

6 Bluetooth Low Energy Basics

The following Bluetooth Low Energy basics will be briefly explained:

- Peripheral device and central
- Master and slave
- Server and client



For more information please check out the Bluetooth Core Specification
<https://www.bluetooth.com/specifications/bluetooth-core-specification>.

The **peripheral device** is the device which announces a service via advertising, while the **central device** discovers peripheral devices via scanning.

The **master** is the device which initiates a connection, while the **slave** device is accepting an incoming connection.

If a device is providing any information to a remote device it is acting as a **server**, while the other device using this information is called the **client**.

The most common setup is where a mobile phone acts as a central device, finds a peripheral device via scanning, then acts as a master to establish a connection and then works as a client to retrieve information.

Then there might be a sensor device, which acts as a peripheral device and waits for an incoming connection while advertising. When a connection comes in, it acts as a slave and accepts the connection, then works as a server to provide its sensor information.

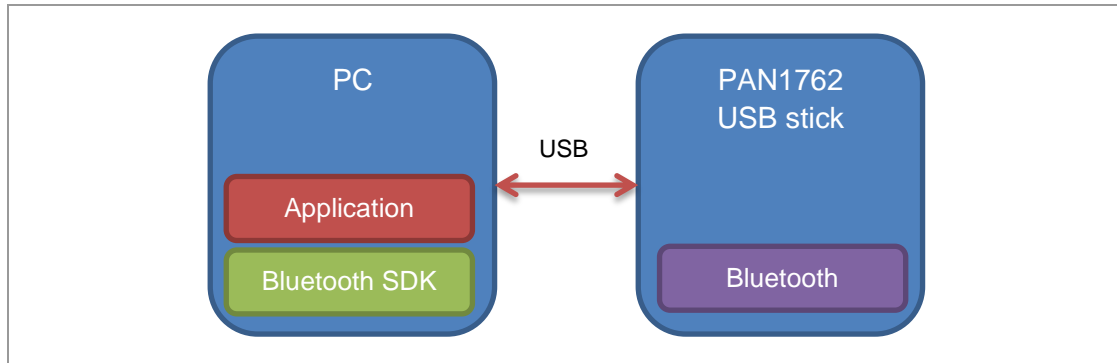


Please note that the device configuration does not need to be fixed during the runtime of a device.

Devices may freely change between being a **peripheral device** or a **central device** and then act as a **master** or a **slave**. When in a connection, each device can be both a **server** and a **client** and use the resources of the remote device freely.

7 Host Mode

With the PAN1762 USB stick a standard PC can act as the host processor.



The on-board “FTDI USB UART” will transparently convert the UART connection to a USB connection for easier use.

The Toshiba Bluetooth SDK has to be run in the context of the host processor, which is a PC for the PAN1762 USB stick.

7.1 PC Application

The software package includes a sample project called “host_mode” that shows how to use the Bluetooth SDK in a plain C application.

It showcases the use of the SPP over Bluetooth LE profile and comes in two project configurations:

- One will compile an application acting as a central device,
- The other will compile an application acting as a peripheral device.



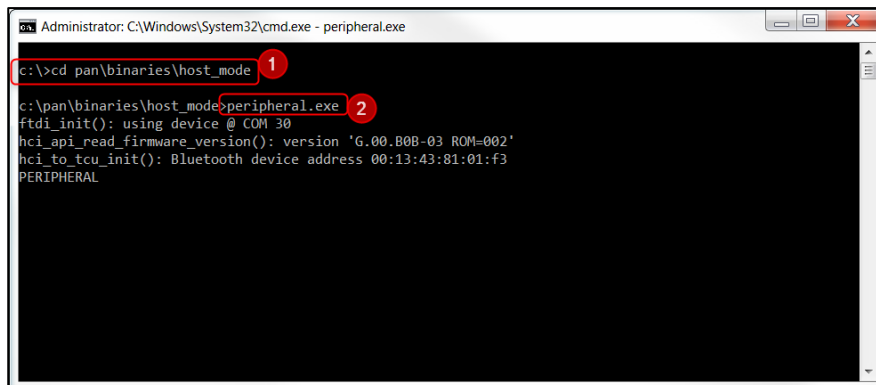
The software package includes precompiled binaries that can be used for testing. The binaries are available under the subdirectory `binaries/host_mode` in the installation directory.

The following requirements must be met:

- ✓ All of the attached PAN1762 USB sticks are removed
- ✓ All the PAN1762 USB sticks are jumpered to host mode as explained in [⇒ 4.4.1 Host Mode](#)

1. Attach one of the PAN1762 USB sticks and launch a Windows command line prompt, for example by executing `cmd.exe` from the **Start menu**.

2. Navigate to the folder `binaries\host_mode` (1) with the `cd` command and execute `peripheral.exe` (2).

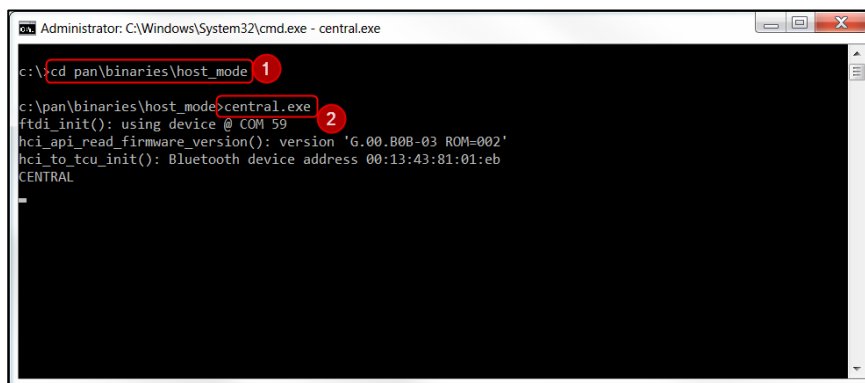


```
Administrator: C:\Windows\System32\cmd.exe - peripheral.exe

c:\>cd pan\binaries\host_mode
c:\pan\binaries\host_mode>peripheral.exe
ftdi_init(): using device @ COM 30
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:f3
PERIPHERAL
```

➔ This PAN1762 USB stick will now act as a peripheral device.

3. Attach the other PAN1762 USB stick and launch another Windows command line prompt.
4. Navigate to the folder `binaries\host_mode` (1) with the `cd` command and execute `central.exe` (2).

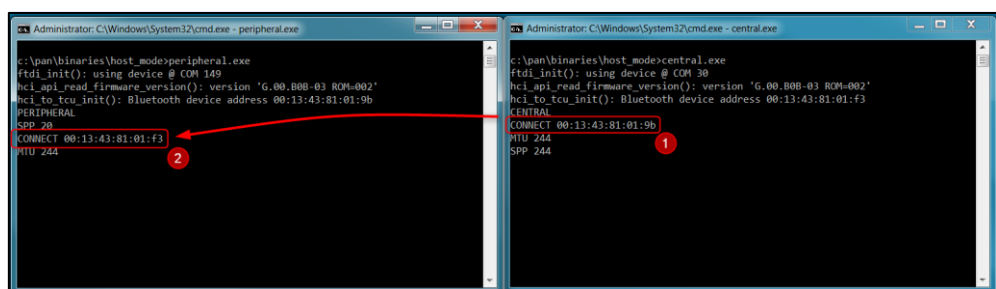


```
Administrator: C:\Windows\System32\cmd.exe - central.exe

c:\>cd pan\binaries\host_mode
c:\pan\binaries\host_mode>central.exe
ftdi_init(): using device @ COM 59
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:eb
CENTRAL
```

➔ This PAN1762 USB stick will now act as a central device.

➔ The central device will immediately recognize the peripheral device and establish a connection.



```
Administrator: C:\Windows\System32\cmd.exe - peripheral.exe
Administrator: C:\Windows\System32\cmd.exe - central.exe

c:\pan\binaries\host_mode>peripheral.exe
ftdi_init(): using device @ COM 149
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:9b
PERIPHERAL
SPP 244
CONNECT 00:13:43:81:01:f3

c:\pan\binaries\host_mode>central.exe
ftdi_init(): using device @ COM 30
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:f3
CENTRAL
CONNECT 00:13:43:81:01:9b
HYU 244
SPP 244
```

Now data can be transferred back and forth by pressing key in the respective terminal window.
In this example the one side sent **hello**, while the other side sent **world**.

```

Administrator: C:\Windows\System32\cmd.exe - peripheral.exe
c:\pan\binaries\host_mode>peripheral.exe
ftdi_init(): using device @ COM 149
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:9b
PERIPHERAL
SPP 20
CONNECT 00:13:43:81:01:f3
MTU 244
hello

Administrator: C:\Windows\System32\cmd.exe - central.exe
c:\pan\binaries\host_mode>central.exe
ftdi_init(): using device @ COM 30
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:f3
CENTRAL
CONNECT 00:13:43:81:01:9b
MTU 244
SPP 244
world_
  
```

Terminate application

Any application can be terminated.

1. Press **CTRL+C**.

→ The connection between the two devices will be shut down accordingly.



Restart after pressing **CTRL+C**

If you want to restart the demo after pressing **CTRL+C**, make sure to press the “reset” button on the corresponding PAN1762 USB stick.

In the following example, `peripheral.exe` (1) was interrupted by pressing **CTRL+C** and the connection between the two devices was terminated.

Afterwards the corresponding PAN1762 USB stick was reset by pressing the “reset” button.

Then `peripheral.exe` (1) was started again and the connection was immediately established again (2).

```

Administrator: C:\Windows\System32\cmd.exe - peripheral.exe
c:\pan\binaries\host_mode>peripheral.exe
ftdi_init(): using device @ COM 149
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:9b
PERIPHERAL
SPP 20
CONNECT 00:13:43:81:01:f3
MTU 244
c:\pan\binaries\host_mode>peripheral.exe 1
ftdi_init(): using device @ COM 149
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:9b
PERIPHERAL
SPP 20
CONNECT 00:13:43:81:01:f3
MTU 244

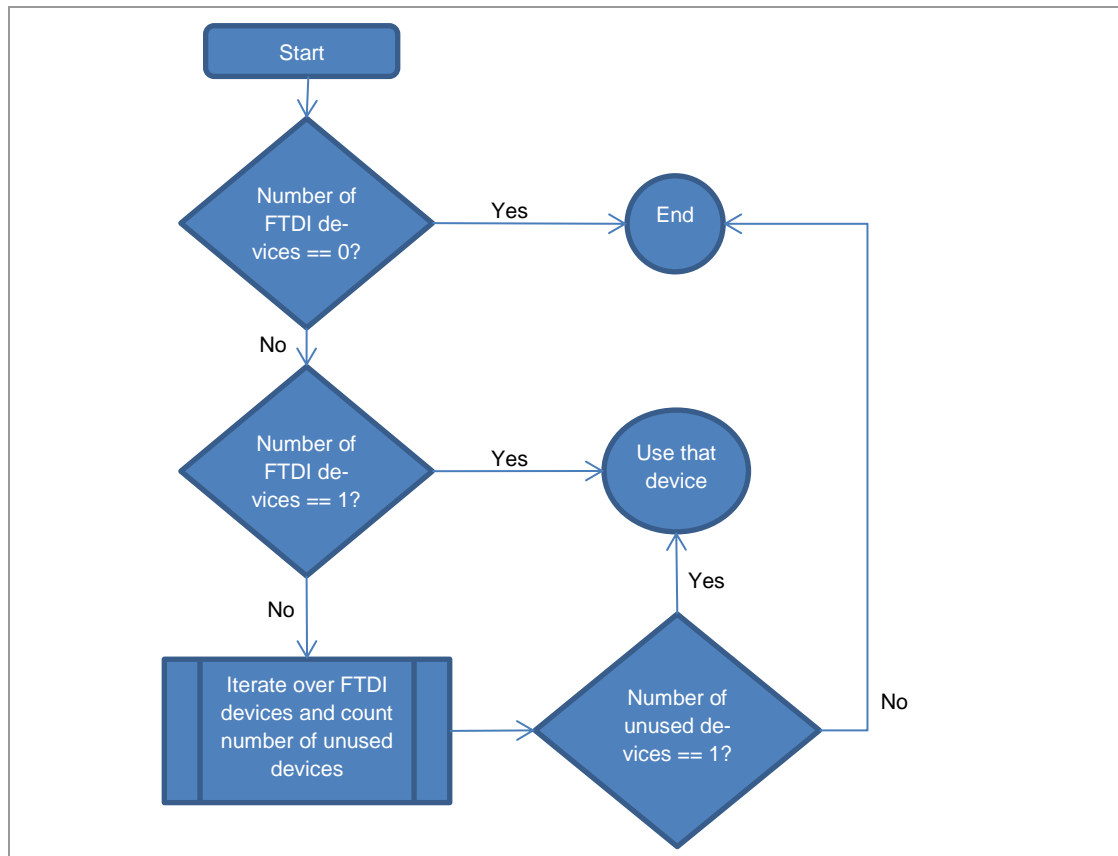
Administrator: C:\Windows\System32\cmd.exe - central.exe
c:\pan\binaries\host_mode>central.exe
ftdi_init(): using device @ COM 30
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:f3
CENTRAL
CONNECT 00:13:43:81:01:9b
MTU 244
SPP 244
DISCONNECT
CONNECT 00:13:43:81:01:9b 2
MTU 244
SPP 244
  
```

7.2 COM Port Handling

The sample applications seem to find out which COM port they should use.

In order to make development easier, the applications do not require that a COM port is given.

Instead use the following heuristic to find the COM port they should use:



By using one PAN1762 USB stick at a time, this scheme will always find the correct COM port to use.

By using two or more PAN1762 USB sticks, then you any of the devices can reset and restart at any time, as long as the application is running on all the other PAN1762 USB sticks.

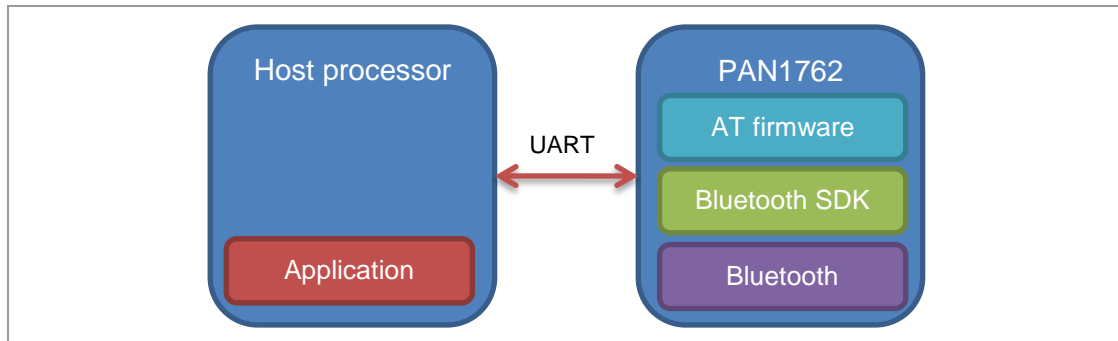
The benefit of this approach is that it is not necessary to remember any of the COM ports as long as the PAN1762 USB sticks are attached and the applications are started one after another.

To start from scratch and restart all the applications, it could be necessary to detach one of the PAN1762 USB sticks, because more than one unused FTDI device will be found by the scheme.

8 AT Command Mode

The AT command firmware is a special PAN1762 module firmware that comes with the Toshiba Bluetooth SDK delivery.

The Toshiba Bluetooth SDK will be run in the context of the PAN1762 module as part of the AT command firmware and does most of the necessary Bluetooth processing that usually the host processor would need to do.



The on-board FTDI USB UART will transparently convert the UART connection to a USB connection for easier use.

In this scenario a standard PC can act as the host processor and the application on the host processor will use the AT command interface of the AT command firmware.



Please note that the AT command firmware in the Toshiba Bluetooth SDK does not support the central role yet.

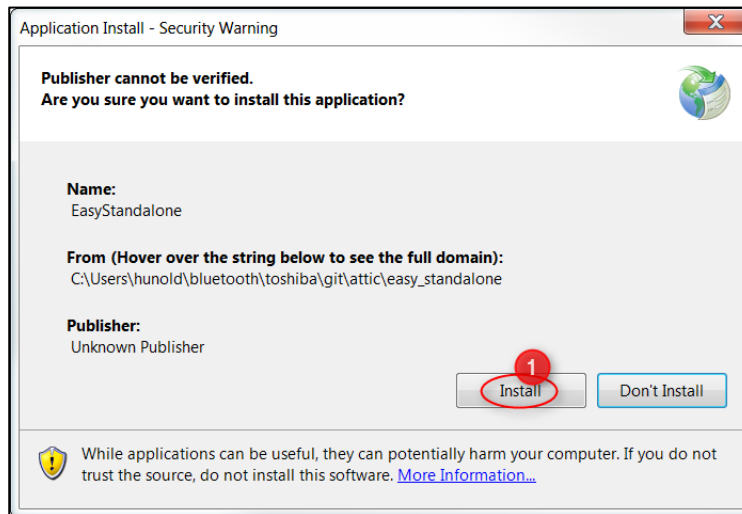
8.1 Installing EasyStandalone

The Toshiba Bluetooth SDK includes the application **EasyStandalone** which can be used to write the AT command firmware to the in-chip flash memory.

If the tool **unzipper** has been used as explained in [⇒ 3.1 Unzipper Tool](#) then the installer for the tool **EasyStandalone** will be present in the directory `binaries\EasyStandalone`.

1. Execute `setup.exe` from the directory `binaries\EasyStandalone`.

2. Click **Install** (1).



- ➔ After the installation has finished **EasyStandalone** will start up automatically and is available from the **Start menu** as well.



EasyStandalone is explained in the “Bluetooth SDK developers guide” in chapter “APPENDIX: Working with EasyStandalone Tool”.

8.2 Programming the Firmware

The following requirements must be met:

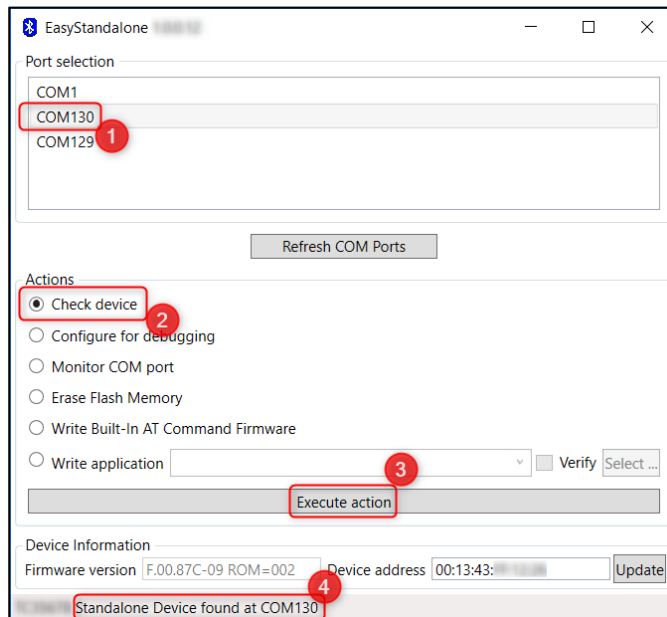
- ✓ All of the attached PAN1762 USB sticks are removed
- ✓ All the PAN1762 USB sticks are jumpered to host mode as explained in ➔ [4.4.1 Host Mode](#)



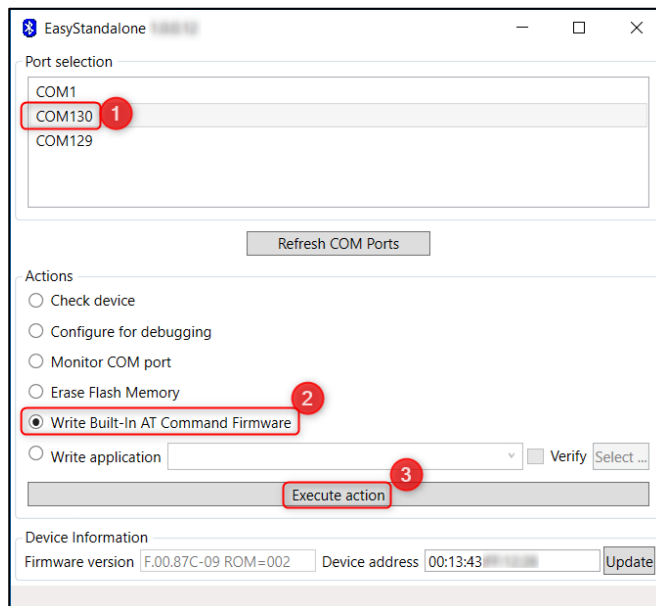
Please double check that the latest version of **EasyStandalone** from the Bluetooth SDK version 4.2.0 is installed and used.

1. Attach one of the PAN1762 USB sticks.
2. Go to the **Start menu** and launch **EasyStandalone**.
3. Make sure that the COM port selection is correct because **EasyStandalone** does not have an automatic detection.

4. Select a COM port (1) from the list of devices in the **Port selection**



5. Choose **Check device** (2) and click **Execute action** (3).
 ➔ The firmware version and the Bluetooth device address will be displayed (4).
6. Make sure that the COM port selection (1) is correct.



7. Choose **Write Built-In Firmware** (2).
8. Click **Execute action** (3) to start the programming.
 ➔ A progress bar will be shown and after a couple of seconds the programming will be finished.
9. Close **EasyStandalone**.
10. Jumper PAN1762 USB stick to "AT command mode" as explained in
 ➔ [4.4.3 AT Command Mode](#).

8.3 PC Application

The software package includes a sample project called “at_command_mode” that shows how to use the AT command firmware in a plain C application.

It showcases the use of the SPP over Bluetooth LE profile and comes in one project configuration that will compile an application acting as a peripheral device.



The software package includes precompiled binaries that can be used for testing. The binaries are available under the subdirectory `binaries/at_command_mode` in the installation directory.

The following requirements must be met:

- ✓ All of the attached PAN1762 USB sticks are removed
- ✓ All the PAN1762 USB sticks are jumpered to AT command mode as explained in [⇒ 4.4.3 AT Command Mode](#)

1. Attach one of the PAN1762 USB sticks and launch a Windows command line prompt, for example by executing `cmd.exe` from the **Start menu**.
2. Navigate to the folder `binaries\at_command_mode` (1) with the `cd` command and execute `peripheral.exe` (2).

```
Administrator: C:\Windows\System32\cmd.exe - peripheral.exe
c:\>cd c:\pan\binaries\at_command_mode 1
c:\pan\binaries\at_command_mode>peripheral.exe 2
Using device @ COM 6
--> 'AT+SYSSBAUD'
<-- 'OK'
<-- '+EV SYS_BAUDRATE_EVENT'
--> 'AT+LESPPINIT'
<-- 'OK'
--> 'AT+LESPPADV'
<-- 'OK'
```

➔ This PAN1762 USB stick will now act as a peripheral device.



Because the AT command firmware in the Toshiba Bluetooth SDK does not support central role yet, we use `central.exe` and a PAN1762 USB stick in host mode configuration to act as the central device instead.

3. Attach the other PAN1762 USB stick and launch another Windows command line prompt. Make sure that this PAN1762 USB sticks is jumpered to **host mode** as explained in [⇒ 4.4.1 Host Mode](#).

4. Navigate to the folder `binaries\host_mode` (1) with the `cd` command and execute `central.exe` (2).

```

Administrator: C:\Windows\System32\cmd.exe - central.exe

c:\>cd pan\binaries\host_mode
c:\pan\binaries\host_mode>central.exe
ftdi_init(): using device @ COM 59
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM=002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:eb
CENTRAL
  
```

→ This PAN1762 USB stick will now act as a central device.



The AT command mode sample application uses the sample COM port handling as explained in [7.2 COM Port Handling](#).

- The central device will immediately recognize the peripheral device and establish a connection.

```

Administrator: C:\Windows\System32\cmd.exe - peripheral.exe
<-- "ATOK"
<-- "EV SYS_BAUDRATE_EVENT"
--> "AT+LESPINIT"
<-- "ATOK"
--> "AT+LESPADV"
<-- "ATOK"
--> "AT+LECONN_COMPLETED_EVENT"
--> "AT+LECONINFO"
<-- "ATOK"
--> "AT+LECONINFO"
<-- "EV SERVER_MTU_CHANGE_EVENT"
--> "AT+GSRVIND"
<-- "ATOK"
--> "AT+GSRVIND"
<-- "EV SERVER_READ_EVENT"
--> "AT+GSRVIND"
<-- "ATOK"
--> "AT+GSRVIND"
<-- "EV SERVER_WRITE_EVENT"
--> "AT+GSRVIND"
<-- "ATOK"
--> "AT+GSRVIND"
<-- "EV SEC_BONDING_EVENT"

Administrator: C:\Windows\System32\cmd.exe - central.exe
c:\>cd c:\pan\binaries\host_mode
c:\pan\binaries\host_mode>central.exe
ftdi_init(): using device @ COM 4
hci_to_tcu_init(): Bluetooth device address 00:13:43:ff:12:29
hci_api_read_firmware_version(): version 'F.00.87C-09 ROM=002'
CENTRAL
CONNECT 00:13:43:ff:12:26
MTU 20
SPP 20
ENCRYPTION_OK
  
```

Now data can be transferred back and forth by pressing key in the respective terminal window. In this example the one side sent **hello**, while the other side sent **world**.

```

Administrator: C:\Windows\System32\cmd.exe - peripheral.exe
--> "AT+GSRVIND"
<-- "ATOK"
--> "AT+GSRVIND"
<-- "ATOK"
--> "EV SERVER_INDICATION_ACK_EVENT"
--> "AT+GSRVIND"
<-- "ATOK"
--> "EV SERVER_INDICATION_ACK_EVENT"
--> "AT+GSRVIND"
<-- "ATOK"
--> "EV SERVER_INDICATION_ACK_EVENT"
--> "AT+GSRVIND"
<-- "ATOK"
--> "EV SERVER_INDICATION_ACK_EVENT"
--> "AT+GSRVIND"
<-- "ATOK"
--> "EV SERVER_INDICATION_ACK_EVENT"
--> "AT+GSRVIND"
<-- "ATOK"
--> "EV SERVER_INDICATION_ACK_EVENT"

Administrator: C:\Windows\System32\cmd.exe - central.exe
c:\>cd c:\pan\binaries\host_mode
c:\pan\binaries\host_mode>central.exe
ftdi_init(): using device @ COM 4
hci_to_tcu_init(): Bluetooth device address 00:13:43:ff:12:29
hci_api_read_firmware_version(): version 'F.00.87C-09 ROM=002'
CENTRAL
CONNECT 00:13:43:ff:12:26
MTU 20
SPP 20
ENCRYPTION_OK
world
  
```

Terminate application

Any application can be terminated.

1. Press **CTRL+C**.

➔ The connection between the two devices will be shut down accordingly.



Restart after pressing **CTRL+C**

If you want to restart the demo after pressing **CTRL+C**, make sure to press the “reset” button on the corresponding PAN1762 USB stick.

In the following example, `peripheral.exe` (1) was interrupted by pressing **CTRL+C** and the connection between the two devices was terminated.

Afterwards the corresponding PAN1762 USB stick was reset by pressing the reset button.

Then `peripheral.exe` (1) was started again and the connection was immediately established again (2).

```

Administrator: C:\Windows\System32\cmd.exe - peripheral.exe
c:\pan\binaries\host_mode>peripheral.exe
ftdi_init(): using device @ COM 149
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM-002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:9b
PERIPHERAL
SPP 20
CONNECT 00:13:43:81:01:f3
MTU 244

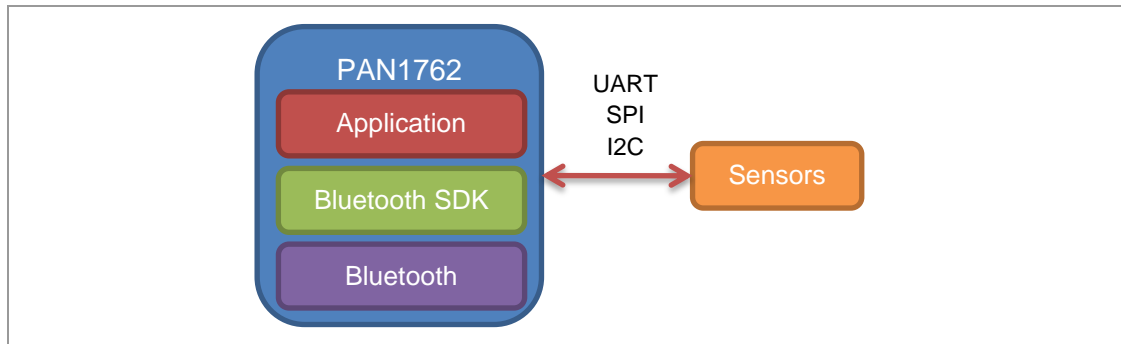
c:\pan\binaries\host_mode>peripheral.exe 1
ftdi_init(): using device @ COM 149
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM-002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:9b
PERIPHERAL
SPP 20
CONNECT 00:13:43:81:01:f3
MTU 244

Administrator: C:\Windows\System32\cmd.exe - central.exe
c:\pan\binaries\host_mode>central.exe
ftdi_init(): using device @ COM 30
hci_api_read_firmware_version(): version 'G.00.B0B-03 ROM-002'
hci_to_tcu_init(): Bluetooth device address 00:13:43:81:01:f3
CENTRAL
CONNECT 00:13:43:81:01:9b
MTU 244
SPP 244

DISCONNECT
CONNECT 00:13:43:81:01:9b 2
MTU 244
SPP 244
    
```

9 Standalone Mode

The PAN1762 USB stick can be used in standalone mode, where the built-in Cortex[®]-M0 microcontroller works as the host processor as well.

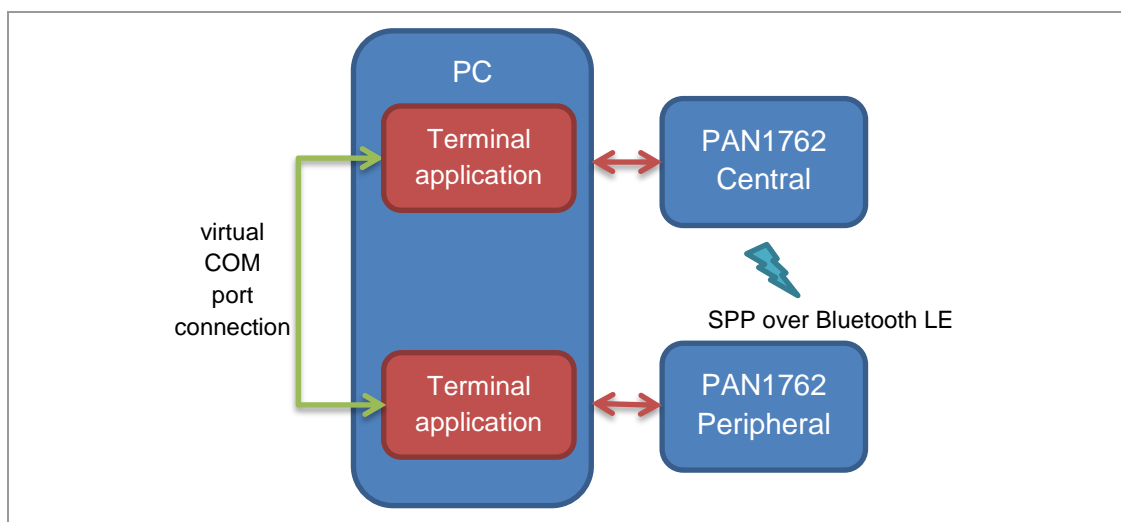


The on-board “FTDI USB UART” is available here as well and can be used by the application running on PAN1762 module to additionally interface with the PC it is connected to.

9.1 Application

The software package includes a sample project called “standalone_mode” that shows how to use the PAN1762 module in a standalone application.

It showcases the use of the SPP over Bluetooth LE profile and comes in two project configurations: one will compile as a firmware for a central device and as a firmware for a peripheral device.



These two different firmwares have to be programmed to individual PAN1762 USB sticks.

When powered up, the PAN1762 USB sticks will immediately connect to each other and form a virtual COM port connection.

From the PC that hosts the PAN1762 USB sticks, two terminal applications can be started and used to transfer data back and forth.



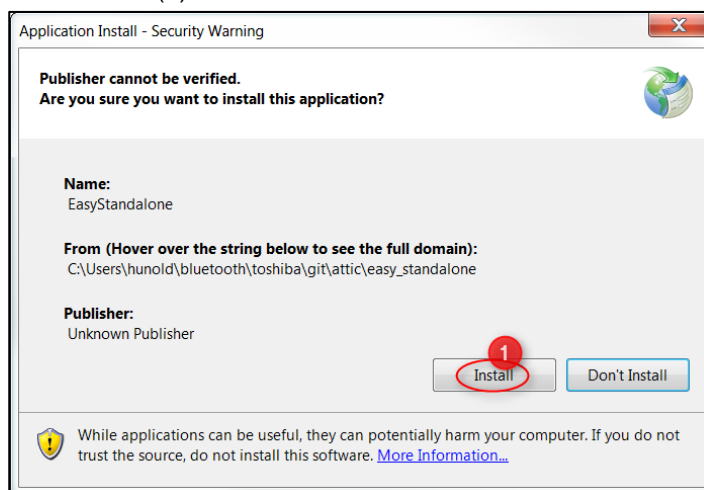
The software package includes pre-compiled binaries that can be used for testing and are available under the installation directory in the subdirectory `binaries/standalone_mode`.

9.2 Installing EasyStandalone

The Toshiba Bluetooth SDK includes the application **EasyStandalone** which can be used to write a firmware to the in-chip flash memory.

If the tool **unzipper** has been used as explained in ⇒ 3.1 **Unzipper Tool** then the installer for the tool **EasyStandalone** will be present in the directory `binaries\EasyStandalone`.

1. Execute `setup.exe` from the directory `binaries\EasyStandalone`.
2. Click **Install** (1).



- ➔ After the installation has finished **EasyStandalone** will start up automatically and is available from the **Start menu** as well.



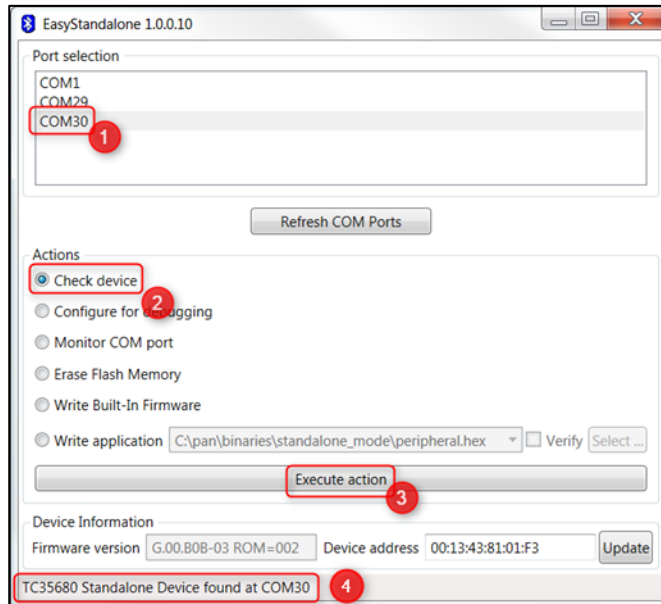
EasyStandalone is explained in the “Bluetooth SDK developers guide” in chapter “APPENDIX: Working with EasyStandalone Tool”.

9.3 Programming the Firmware

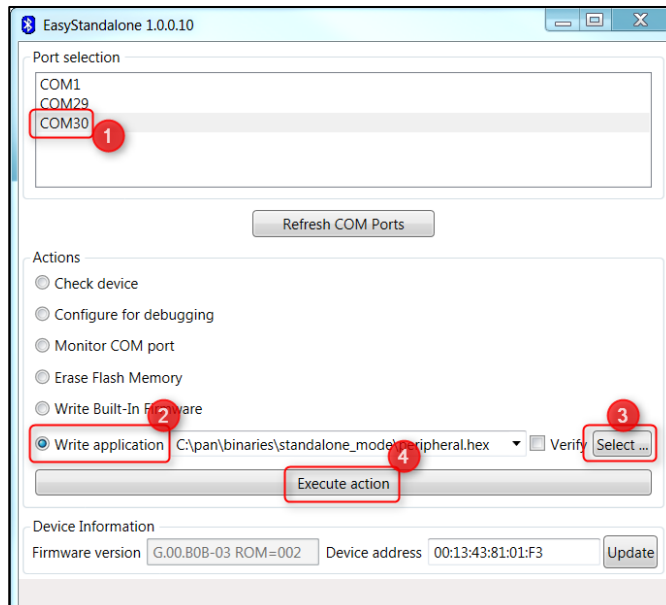
The following requirements must be met:

- ✓ All of the attached PAN1762 USB sticks are removed
- ✓ All the PAN1762 USB sticks are jumpered to host mode as explained in ⇒ 4.4.1 **Host Mode**

1. Attach one of the PAN1762 USB sticks.
2. Go to the **Start menu** and launch **EasyStandalone**.
3. Make sure that the COM port selection is correct because **EasyStandalone** does not have an automatic detection.
4. Select a COM port (1) from the list of devices in the **Port selection**.



5. Choose **Check device** (2) and click **Execute action** (3).
➔ The firmware version and the Bluetooth device address will be displayed (4).
6. Make sure that the COM port selection (1) is correct.



7. Choose **Write application** (2).
8. Click **Select...** (3) to choose the `peripheral.hex` firmware from the subdirectory `binaries/standalone_mode`.

9. Click **Execute action** (4) to start the programming.
 - ➔ A progress bar will be shown and after a couple of seconds the programming will be finished.
10. Now **EasyStandalone** can be closed.
11. Jumper the PAN1762 USB stick to “standalone mode” as explained in
 - ⇒ 4.4.2 Standalone Mode.



Repeat the steps above with the other PAN1762 USB stick, but using the firmware file `central.hex`.

9.4 Testing the Setup

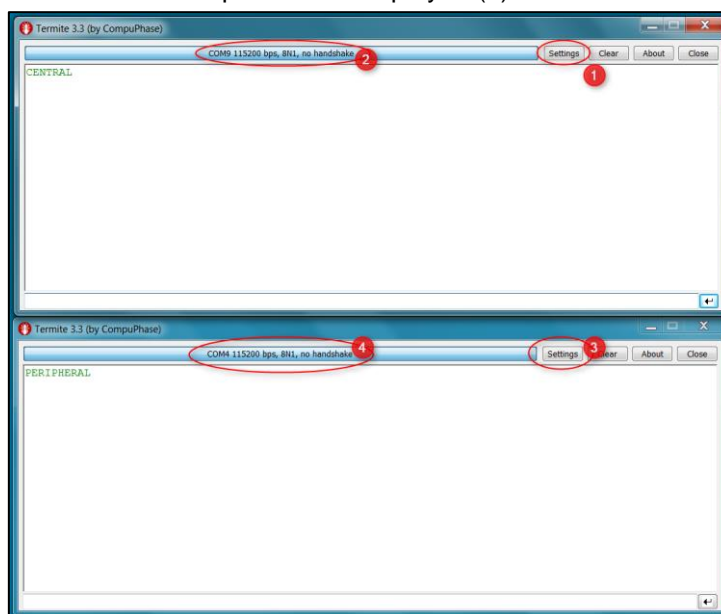
The following requirements must be met:

- ✓ Both PAN1762 USB sticks are jumpered to “standalone mode” as explained in
 - ⇒ 4.4.2 Standalone Mode
- ✓ Installed terminal application

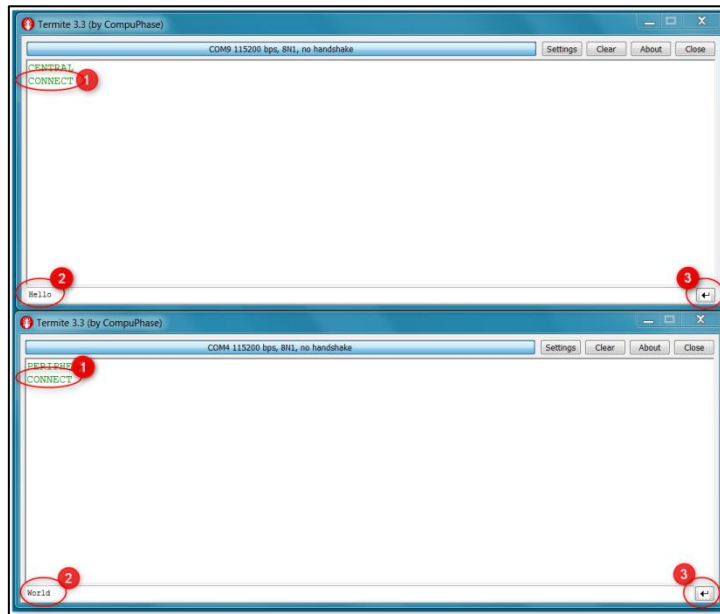


If you do not have any terminal application yet, you can try **Termite** from https://www.compuphase.com/software_termite.htm. In the following steps **Termite** is used.

1. Use any terminal application to connect to the well-known COM ports of the PAN1762 USB sticks.
2. Click **Settings** (1) to choose the desired COM port.
 - ➔ The chosen COM port will be displayed (2).



3. Press the “reset” button on the PAN1762 USB sticks.
 - ➔ They will identify themselves by printing either **CENTRAL** or **PERIPHERAL** to the terminal window.
 - ➔ The central device will immediately try to connect to the peripheral device. If a connection has been established, **CONNECT** (1) will be printed in each of the terminal windows.



Now you can use the input textbox (2) and the “send” button (3) to send data back and forth between the two devices.

10 Development

The source code for all the sample applications is available under the directory `configurations`.

How to recompile the application from the source code for the different usage scenarios is explained in the following chapters.

10.1 Host Mode

10.1.1 Visual Studio® 2017

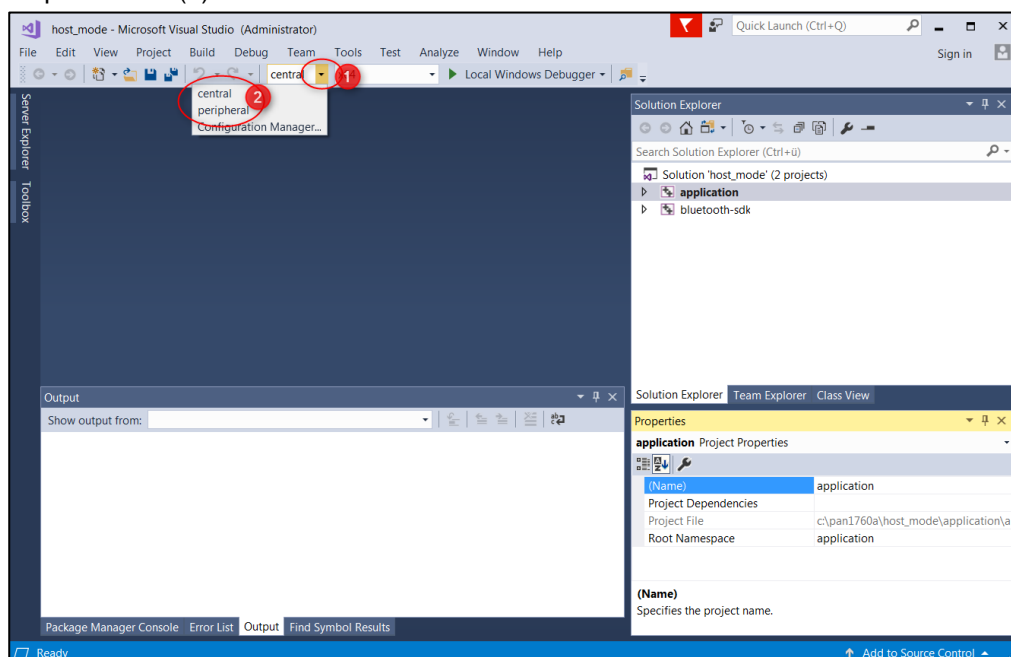
All sample application projects for the host mode that run on the PC are based on **Visual Studio 2017**. It is sufficient to install the **Visual Studio 2017 Community** edition.

For further information please visit <https://www.visualstudio.com/downloads/>.

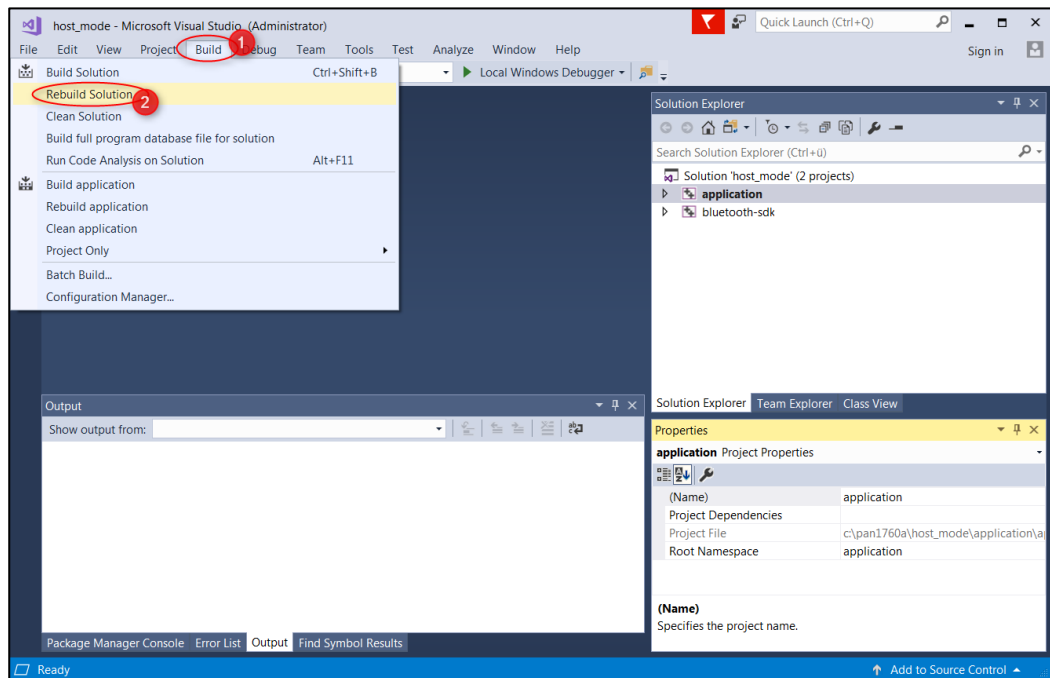
10.1.2 Working with the Sample Project

The sample project solution file is located under `host_mode\host_mode.sln`.

1. Open the sample project in **Visual Studio 2017**.
2. Click on the arrow “Solution Configurations” (1) to select the configuration from the drop down list (2).



3. Navigate to **Build** (1) and select **Rebuild Solution** (2) to build the currently selected configuration.



→ The resulting binaries can be found in the directory `host_mode\x64\` as `peripheral.exe` and `central.exe`.

10.2 AT Command Mode

10.2.1 Visual Studio® 2017

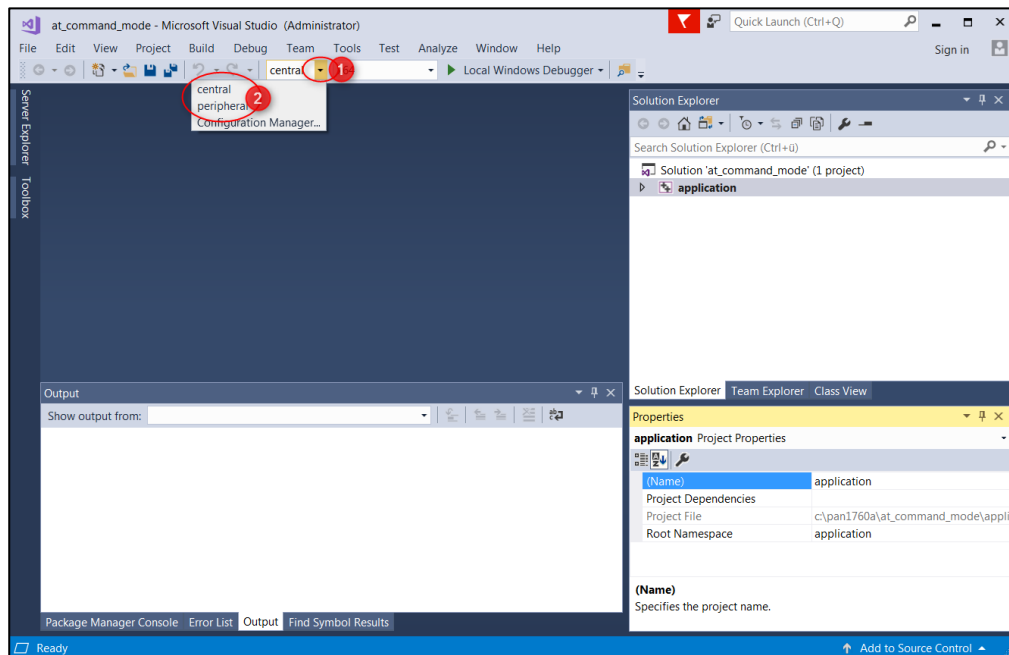
All sample application projects for the AT command mode that run on the PC are based on **Visual Studio 2017**. It is sufficient to install the **Visual Studio 2017 Community** edition. For further information please visit <https://www.visualstudio.com/downloads/>.

10.2.2 Working with the Sample Project

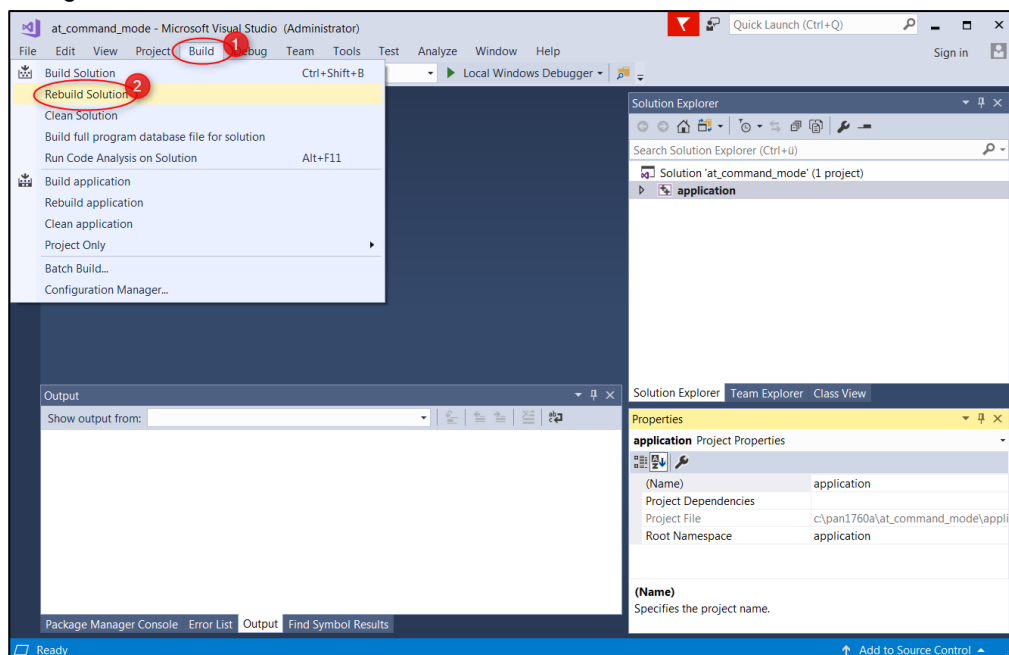
The sample project solution file is located under `at_command_mode\at_command_mode.sln`.

1. Open the sample project in **Visual Studio 2017**.

2. Click on the arrow “Solution Configurations” (1) to select the configuration from the drop down list (2).



3. Navigate to **Build** (1) and select **Rebuild Solution** (2) to build the currently selected configuration.



➔ The resulting binaries can be found in the directory `at_command_mode\x64\` as `peripheral.exe`.

10.3 Standalone Mode

10.3.1 IAR Embedded Workbench

All the sample application projects that run on the embedded Cortex-M0 microcontroller of PAN1762 are based on **IAR Embedded Workbench**.

Because the code sizes of any of the projects exceed 32 kB you either need to have a licensed version or use the 30-day trial version.

For further information please visit <https://www.iar.com/iar-embedded-workbench/>.

10.3.2 Working with the Sample Project – Windows

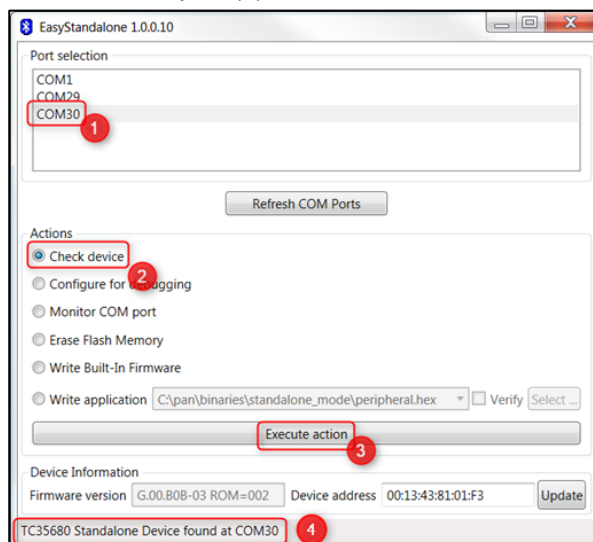
10.3.2.1 Device Preparation

In order to be able to debug applications reliably the in-device flash memory of the PAN1762 module must be configured correctly using **EasyStandalone**.

The following requirements must be met:

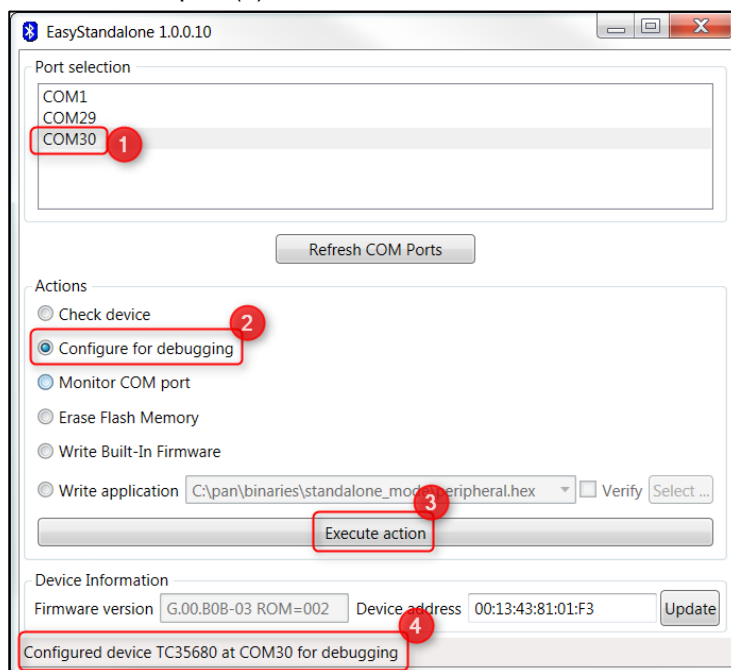
- ✓ All of the attached PAN1762 USB sticks are removed.
- ✓ All the PAN1762 USB sticks are jumpered to “host mode” as explained in ⇒ [4.4.1 Host Mode](#)

1. Attach one of the PAN1762 USB sticks.
2. Go to the **Start menu** and launch **EasyStandalone**.
3. Make sure that the COM port selection is correct because **EasyStandalone** does not have an automatic detection.
4. Select a COM port (1) from the list of devices in the **Port selection**.



5. Choose **Check device** (2) and click **Execute action** (3).
⇒ The firmware version and the Bluetooth device address will be displayed (4).

6. Select a COM port (1) from the list of devices in the **Port selection**.



7. Choose **Configure for debugging** (2).
8. Click **Execute action** (3) again.
 - ➔ The in-device flash memory will now be erased (4).

10.3.2.2 Pack-File Installation

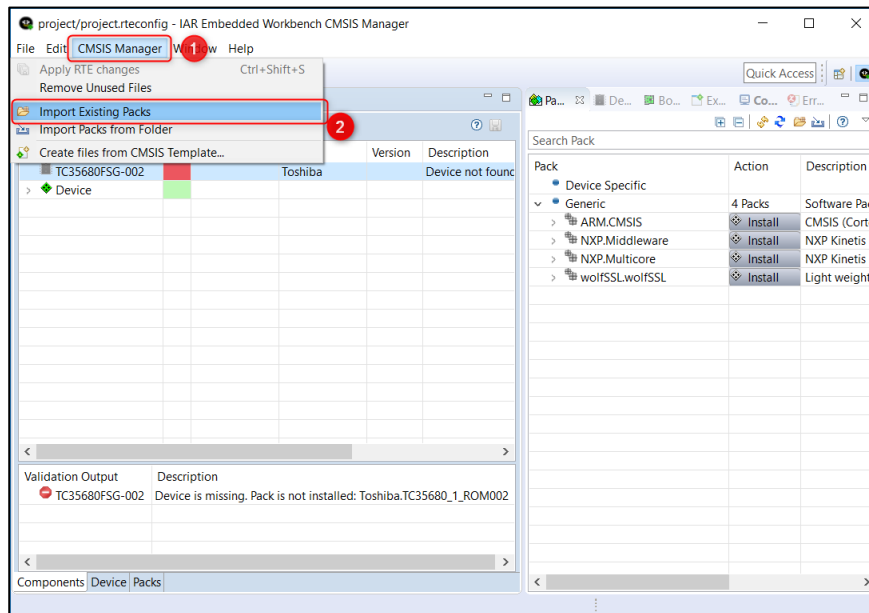
Drivers for the peripherals of PAN1762 are provided in a CMSIS-compliant “pack” file which is included in the Toshiba Bluetooth SDK.

If the “pack” file is not already available in the system it must be installed using the **IAR Embedded Workbench CMSIS Manager** when the sample project solution is opened for the first time.

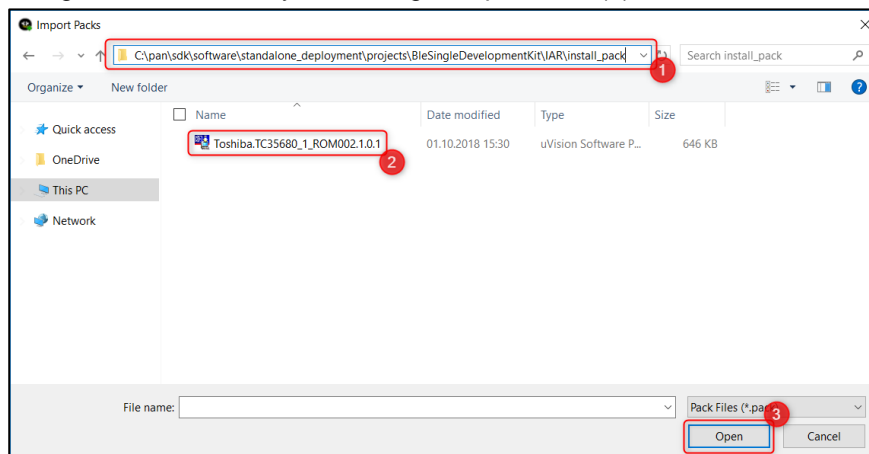
The sample project solution file is located under `standalone_mode\project.eww`.

1. Open the sample project in **IAR Embedded Workbench**.
 - ➔ The IAR Embedded Workbench CMSIS Manager will automatically be started.

2. Navigate to **CMSIS Manager** (1) and click **Import Existing Packs** (2).



3. Navigate to the directory containing the “pack” file (1).



4. Select `Toshiba.TC35680_1_ROM002.1.0.1.pack` (2) and click **Open** (3).

➔ The “pack” file will be installed automatically.



During the installation the **IAR Embedded Workbench CMSIS Manager** may try to download additional files from the Internet.

However, this application does not honor the proxy settings from the environment and the download may fail, for example, if you are behind a corporate firewall.

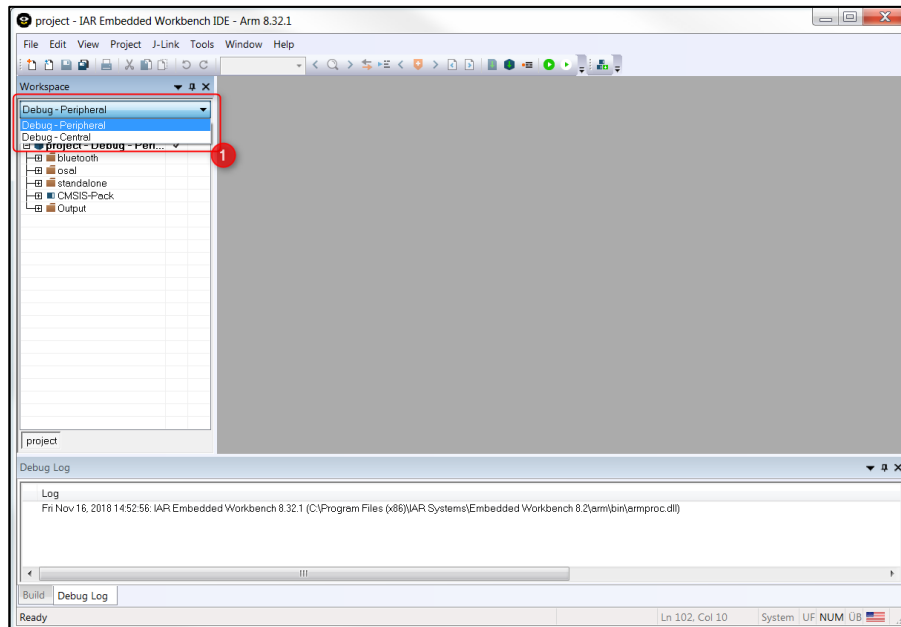
This is a known problem. Please follow the advice on the following page to fix this problem:

<https://www.iar.com/support/tech-notes/general/proxy-configuration-for-cmsis-pack-download/>

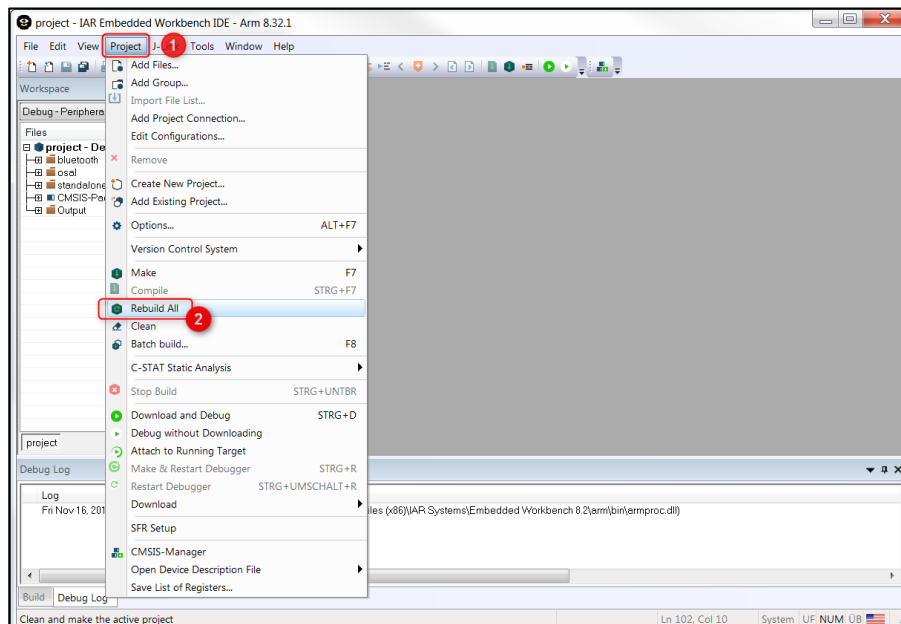
10.3.2.3 Compilation

The sample project solution file is located under `standalone_mode\project.eww`.

1. Open the sample project in **IAR Embedded Workbench**.
2. Click on the Workspace drop down list "Configuration" (1) to select the configuration.



3. Navigate to **Project** (1) and select **Rebuild All** (2) to build the currently selected configuration.

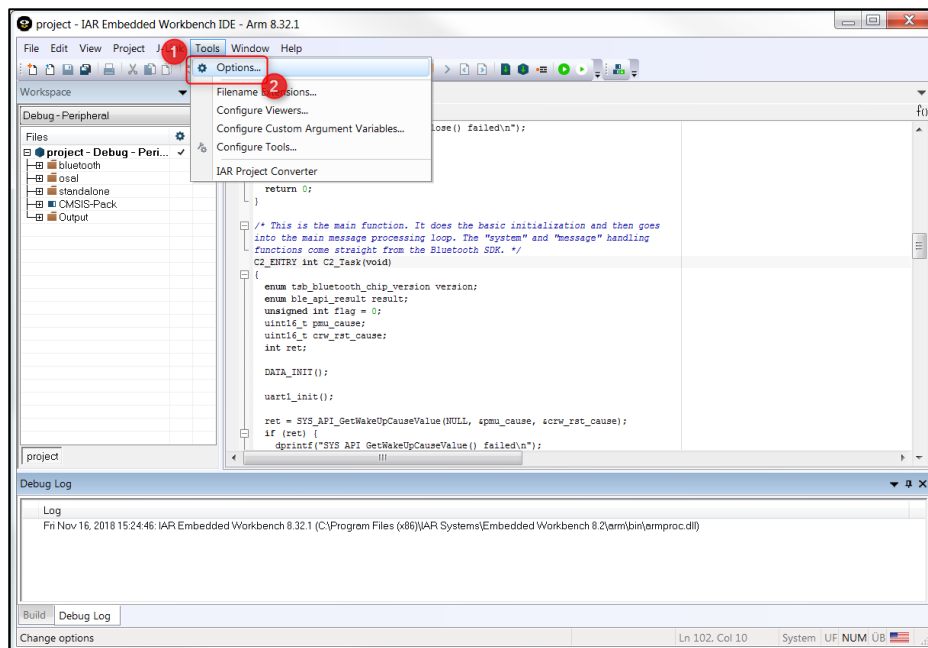


10.3.2.4 IDE Configuration

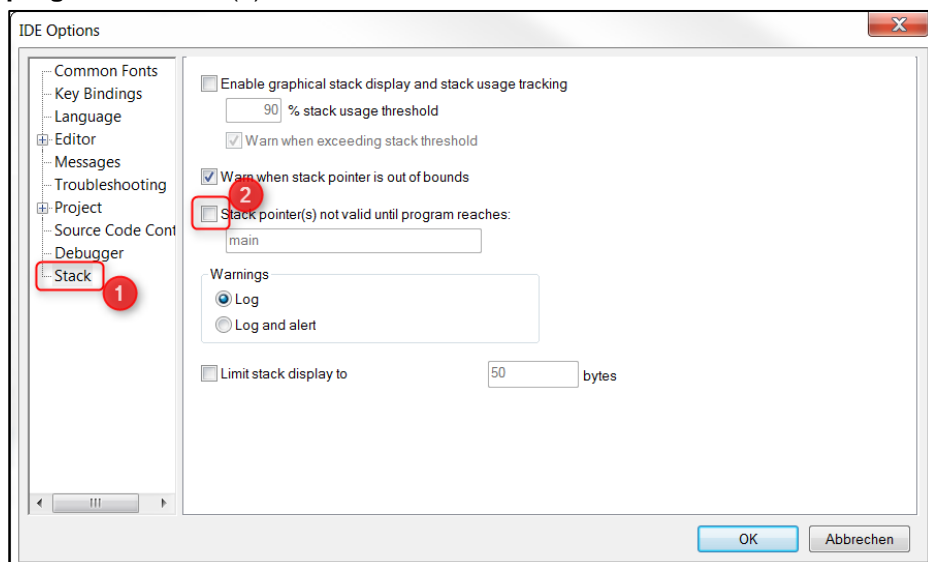
Avoid an annoying warning message

In order to avoid an annoying warning message it is necessary to change the configuration of the IAR Embedded Workbench.

1. Navigate to **Tools** (1) and click **Options...** (2).



2. Navigate to **Stack** (1). Make sure that the option **Stack pointer(s) not valid until program reaches** (2) is not selected.



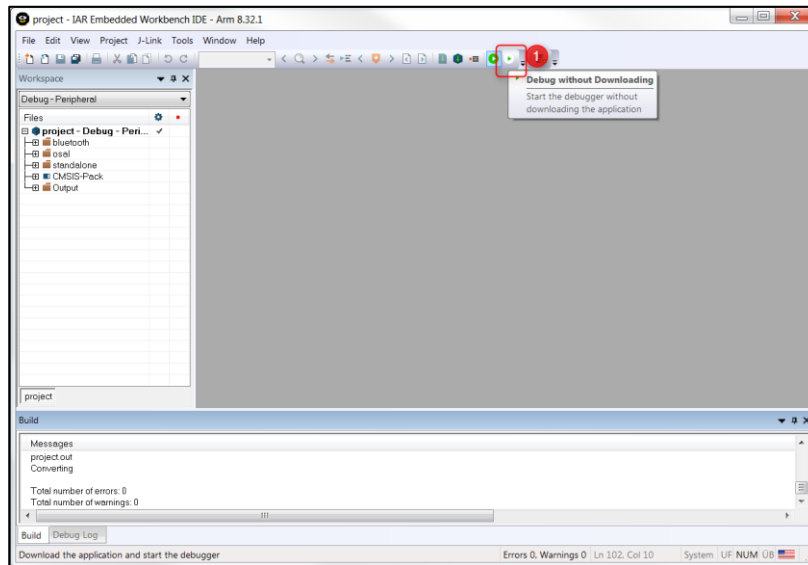
The reason is that the application does not contain a traditional function `main()` but uses a different name instead.

10.3.2.5 Debugging

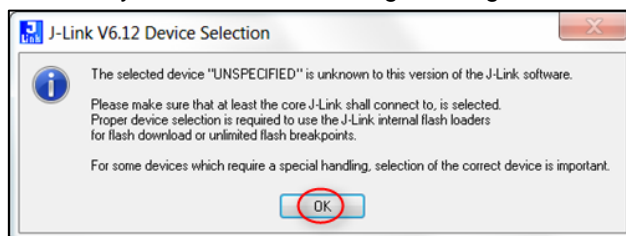
The following requirements must be met:

- ✓ The PAN1762 USB stick is jumpered to “standalone mode” as explained in [⇒ 4.4.2 Standalone Mode](#)
- ✓ The in-device flash memory is erased as explained in [⇒ 10.3.2.1 Device Preparation](#)

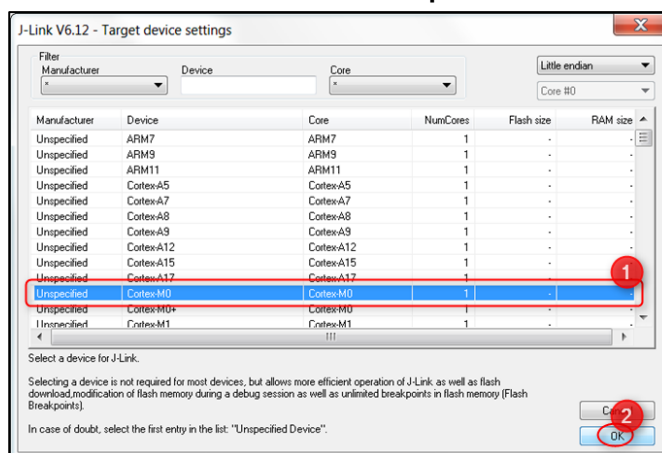
1. Click the option **Debug without Downloading** (1) to start the debug process.



- ➔ At the very first start the following message box will appear.



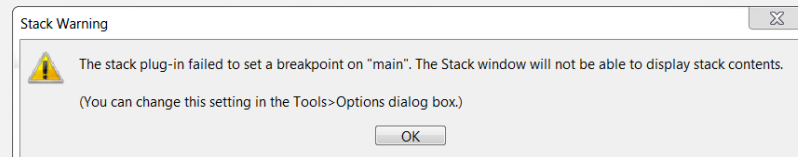
2. Click **OK**.
3. Choose from the list of devices **Unspecified/Cortex-M0** (1) and click **OK** (2).



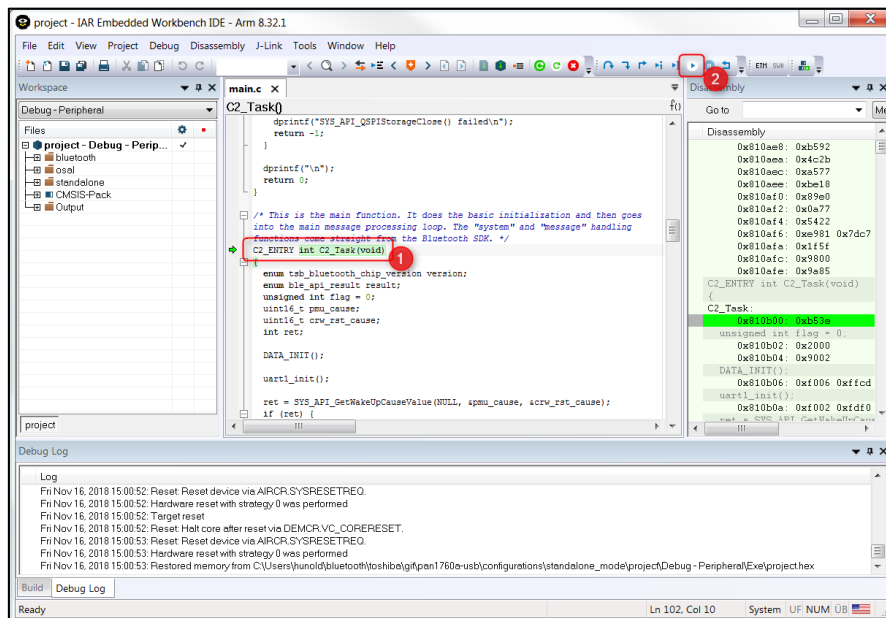


Stack Warning Message

If the following warning message appears, please make sure that you followed the project configuration advice in [⇒ 10.3.2.4 IDE Configuration](#).



- ➔ The application will now be started up but when it reaches the function `C2_Task()` (1) the application will be immediately interrupted.



4. Click the icon "Go" (2) to continue the application.

11 Appendix

11.1 Changes to the Toshiba Bluetooth SDK

During the installation using the tool **unzipper** extracts the Toshiba Bluetooth SDK to the subdirectory `sdk`.

As explained all the sample projects in the software package internally use the Toshiba Bluetooth SDK.

Some changes have been done to the Toshiba Bluetooth SDK to add missing features. Other changes have been done to circumvent problems. All these changes are regularly submitted to Toshiba for further inspection. The might not be adopted in upcoming versions of the SDK.

All changed files are contained in the file `bluetooth_sdk_differences.zip`. The individual changes are described in “unified diff” format in the file `bluetooth_sdk.diff` included in the ZIP file as well.

These changes are briefly explained in the following chapters.

11.1.1 Host Mode Deployment

SPP over Bluetooth LE Handling

The function `spp_over_ble_profile_init()` has been modified and now accepts a device name as a parameter.

This is necessary for some of the sample projects where the device name of the device running the SPP over Bluetooth LE profile must be configurable.

11.1.2 Standalone Deployment

Please note that the code for the SPP over Bluetooth LE profile from the host mode deployment is used.

11.2 Contact Details

11.2.1 Contact Us

Please contact your local Panasonic Sales office for details on additional product options and services:

For Panasonic Sales assistance in the **EU**, visit

<https://eu.industrial.panasonic.com/about-us/contact-us>

Email: wireless@eu.panasonic.com

For Panasonic Sales assistance in **North America**, visit the Panasonic website “Sales & Support” to find assistance near you at

<https://na.industrial.panasonic.com/distributors>

Please visit the **Panasonic Wireless Technical Forum** to submit a question at

<https://forum.na.industrial.panasonic.com>

11.2.2 Product Information

Please refer to the Panasonic Wireless Connectivity website for further information on our products and related documents:

For complete Panasonic product details in the **EU**, visit

<http://pideu.panasonic.de/products/wireless-modules.html>

For complete Panasonic product details in **North America**, visit

<http://www.panasonic.com/rfmodules>