AVR32798: EXTBT User Guide

Features

- **Hardware Description**
- **Powering up the Board**
- **Setup and Description of the Demonstration Firmware**
- **Use Cybercom t2BF 32-bit AVR® freeBT Bluetooth® stack and SPP profile**

32-bit AVR®
Microcontrollers

Application Note

1 Introduction

The Atmel EXTBT is a Bluetooth extension module for Atmel AVR development kits. The kit is equipped with the FREE2MOVE™ F2M03GXA Bluetooth module and provides Bluetooth functionality through a UART interface to any AVR devices. This guide shows the user how to quickly get started with this kit and describes the EXTBT hardware.

Figure 1-1. EXTBT Bluetooth extension board (top).

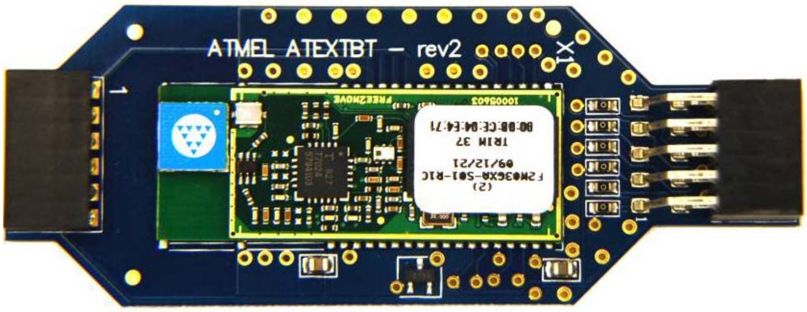


Figure 1-2. EXTBT Bluetooth extension board (bottom).

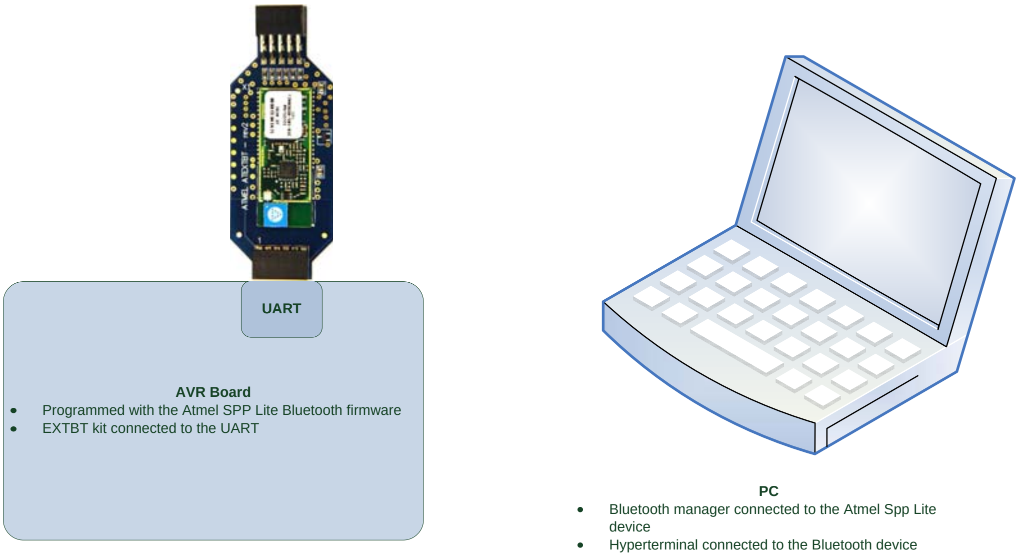
Rev. 32153A-AVR-11/10

2 Getting started

2.1 Setup

The Atmel AVR board, connected to the Atmel EXTBT kit, will be recognized as a Bluetooth device by the PC. The demonstration firmware allows communicating with the PC using a serial terminal connected to the Bluetooth COM port [using serial port profile (SPP)].

Figure 2-1. Setup block diagram.


UART
AVR Board
Programmed with the Atmel SPP Lite Bluetooth firmware
EXTBT kit connected to the UART
PC
Bluetooth manager connected to the Atmel Spp Lite device
Hyperterminal connected to the Bluetooth device

2.1.1 Powering up and connecting the module

The EXTBT board is powered through the 10-pin header, J1 (labeled WLESS), or the 6-pin header, J2 (labeled EVK1105 EVK1104).

The following sections described the connection with some of the released Atmel AVR UC3 kits. Other Atmel AVR kits using the same interface might also work.

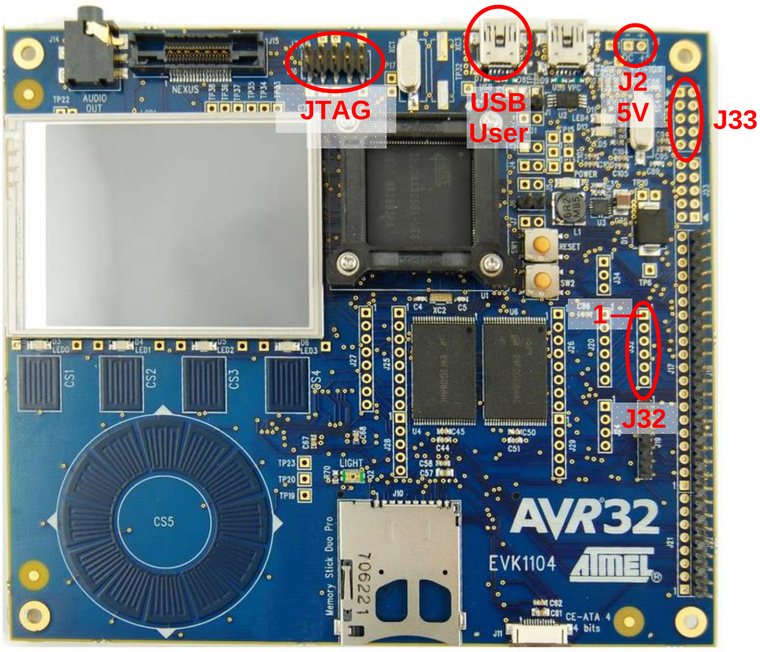
2.1.1.1 EVK1100

There is no dedicated header connection for the Atmel EVK1100 kit; manual wiring is required.

2.1.1.2 EVK1104

Connect the EXTBT J2 connector (labeled EVK1104) to the EVK1104 connector labeled J32.

Figure 2-2. EVK1104.



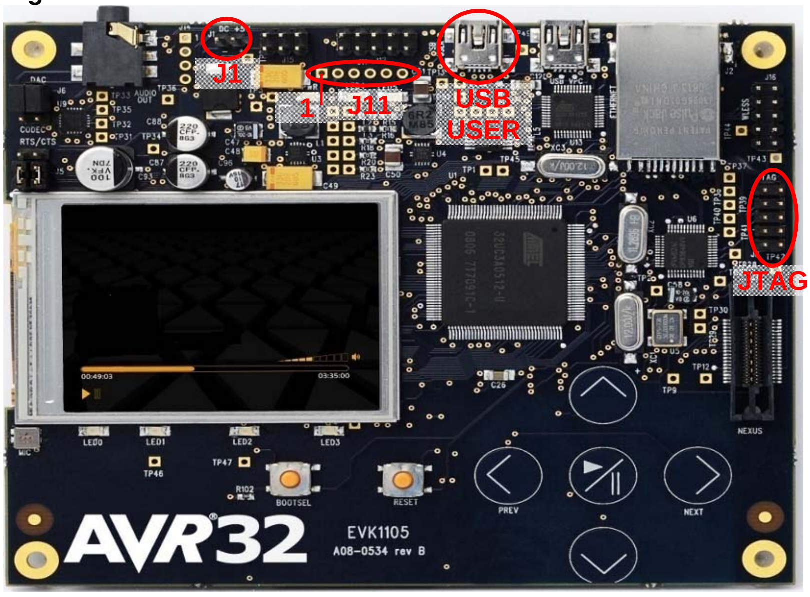
Note: As the Atmel AT32UC3A3256 PA5 (USART1.RXD) and PA6 (USART1.TXD) are already connected to the board controller MCU (U12 AT32UC3B1256), the Atmel EVK1104 must not be powered via the USB_VPC connector. Only the USB user connector or the J2 connector (5V) must be used.

If the communication with the Atmel EXTBT fails, it may be required to erase the flash program of the Atmel AT32UC3B1256 MCU. To do so, connect a JTAG debugger on the J33 header, open a shell and type: `avr32program chiperase`. If you later need to reprogram the AT32UC3B1256 firmware, use the project from the AVR Software Framework (folder: avr32\applications\evk11xx-uc3b-virtual-com-port).

2.1.1.3 EVK1105

Connect the EXTBT J2 connector (labeled “EVK1105”) to the Atmel EVK1105 connector labeled J11.

Figure 2-3. EVK1105.



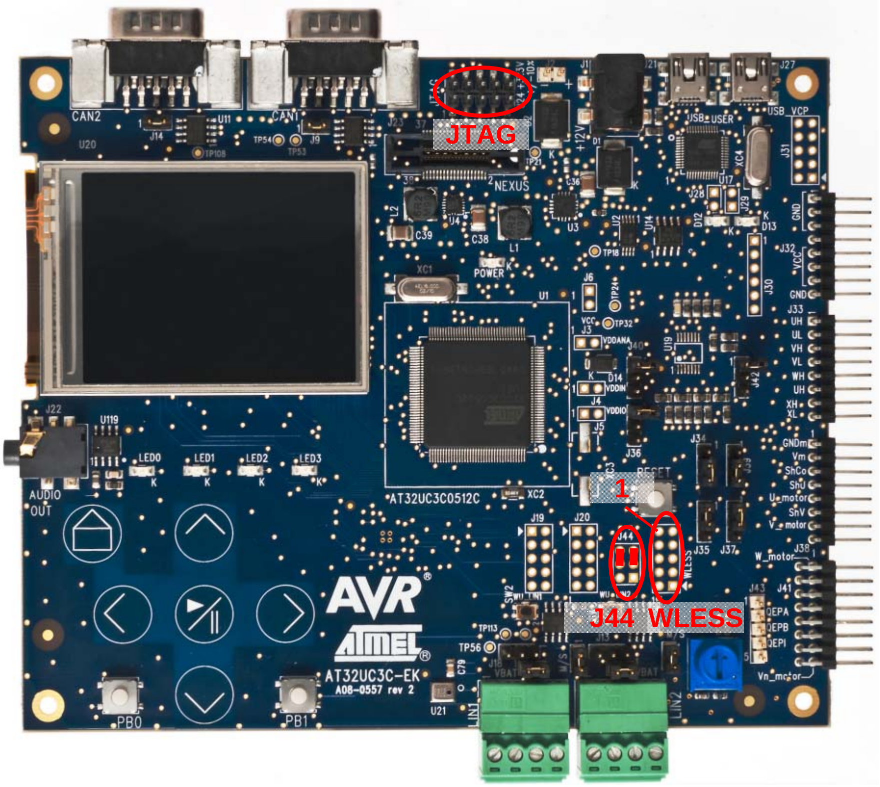
Note: As the Atmel AT32UC3A0512 PA0 (USART1.RXD) and PA1 (USART0.TXD) are already connected to the board controller MCU (U13 AT32UC3B1256), the Atmel EVK1105 must not be powered via the USB_VPC connector. Only the USB user connector or the J1 connector (5V) must be used.

If the communication with the Atmel EXTBT fails, it may be required to erase the flash program of the Atmel AT32UC3B1256 MCU. To do so, connect a JTAG debugger on the J12 header, open a shell and type: `avr32program chiperase`. If you later need to reprogram the AT32UC3B1256 firmware, use the project from the AVR Software Framework (folder: avr32\applications\evk11xx-uc3b-virtual-com-port).

2.1.1.4 UC3C-EK

Connect the EXTBT J1 connector (labeled WLESS) to the Atmel UC3C-EK WLESS header. The J44 header should be configured with two jumpers, as described in Figure 2-4.

Figure 2-4. UC3C-EK.



2.1.2 Loading the firmware on the AVR Board

The avr32798-*version*.zip provided with this application note contains the firmware binaries for each kit described above.

To program the AVR device using the AVR GNU Toolchain and the JTAG interface, power the kit, plug your favorite debugger onto the JTAG connector, and open a Dos or UNIX® shell, and type:

```
avr32program program –finternal@0x80000000 file.elf –evr –O0x80000000
```

where file.bin is the target kit name.

Programming scripts are also available (see uc3-bluetooth-t2bf-spp-demo-x .cmd and .sh, where “x” is the kit name).

To program the AVR device using the USB DFU boot loader and the USB interface, refer to the application note: AVR32784 AVR UC3 USB DFU boot loader.

Note: If the kit AVR MCU was already loaded with the Atmel boot loader, erase the whole flash (including the boot loader) with the command: `avr32program chiperase.`

The firmware source code is delivered in the AVR Software Framework (ASF, version 2.1 or higher); see in folder avr32\applications\uc3-bluetooth-t2bf-spp-demo.

2.1.3 Demonstration setup

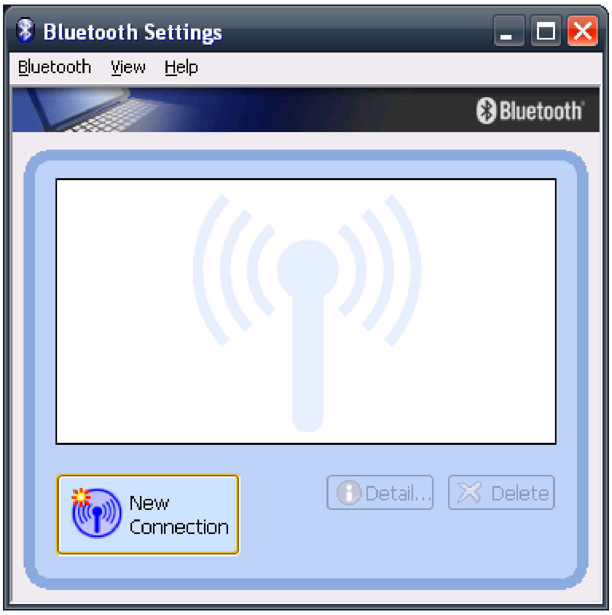
The demonstration setup is depends on the AVR board and the computer used.

- Press the reset button of the AVR board. The “App OK” message should be displayed on the screen. It means the connection with the EXTBT module is successful. An “App failed” message means a communication failure with the EXTBT module. In that case, check the EXTBT connection to the AVR board.

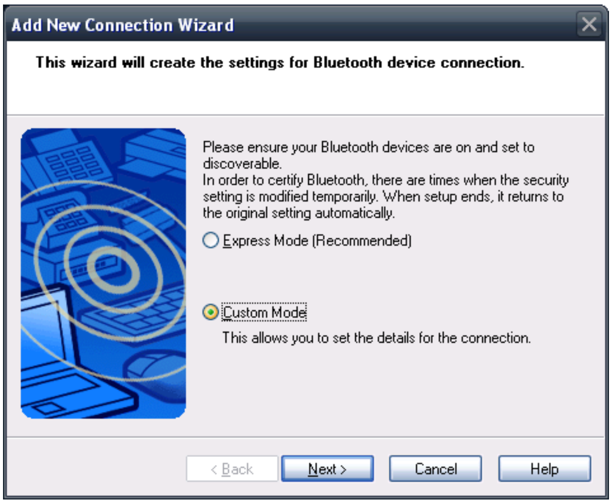
- Install a Bluetooth manager for your operating system

The following section describes the configuration of a Bluetooth manager on a DELL® laptop PC using Windows® XP and a Toshiba® Bluetooth card D360.

- Launch the application, click on “New Connection,” and search for new Bluetooth devices



- Select “Custom Mode” and search for new device



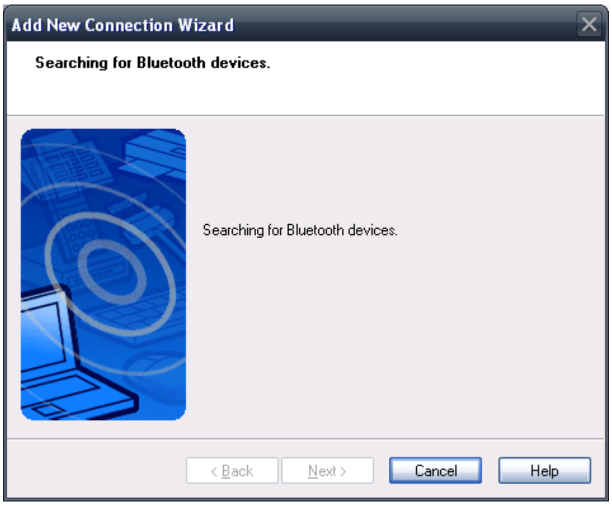
This wizard will create the settings for Bluetooth device connection.

Please ensure your Bluetooth devices are on and set to discoverable.
In order to certify Bluetooth, there are times when the security setting is modified temporarily. When setup ends, it returns to the original setting automatically.

○ Express Mode (Recommended)

● Custom Mode

This allows you to set the details for the connection.

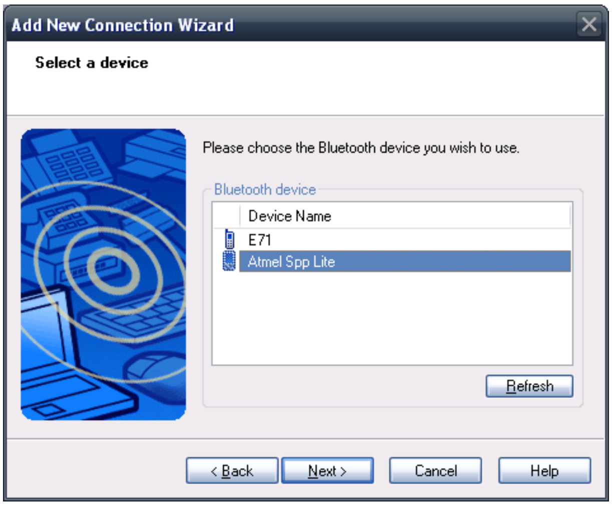


Add New Connection Wizard

Searching for Bluetooth devices.

Searching for Bluetooth devices.

- Select the “Atmel Spp Lite” Bluetooth device



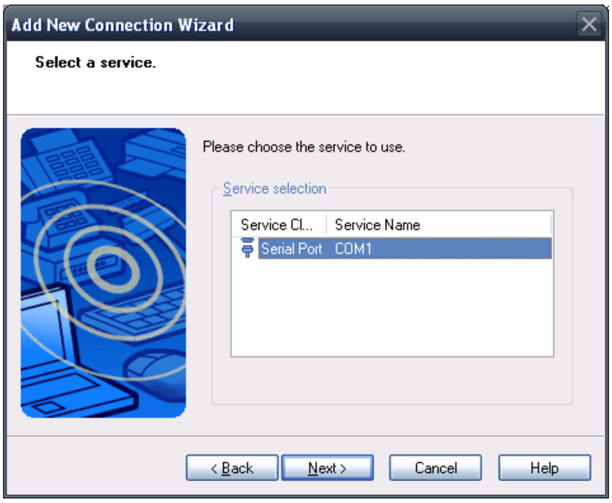
Add New Connection Wizard

Select a device

Please choose the Bluetooth device you wish to use.

Bluetooth device

Device Name
E71
Atmel Spp Lite



Add New Connection Wizard

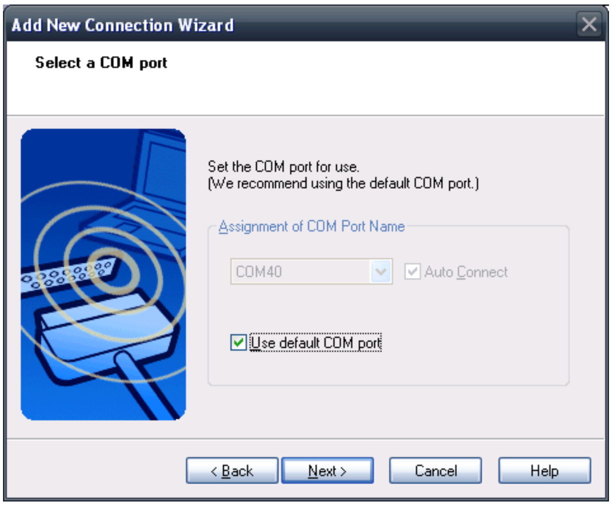
Select a service.

Please choose the service to use.

Service selection

Service Cl...	Service Name
Serial Port	COM1

- Setup the COM port to use



Add New Connection Wizard

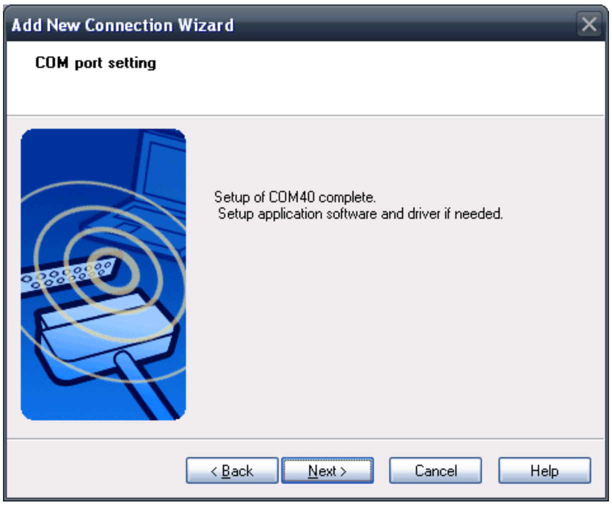
Select a COM port

Set the COM port for use.
(We recommend using the default COM port.)

Assignment of COM Port Name

COM40 ☑ Auto Connect

☑ Use default COM port



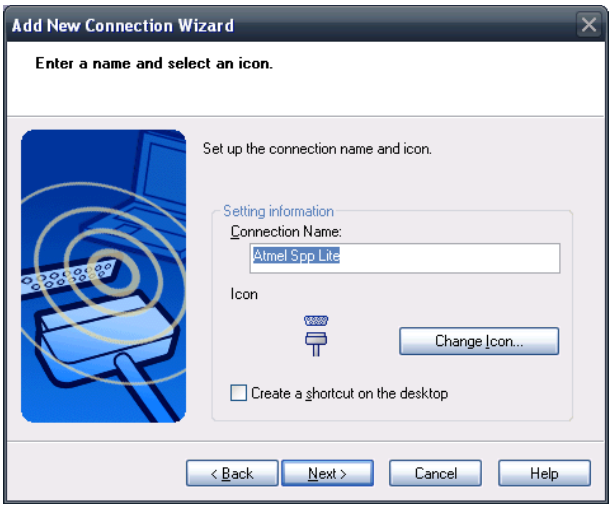
Add New Connection Wizard

COM port setting

Setup of COM40 complete.
Setup application software and driver if needed.

Once the Bluetooth device authentication has been completed, a Bluetooth serial COM port (COM 40 in our example) will be available.

- Click on “Next”, and finish



Add New Connection Wizard

Enter a name and select an icon.

Set up the connection name and icon.

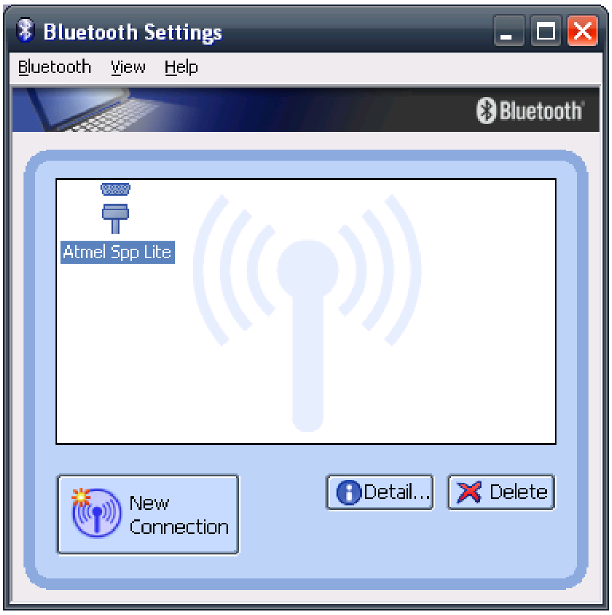
Setting information

Connection Name: Atmel Spp Lite

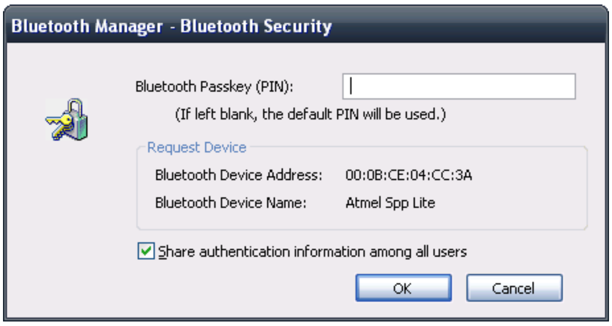
Icon

☐ Create a shortcut on the desktop

- The Atmel Spp Lite Bluetooth device can now be accessed as a COM port (40 in our example)



- Open the Bluetooth manager, connect to the Atmel Spp Lite device, and enter the following Bluetooth passkey: 0000



Bluetooth Passkey (PIN):

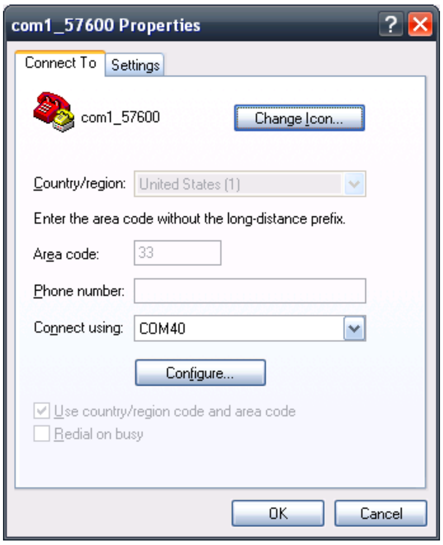
(If left blank, the default PIN will be used.)

Request Device

Bluetooth Device Address:	00:0B:CE:04:CC:3A
Bluetooth Device Name:	Atmel Spp Lite

☑ Share authentication information among all users

- Open a serial terminal connection and select and configure the Bluetooth device COM port (57600bps/8 data bits/no parity/1 stop bit/no flow control):



com1_57600

Country/region: United States (1)

Enter the area code without the long-distance prefix.

Area code: 33

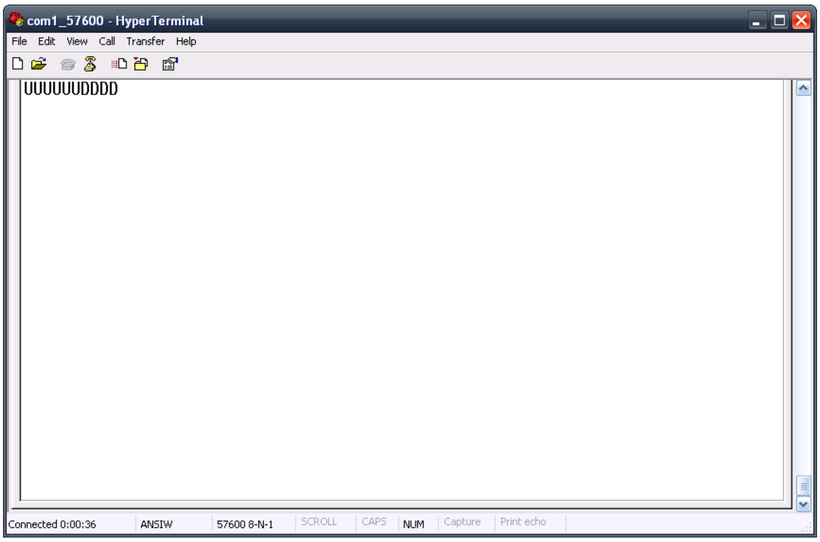
Phone number:

Connect using: COM40

☑ Use country/region code and area code

□ Redial on busy

- Typing in the serial terminal will display the message on the AVR board LCD
- Pressing some of the AVR board keys will display messages in the Hyperterminal (the UP button will display “U”, the DOWN button will display “D” etc.)



```
UUUUUUDDDD
```

It is also possible to use the AVR board buttons to establish the Bluetooth connection with the PC instead of initiating the connection with the PC Bluetooth manager.

- Pressing the inquiry function button will search for a Bluetooth connection. Expected result: The “inquiry complete” message is displayed.
- Press the device select function button until the PC computer name is displayed.
- Press the connect/disconnect function button to connect or disconnect to the PC computer respectively.

The demonstration navigation is board dependant and described in Table 2-1.

Table 2-1. EXTBT Demonstration Functions.

Kit	EXTBT Connection	Function Enquiry	Function Device Select	Function Connect/ Disconnect	Function Send Data
Atmel EVK1100	No dedicated header. Requires a manual connection	PB0	PB1	PB2	Joystick
Atmel EVK1104	J2	Touch button CS1	Touch button CS2	CS3	Touch wheel and CS4 button
Atmel EVK1105	J11	QTouch® Up	QTouch Right	QTouch Left	QTouch Center and QTouch Down
Atmel UC3C-EK	WLESS header, J44 to configure	QTouch Up	QTouch Right	QTouch Left	QTouch Center and QTouch Down

- The Atmel EXBT module is then able to send and receive data over the Bluetooth SPP connection

3 Hardware description

This section presents the hardware blocks of the Atmel EXTBT design.

3.1 Block diagram

Figure 3-1. Block diagram.

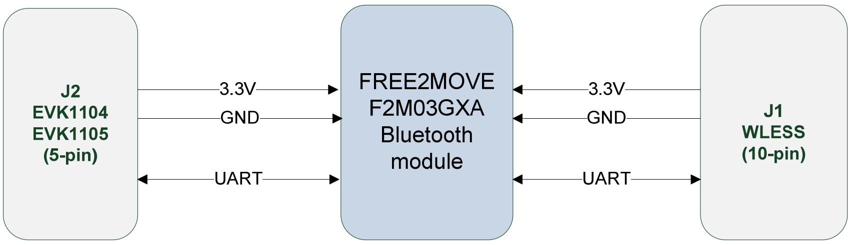


Figure 3-2. EXTBT Top view.

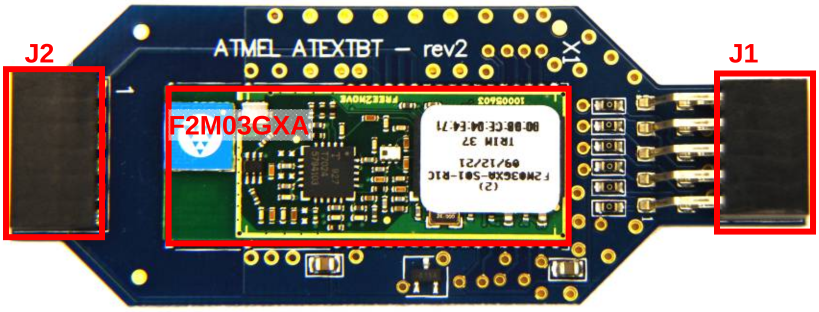


Table 3-1. J1 (10-pin) header description.

Pin	Function
1	F2M03GXA UART RTS
2	F2M03GXA UART CTS
3	F2M03GXA UART RX
4	F2M03GXA UART TX
5	F2M03GXA test SPI CS
6	F2M03GXA test SPI MOSI
7	F2M03GXA test SPI MISO
8	F2M03GXA test SPI SCK
9	GND
10	VCC 3.3V

Table 3-2. J2 (6-pin) header description.

Pin	Function
1	VCC 3.3V
2	F2M03GXA UART RTS
3	F2M03GXA UART CTS

Pin	Function
4	F2M03GXA UART RX
5	F2M03GXA UART TX
6	GND

3.2 F2M03GXA Bluetooth module

The Bluetooth communication module is the FREE2MOVE F2M03GXA (http://www.free2move.se/index.php?type=view&pk=25).

- Fully qualified end product with Bluetooth v2.0+EDR, CE and FCC
- Transmit power up to +18dBm
- Range: Up to 1000m (line of sight)
- Enhanced data rate (EDR) compliant with v2.0.E.2 of specification for both 2Mbps and 3Mbps modulation modes
- Serial interface up to 4Mbps
- Supports many Bluetooth profiles
- Support for 802.11b/g coexistence

3.3 Schematics

The Atmel EXTBT schematics can be found on www.atmel.com/avr.

4 Firmware description

The firmware source code is delivered in the Atmel AVR Software Framework (ASF version 2.1 or higher); see in folder avr32\applications\uc3-bluetooth-t2bf-spp-demo for more details. Documentation is hmtl based and included in the software package.

4.1 UART configuration

The UART configuration is done by editing the file uart_config.c file in the \avr32\applications\uc3-bluetooth-t2bf-spp-demo\conf folder of the ASF.

4.2 Cybercom Bluetooth stack

The firmware is based on the Cybercom Bluetooth 32-bit AVR freeBT stack and SPP profile (serial port profile). The AVR32 freeBT is a subset of t2BF, the Cybercom Telematics Bluetooth Framework.

General:

- No multipoint in Lite version, only one-to-one device
- Foot print range: 70kb ROM, <20kb RAM (IAR EW32 V3.x)
- Power saving with sniff mode
- Simple to program with function-based API, no complex state machines needed in application

GAP (Generic Access Profile):

- Inquiry for remote devices, maximum 10 results
- Name search

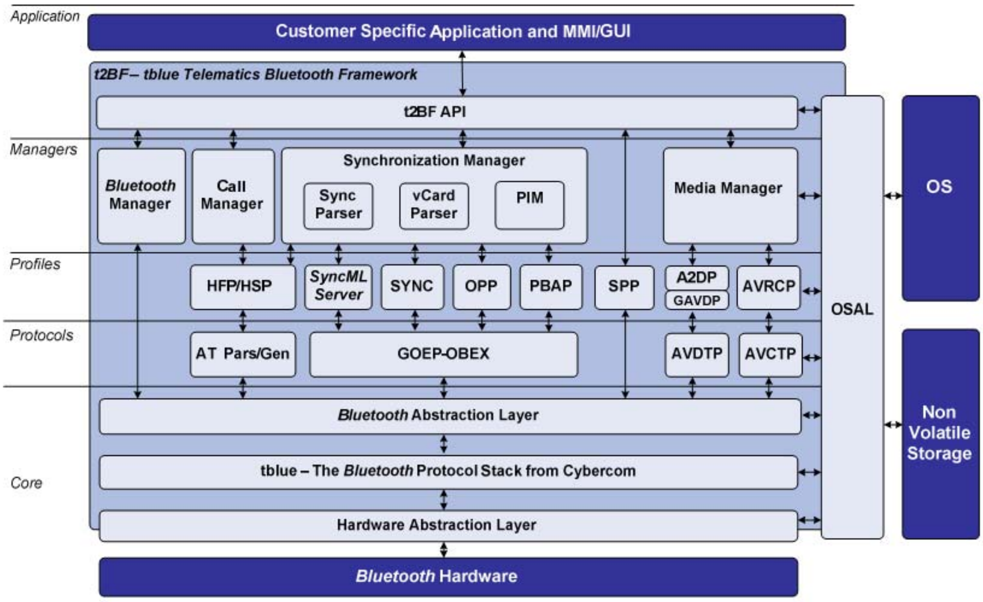
- Set device name
- Set class of device
- Security mode 2, PIN code can vary
- Set page and link timeouts
- Read RSSI
- Pairing and peer database (maximum two peers), stored in NVS

SPP (Serial Port Profile):

- SPP server port, maximum one possible. SDP service record containing service name
- Adjustable
- SPP client port, maximum one possible
- One port on one link possible
- Credit based flow control, three local credits per port
- Send and receive functions

The 32-bit AVR freeBT is a subset of t2BF, the Cybercom Telematics Bluetooth Framework.

Figure 4-1. Cybercom Telematics Bluetooth Framework.



4.2.1 Bluetooth compliance

t2BF, with all its packages and specified functionality, is compatible with the Bluetooth Specifications v1.0 or higher official released by the Bluetooth SIG (Special Interest Group).

32-bit AVR freeBT is based on a Bluetooth stack core named Cybercom tblue. This stack is today a Bluetooth v2.0+EDR qualified and listed subsystem with Bluetooth Identifier B013600. t2BF and its framework, including stack, profiles, etc., are tested with the PTS (Profile Tuning Suite) from the Bluetooth SIG.

The tests have been performed using a demo application communicating with the t2BF API. The t2BF framework in different constellations is also running in final qualified and Bluetooth listed products available on the market. The framework is also continuously tested with different mobile devices to secure interoperability. A well defined test matrix has been worked out by Cybercom, and all tests performed with different devices are stored in an SQL database. A web site is available showing a subset of the test results (see http://tblue.west.cybercomgroup.com).

5 References and further information

What is the next step?

If you want more detailed information concerning the hardware of this kit, if you want to check out all existing application notes and software running on this kit, if you want to start a development, whether for evaluation or rapid prototyping, the following sections lists the main available resources to actually get beyond getting started:

- Schematics and BOM
- Application notes
- Software Framework
- Tools
- Support

5.1 Detailed hardware references (and associated errata)

More detailed hardware information for this kit can be found on the Atmel web page dedicated to this kit: www.atmel.com/extbt.

5.2 Software framework

To start a development, whether for evaluation or rapid prototyping, the Atmel AVR Software Framework is a complete and useful source of existing source code a developer can start from. It has been designed to help develop and glue together the different components of a software design, and to be easily integrated into an operating system (OS), as well as to operate in a standalone way.

The 32-bit AVR Software Framework is available in two formats:

- As a standalone zip package containing IAR™ projects and GNU makefile projects, downloadable from this Atmel web page: http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4192. This web page also contains all AVR Software Framework-related application notes
- As a plug-in in Atmel AVR32 Studio that allows the user to create pre-defined example AVR32 Studio projects and add pre-written drivers to existing AVR32 Studio projects: http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4116 This web page also contains all AVR32 Studio-related application notes

To be able to build the software example program the binary, and step through the code in debug mode, other tools are necessary. References to these tools are provided in the Tools chapter.

5.3 Tools

To be able to develop applications for 32-bit AVR devices build binaries for 32-bit AVR targets, and program a 32-bit AVR device, Atmel and its partners provide several tools supported on multiple host targets.

5.3.1 IDE and compilers: Design software

- Atmel AVR32 Studio: AVR32 Studio is a free integrated development environment (IDE) for 32-bit AVR that enables you to write, build, deploy, and debug your C/C++ and assembler code. It also includes the 32-bit AVR Software Framework plug-in. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4116. This web page also contains all AVR32 Studio-related application notes
- Atmel AVR32 GNU Toolchain: AVR32 GNU Toolchain is a set of standalone command line programs used to create applications for 32-bit AVR microcontrollers. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4118
- IAR Embedded Workbench®: IAR Embedded Workbench with its optimizing C and C++ compiler provides full support and generates very compact and efficient code for 32-bit AVR devices of the AP7000 and UC3 families. http://www.iar.se/website1/1.0.1.0/124/1/

5.3.2 Programming and on-chip debugging tools: Debug tools

- Atmel AVR JTAGICE mkII: The AVR JTAGICE mkII from Atmel is a powerful development tool for on-chip debugging of all AVR 8-bit RISC MCUs and 32-bit AVR DSP/MCUs with IEEE 1149.1 compliant JTAG interface. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3353
- Atmel AVR ONE!: The AVR ONE! is a powerful development tool for on-chip debugging and programming of all 32-bit AVR and AVR XMEGA® devices. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4279
- Atmel AVR Dragon: The AVR Dragon sets a new standard for low-cost development tools. AVR Dragon supports all programming modes for the AVR device family. http://www.atmel.com/dyn/Products/tools_card.asp?tool_id=3891

5.4 Support

Atmel has several support channels available:

• Web portal:	http://support.atmel.no/	All Atmel microcontrollers
• Email:	avr@atmel.com	All AVR products
• Email:	avr32@atmel.com	Al 32-bit AVR products

Please register on the web portal to gain access to the following services:

- Access to a rich FAQ database
- Easy submission of technical support requests
- History of all your past support requests
- Register to receive Atmel microcontrollers' newsletters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Milennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chou-ku, Tokyo 104-0033
JAPAN
Tel: (+81) 3523-3551
Fax: (+81) 3523-7581

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.