



Contents

1. Overview	4
2. Glossary	7
3. Features	8
4. Hardware description	10
4.1. Radio modules	10
4.1.1. deRFarm7	10
4.1.2. deRFmega128	11
4.2. Application platforms	12
4.2.1. deRFgateway board	12
4.2.2. deRFnode boards	13
4.2.3. Connecting the radio modules to the application platforms	14
4.3. deRFusb-23E00 JTAG radio stick (JTAG analyzer)	16
4.4. deRFextract Tool	17
4.5. SAM-ICE adapter	18
4.6. Accessories	18
4.6.1. Cables	18
4.6.2. Power supply and adapters	19
4.6.3. CD content	19
5. Preparing and Running the deCON6 application	20
5.1. Java Installation and deCON6 installation	21
5.2. Connecting coordinator and nodes to a network	22
5.2.1. Prepare network interface	22
5.2.2. Connect coordinator	24
5.2.3. Power up the nodes	25
5.2.4. Using deCON6	26
5.2.5. Configure node visualization	27
5.2.6. I/O Settings tab	29
5.2.7. User data tab	30
5.2.8. Ping tab	32
5.2.9. Show network configuration	33
5.2.10. Monitoring node activity	34
5.3. USB driver installation	35
6. Developing custom network applications and programming	38
6.1. Software packages	38
6.1.1. Native examples	38
6.1.2. MAC-based examples	41
6.1.3. 6LoWPAN-based examples	44
6.1.4. IPv4 to IPv6 gateway example	47
6.2. Development tools	50
6.2.1. AVR Development	51
6.2.2. ARM Development	62
7. Analyzer	73
7.1. Perytons Analyzer	73
7.2. BitCatcher	75
8. Revision notes	76
9. References	76



Document history

Date	Version	Description
2013-04-30	1.0	Initial version



1. Overview

This document is intended for engineers and software developers evaluating and operating sensor networks designed in compliance to 6LoWPAN specifications and the IEEE 802.15.4™ standard.

The deRFdevelopmentKit 6LoWPAN 2.4 GHz serves as a development environment for quick and easy startup, development and evaluation of 6LoWPAN sensor networks.

The kit contains a complete development environment for a 6LoWPAN sensor network. If it is required to operate more than the two nodes of the kit, you can order additional deRFnodes and deRFmodules to connect as many sensors or actuators necessary for your setup. The kit provides all the software you need to evaluate and develop nodes that operate within a 6LoWPAN sensor network.

To use 6LoWPAN in IPv4 environments the coordinator converts the wired IPv4 packets into wireless IPv6 bidirectionally.

The kit CD contains a comprehensive software package with many example applications that cover the typical IEEE 802.15.4 features from simple point-to-point connections to network applications with routing and beacons. Documentation, data sheets and manuals of the respective hard- and software components are also included on the kit CD.

The nodes are delivered as coordinator, router and end device with the 6LoWPAN demo application Control4SLOW and can be controlled and monitored via the deCON6 tool provided on the kit CD.

With the included intuitively to operate GUI deCON6 the user can get information about the network structure and the individual sensor nodes in only a few steps. Additional customized components can be connected and controlled via GUI without software changes. deCON6 runs the complete communication to all nodes within the network. Thus the status of every individual node can be displayed as well as the current battery condition and the defined user signals.

Included in the deRFdevelopmentKit is an analyzer stick (USB radio stick) with a 30-day trial license of Perytons network and protocol analyzer software (full version). With Perytons analyzer tool low level data transport and network structures can be observed. The different protocol modules allow the graphical representation of the packet content for each layer of a wireless radio stack.

A basic illustration of a 6LoWPAN network shows **Figure 1**.

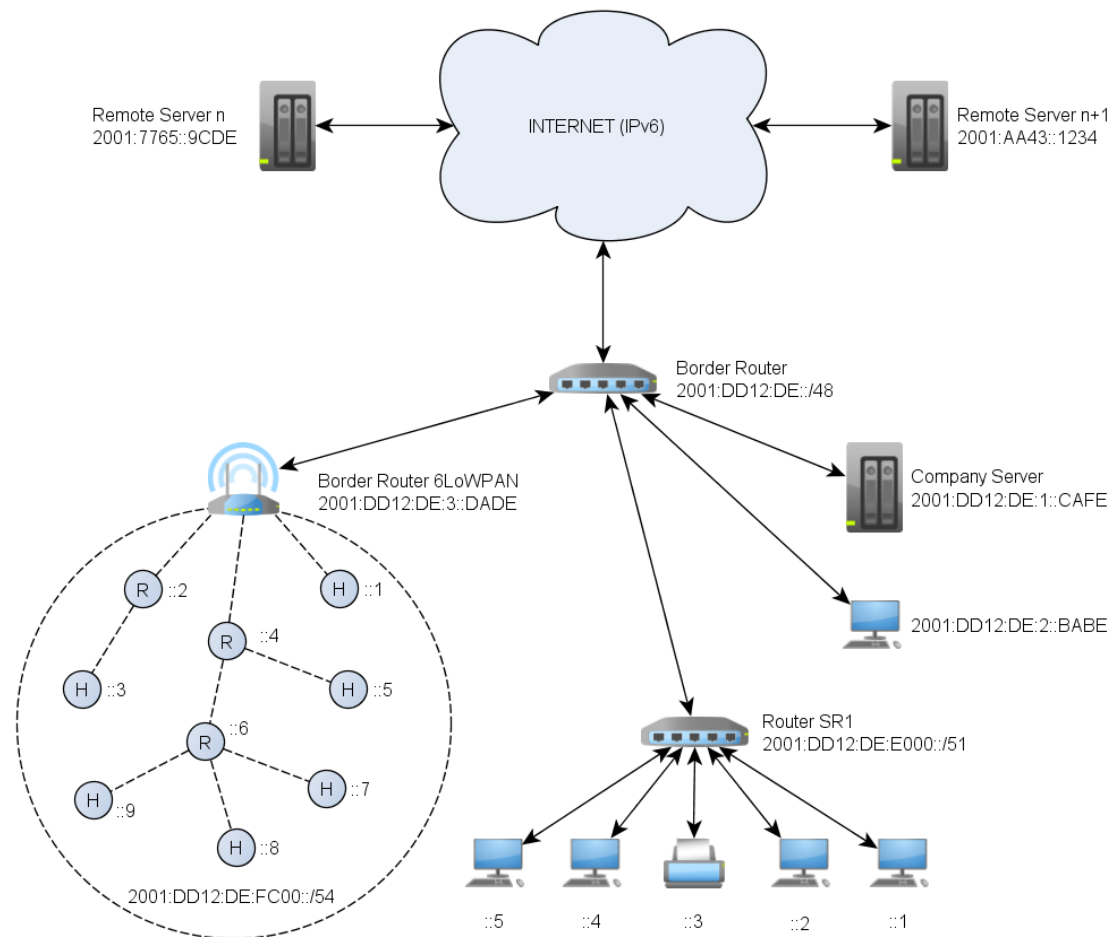


Figure 1: 6LoWPAN network structure

Significant is the border router which acts as 6LoWPAN gateway and connects the WSN to the global IP infrastructure.

6LoWPAN is an acronym for IPv6 over Low Power Wireless Personal Area Networks (with low energy consumption); it is also used as synonym for the communication protocol. 6LoWPAN offers a smart possibility to apply IPv6 on 802.15.4 based nodes. The idea behind this approach is to create an option to integrate sensor networks into the already existing IP infrastructure. Applying 6LoWPAN the effort of gateways (which execute the protocol implementation) is omitted.

6LoWPAN offers a transport layer for different applications by integrating well known IP protocols into wireless sensor nodes allowing a transparent connection between wireless and existing IP networks. Each node can have a world-wide unique IP address and can be monitored online. A larger range can be achieved by using routers. With the 6LoWPAN solution from dresden elektronik the user can start the wireless communication between the devices without any programming.

By using IPv6 6LoWPAN is optimally suited to embed radio link and sensor networks respectively into an existing IP infrastructure, to link e.g. individual sensor nodes or networks to the internet. 6LoWPAN is interesting in the field of industrial automation, too; it is increasingly applied in the field of machine-to-machine communication.



deRFtraining – Workshop Smart Wireless Solutions

For your introduction into wireless network communications dresden elektronik offers one-day workshops. Based on the development kit 6LoWPAN 2.4 GHz you will be introduced to the most important theoretical basics and learn how a 6LoWPAN sensor network is set up and activated using the development kit and how own sensors can be implemented.

Further information about workshop dates can be found at <http://www.dresden-elektronik.de/funktechnik/news>.



2. Glossary

Term	Description
6LoWPAN	IPv6 over Low-power Wireless Personal Area Networks
BSP	Board Support Package, for e.g. MAC Stack Addon
Coordinator	FFD device which routes packets and can integrate new nodes into the network
FTDI	Future Technology Devices International
GNU	GNU General Public License, the most widely used free software license.
GUI	Graphical User Interface
Hop	Stopover of a route as well as the way from one network node to the next
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IEEE 802.15.4	Standard applicable to low-rate Wireless Personal Area Networks (WPAN)
IPv6	Internet protocol version 6, version of the Internet Protocol (IP) intended to succeed IPv4 which is the protocol currently used to direct almost all internet traffic
MAC	Medium Access Control ... layer, address etc.
OSI	Open Systems Interconnection (OSI) model, a prescription of characterizing and standardizing the functions of a communications system in terms of abstraction layers
PAL	Platform Abstraction Layer
PHY	OSI model layer 1: The physical layer defines electrical and physical specifications for devices. It defines the relationship between a device and a transmission medium including the layout of all hardware components.
Router	A device that forwards data packets between wireless and computer networks
TAL	Transceiver Abstraction Layer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

3. Features

The following figure gives an overview of the parts contained in the deRFdevelopmentKit 6LoWPAN 2.4 GHz.



Figure 2: 6LoWPAN development kit content



The kit consists of:

- 1 x radio module deRFarm7-25A00 in its initial state, marker on the bottom side: black
- 2 x radio module deRFmega128-22A00 in their initial state, markers on the bottom side: 1x fluorescent red, 1x green
- 1 x development board deRFgateway-1TNP2
- 2 x development board deRFnode-2TNP2
- 1 x USB radio stick analyzer, deRFusb-23E00 JTAG, 2.4 GHz
- 1 x SAM-ICE adapter
- 1 x deRFextract Tool
- 1 x Kit CD (with extensive software package and documentation)
- 1 x CAT5 Ethernet cable
- 2 x USB cable, each with standard A and B plug
- 1 x power supply, 5 x adapter

Features:




- Development environment for 6LoWPAN applications
- Designed for frequency range of 2.4 GHz
- Support for user-specific sensors, the connection of own sensors/nodes is possible
- Kit CD with many software examples, documentation and necessary drivers
- Analyzing tool for IEEE 802.15.4, ZigBee and 6LoWPAN
- Example applications for 6LoWPAN
- Freely programmable USB radio stick



4. Hardware description

4.1. Radio modules

The nodes of the kit (with preinstalled firmware for different functionality, see **Section 5**) can be identified at the bottom side by their color; the radio modules are labeled each with a marker: 1x black, 1x fluorescent red, 1x green; the colors stand for coordinator, router and end device.

- Coordinator 
- Router 
- End Device 

Connecting the radio modules to the deRFgateway-1TNP2 or deRFnode-2TNP2 is described in **Section 4.2.3**.

4.1.1. deRFarm7

The compact designed pluggable radio module deRFarm7-25A00 contains a powerful ARM7 microcontroller with 512 kBytes high-speed flash, on-chip USB 2.0 full speed transceiver, Ethernet MAC 10/100 base-T in RMII-Mode and an onboard transceiver for 2.4 GHz.

The 46-pin interface gives access to most hardware functions of the microcontroller. The 2.4 GHz deRFarm7 is able to cover up to 200 m (650 feet) for line of sight conditions with its ceramic chip antenna. A 128-bit AES encryption unit is installed. The 512 kB Flash and 128 kB RAM of the deRFarm7 module provides enough resources to be used for any tasks within a wireless sensor network.



Figure 3: deRFarm7-25A00 top view



bottom view (smaller sized)

Note: From the top view the deRFarm7 and the deRFmega128 look alike. They can only be distinguished from the bottom side by the colored marker (e.g. black for coordinator) and the printed label.

For technical data and detailed information please refer to **[1]** and **[2]**



4.1.2. deRFmega128

The pluggable compact radio module deRFmega128-22A00 includes Atmel's single-chip ATmega128RFA1 microcontroller, which combines an 8-bit AVR microcontroller with a 2.4 GHz transceiver.

The radio module's 46-pin interface allows full access to all functions of the ATmega128RFA1. With the chip ceramic antenna distances of more than 200 m (line of sight) can be reached. The integrated transceiver uses 128-bit AES encryption. A serial 1-Mbit-EEPROM offers high memory capacity e.g. for a firmware update over-the-air.



Figure 4: deRFmega128-22A00 top view



bottom view, (router, smaller sized)

For technical data and detailed information please refer to [3] and [4]

4.2. Application platforms

Together with the radio modules the dresden elektronik deRFgateway and deRFnode series platforms are optimally suited to run 6LoWPAN wireless networks.

Note: For powering the deRFgateway/deRFnode board: Always check or set the correct jumper position (JP2); in **Figure 5** the JP2 header power supply options are shown. In **Figure 6** and **Figure 7** the jumper is set to 'USB powered' (external power supply).

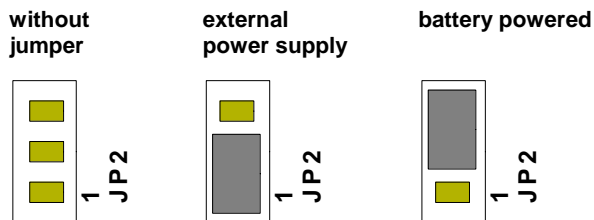


Figure 5: Power supply jumper setting options on deRFgateway/deRFnode board

For more information please refer to, [5], [6] and [7].

4.2.1. deRFgateway board

The deRFgateway-1TNP2 is a demonstration and application platform for the ARM7 based dresden elektronik radio modules. It supports ARM programming and communication over Serial, USB and Ethernet interface.

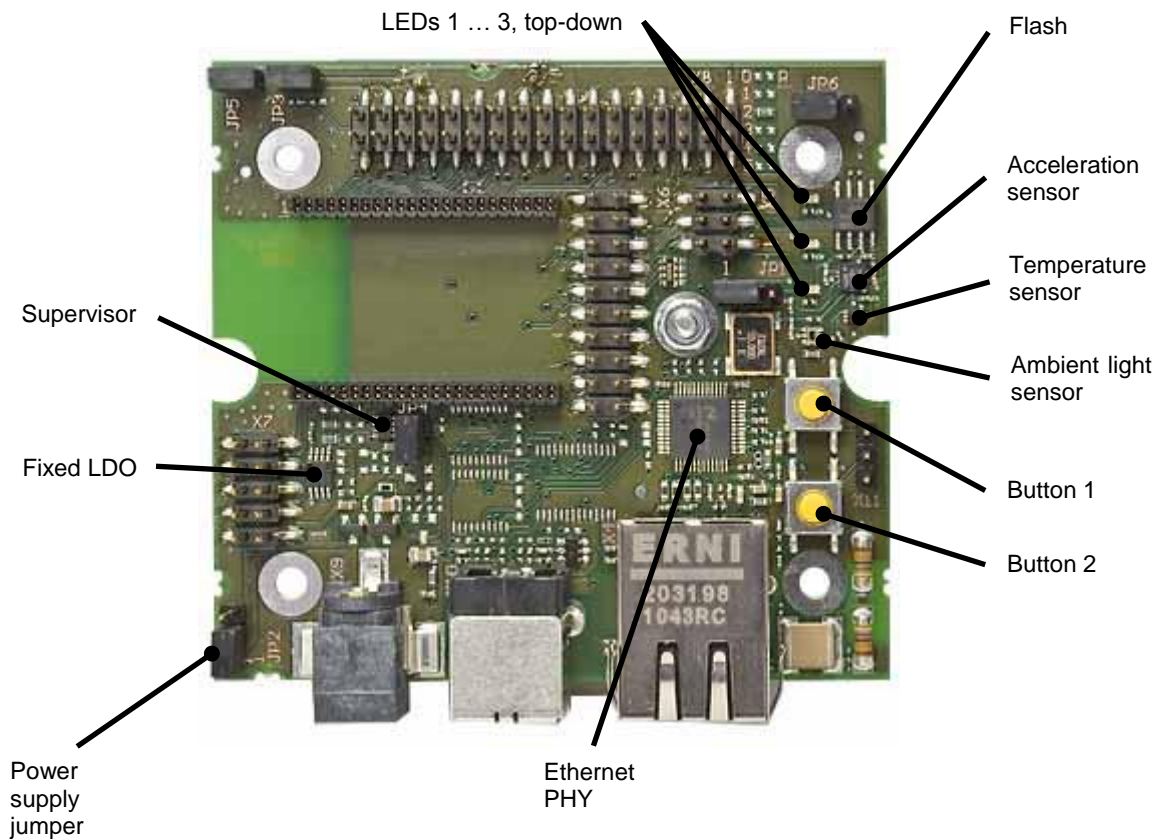


Figure 6: deRFgateway-1TNP2 – board features

The deRFgateway-1TNP2 is a demonstration and application platform for ARM based dresden elektronik radio modules. It supports ARM programming and communication over Serial, USB and Ethernet interface.

The deRFgateway platform is equipped with several other onboard features such as three different sensors (temperature, ambient light and acceleration), two user defined buttons and three LEDs. The USB interface connects directly to the USB port of the ARM based dresden elektronik radio modules. A supervisor and power supply monitoring complete the available board features.

4.2.2. deRFnode boards

The deRFnode-2TNP2 is a demonstration and application platform for the AVR and ARM based dresden elektronik radio modules. It supports AVR and ARM programming and communication over Serial and USB interface.

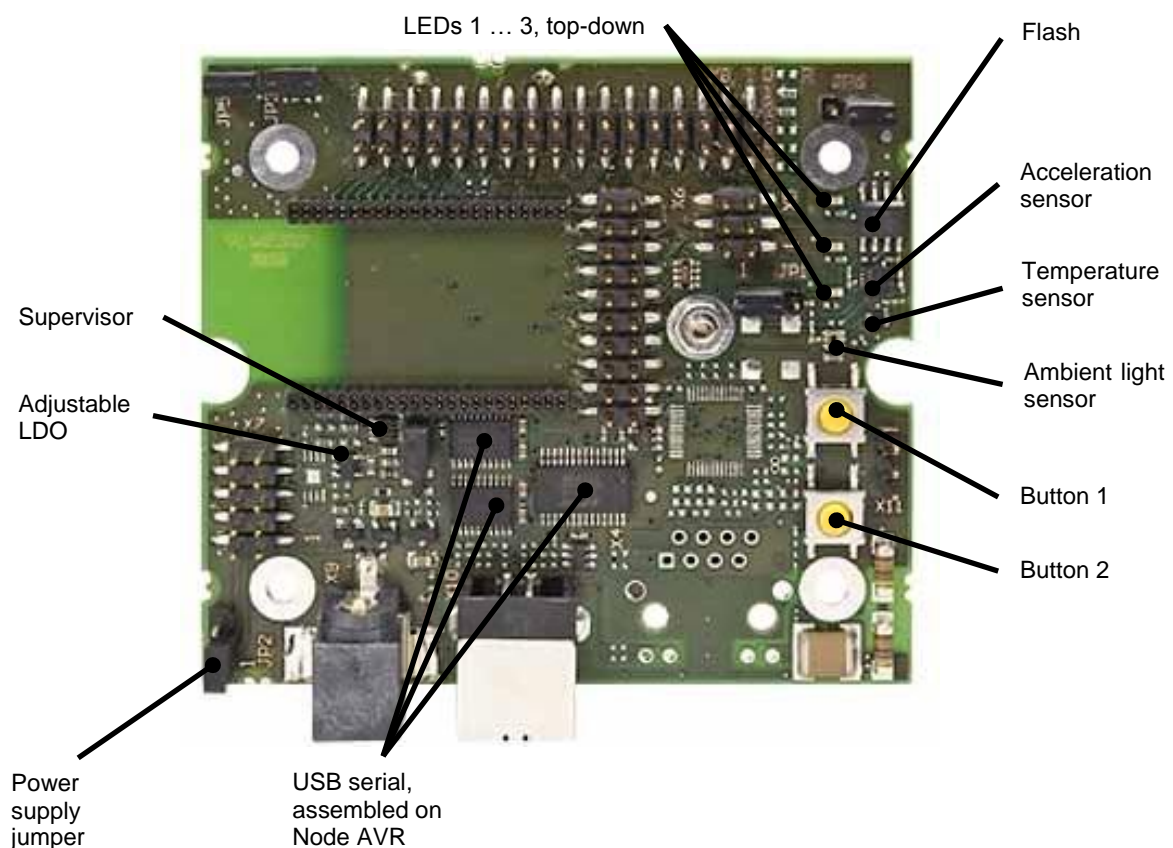


Figure 7: deRFnode-2TNP2 – board features

The deRFnode platform is equipped with several onboard features such as three different sensors (temperature, ambient light and acceleration), two user defined buttons and three LEDs, USB interface for AVR, a supply voltage supervisor and an expansion header for additional connections and own sensors/actuators.

4.2.3. Connecting the radio modules to the application platforms

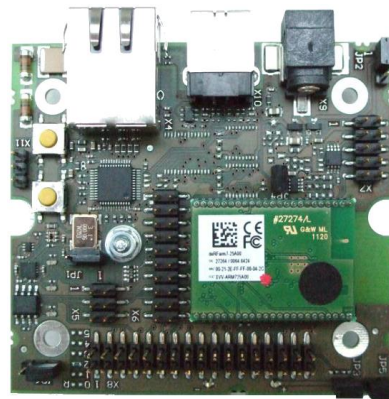
The deRFarm7-25A00 radio module has to be attached to the deRFgateway and always operates as the coordinator of the WSN. The two deRFmega128-22A00 radio modules have to be attached to the two deRFnode boards to setup the router and end device.

Required parts:

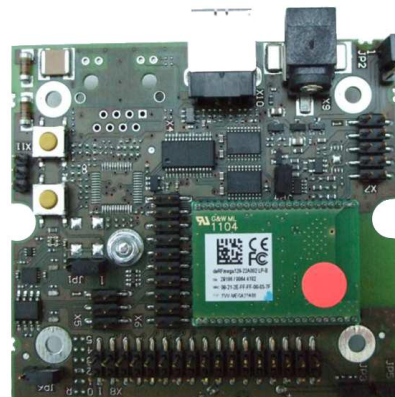
- 1 x deRFarm7-25A00
- 1 x deRFgateway-1TNP2
- 2 x deRFmega128-22A00
- 2 x deRFnode-2TNP2

For further use, note the different markers for coordinator, router and end device functionality on the radio modules, also refer to **Section 4.1**.

Coordinator
deRFarm7 + deRFgateway



Router
deRFmega128 + deRFnode



End device
deRFmega128 + deRFnode

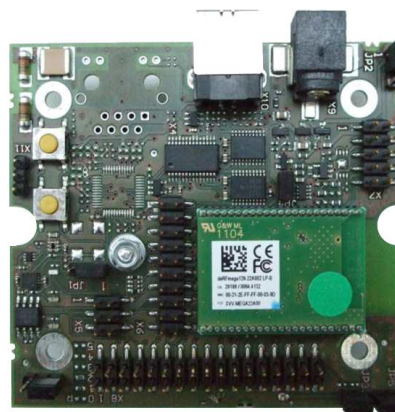


Figure 8: Application platforms with mounted radio modules

Attach the radio modules carefully - topside down - to the deRFgateway/deRFnode boards. Double-check that the chip ceramic antenna points to the outside of the board.

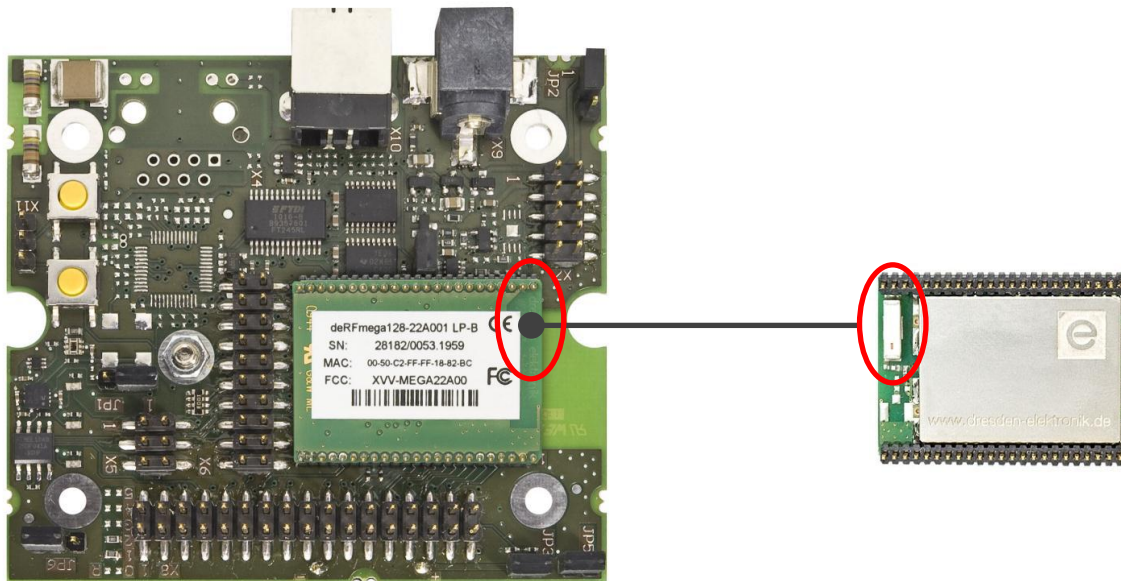


Figure 9: deRFnode with mounted radio module

To remove a radio module from the deRFgateway/deRFnode board, use the deRFextract tool, see **Section 4.4**.

4.3. deRFusb-23E00 JTAG radio stick (JTAG analyzer)

Included in the development kit is a freely programmable USB radio stick with JTAG header and analyzer firmware compatible with Perytons Network Analyzer.

The deRFusb-23E00 JTAG contains a Cortex-M3 microcontroller with 256 kBytes high-speed flash and a 2.4 GHz ISM-band transceiver. The AT86RF231 transceiver is intended for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and proprietary ISM applications; it uses a 128-bit AES encryption. With the integrated chip antenna distances of more than 200 m can be reached for line of sight conditions. The radio stick is equipped at its rear side with a 10-pin connector for development purposes (JTAG and Debug) and three LEDs for status indication.

With Perytons network and protocol analyzer software network structures, data flows and runtime performance can be monitored. The gained in-depth knowledge of the used protocols allows for optimized network configurations with little effort. For more details please refer to **Section 6.2.2** and **Section 7**.

The radio stick is labeled with a blue marker on the top side; the color stands for the analyzer functionality.

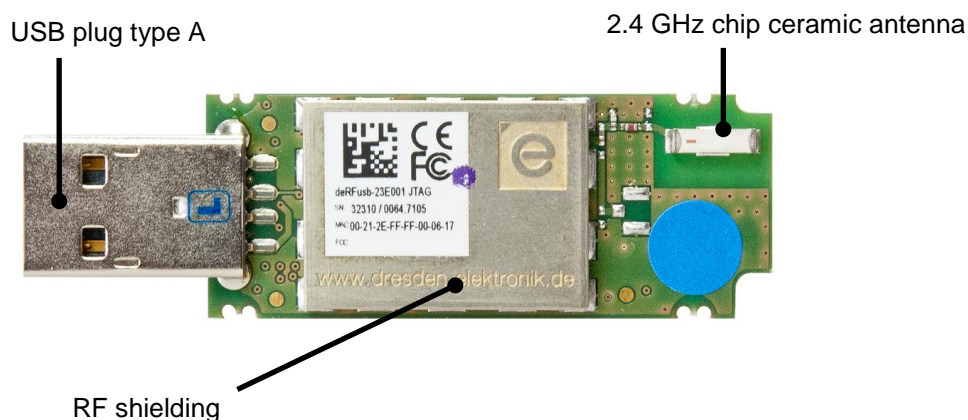


Figure 10: Top view deRFusb-23E00 JTAG

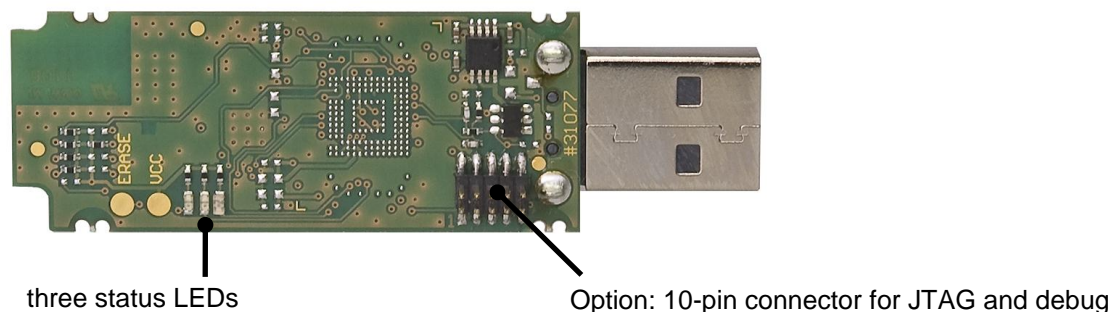


Figure 11: Bottom view deRFusb-23E00 JTAG

More information and technical background you can find in [8], [9].and [10].

Alternatively the USB radio stick can be reprogrammed with own software and used for any other application, e.g. turned into a 6LoWPAN network coordinator.

4.4. deRFextract Tool

The transparent lever part helps to remove a plugged in deRFarm7 or deRFmega128 radio module from the deRFgateway or deRFnode board.

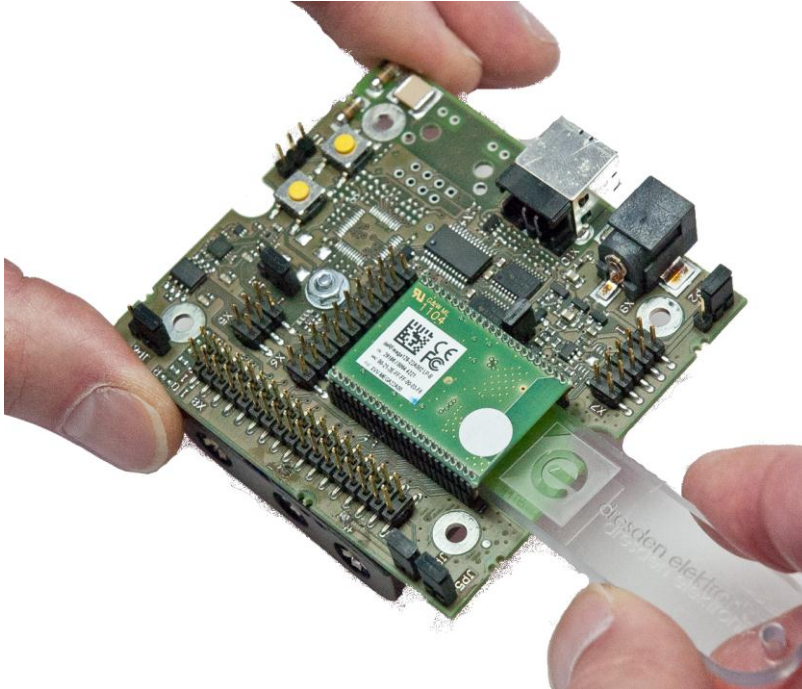


Figure 12: Using deRFextract Tool

Notes: To remove a plugged in radio module, please handle the deRFnode board, the deRFextract Tool and also the radio module with the utmost care.

Do not tilt; do not bend the pins of the radio module!

4.5. SAM-ICE adapter

The SAM-ICE adapter enables programming and debugging of the deRFusb-23E00 JTAG radio stick. It is required to connect the deRFusb-23E00 JTAG connector to the SAM-ICE programmers JTAG interface (20-pin cable) and offers a RS232 connection to the Debug Unit (DBGU) of the microcontroller.

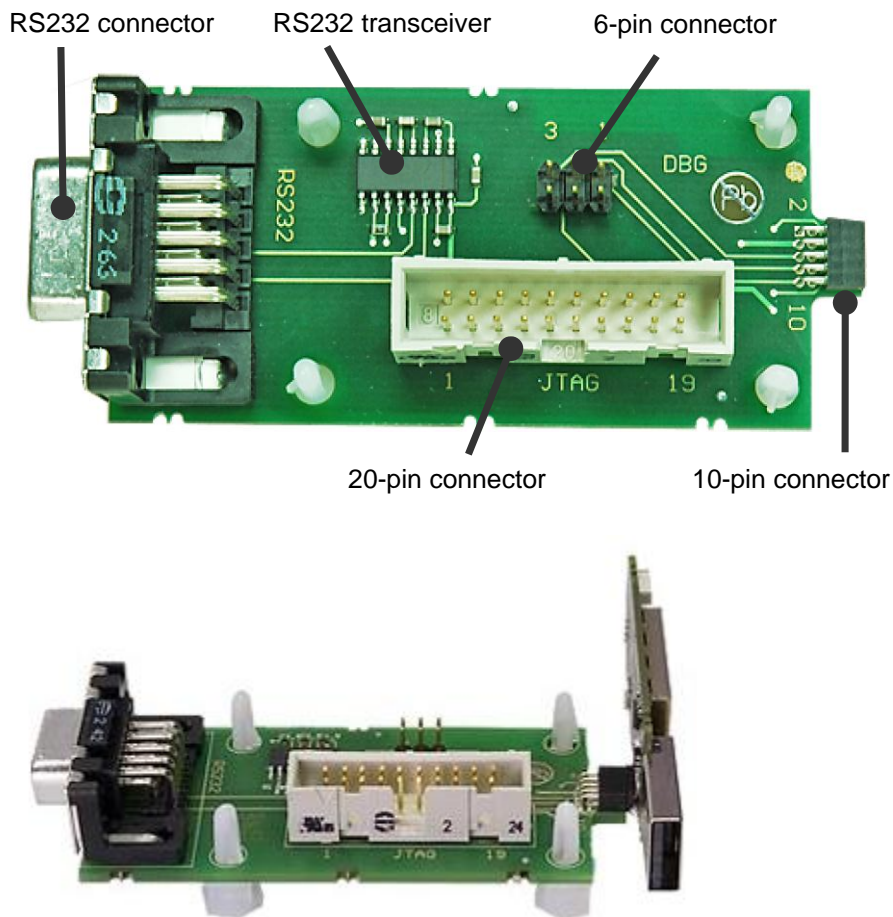


Figure 13: SAM-ICE adapter components and attached to deRFusb-23E00 JTAG stick

For more information please refer to [11].

4.6. Accessories

4.6.1. Cables

An Ethernet cable (CAT5) is available to connect the coordinator node via the deRFgateway's RJ45 socket to the wired network. The cable requires the usage of a hub or switch and cannot be used for direct connection of the deRFgateway to a PC or laptop. Please use a cross-link cable for those direct connections.

Two USB cables, each with a standard A and B plug are included. Though not required in a typical application (router and end device can be operated battery driven) the USB cables are also intended to be used as (USB) power supply as well as communication interface for debugging own applications.



4.6.2. Power supply and adapters

A power supply unit and five adapters are included within the kit. The power supply is used to feed the coordinator node via the deRFgateway's 5 VDC connector socket. The different adapters ensure operating the 6LoWPAN deRFdevelopmentKit worldwide.

4.6.3. CD content

An overview of the CD content structured in four root directories is given below.

Name	Description
3rdParty	Contains different tools and applications from third party vendors, necessary for proper working.
Doc	Contains comprehensive documentation files such as: Data sheets for products of dresden elektronik and Atmel Corporation, measurement reports and certification grants, product info overview (dresden elektronik wireless products), schematic files, manuals and documentation of the wireless modules.
Driver	Contains necessary USB drivers.
Packages	Contains software packages to run 6LoWPAN, BitCloud, deFRnative examples, MAC and Perytons analyzer.

Table 1: Development Kit CD structure






5. Preparing and Running the deCON6 application

In the following section required preparations to run the deCON6 application are described. At first the installation and startup procedures are explained. Then the connection of the coordinator to a PC and the establishment of the wireless network follow. Next the deCON6 application is used to configure the nodes. Finally an overview of LED activities as a means of observing node activity is given and the USB driver installation described.

Basic features of the deCON6 GUI are:

- deCON6 communicates with the deRFgateway (the coordinator by Ethernet (TCP) or serial interface (RS232 / USB).
- The visualization of nodes within deCON6 can be configured individually for each node. Own descriptions can be provided as XML files.
- The GPIOs of configured nodes can be controlled directly by the deCON6 GUI. No programming is required.

The nodes of the kit are preprogrammed with the firmware of dresden elektronik. As mentioned in **Section 4.1** the different devices can be identified by their marker color: black for coordinator, fluorescent red for router and green the end device.

- Coordinator 
- Router 
- End Device 

The 6LoWPAN radio network can be monitored and controlled using the deCON6 GUI.

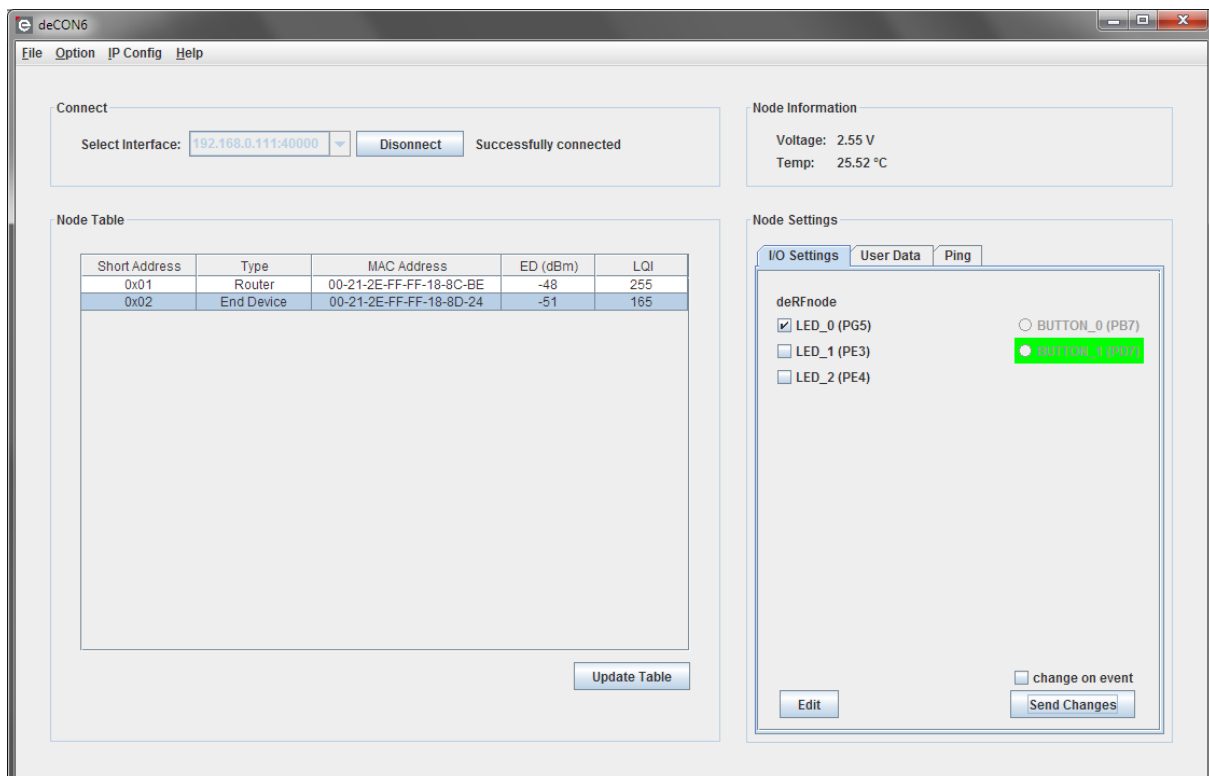


Figure 14: 6LoWPAN network monitoring

The deCON6 application is written in Java and requires Java to be installed on the computer.



5.1. Java Installation and deCON6 installation

Java installation:

To install the Java Runtime Environment, download it from the internet or use the version provided on the CD (recommended), see below. Prior to installing the Java Runtime Environment you might want to check if it is already installed on your PC. Please follow the instructions below.



To check if the Java Runtime Environment is already installed:

- Open the console 'Command Prompt': Start | Execute | cmd
- Enter: java -version

If installed the following window with the installed java version appears:

```
C:\>java -version
java version "1.7.0_04"
Java(TM) SE Runtime Environment (build 1.7.0_04-b20)
Java HotSpot(TM) Client VM (build 23.0-b21, mixed mode, sharing)
C:\>
```

Figure 15: Java installation

If not or not correctly installed an info like 'unknown or wrong command' or 'command could not be found' will be prompted.

To install Java, use the following exe files optionally from the CD:

- \3rdParty\Java\jre-6u25-windows-i586.exe (for 32-bit operating system)
- \3rdParty\Java\jre-6u25-windows-x64.exe (for 64-bit operating system)

Start the setup and follow the given instructions.

Note: To check your PC's operating system version click on the Windows Start icon:

- Control Panel | System and Security | System (Windows 7 Editions)

deCON6 installation:

The deCON6 application is provided with a setup for Windows. You can execute the setup directly from the CD. To access it please change to:

- \Packages\6LoWPAN or download the latest version from http://www.dresden-elektronik.de/funktechnik/uploads/media/deCON6_Setup_Win32_V1_01.exe and follow the setup instructions.



5.2. Connecting coordinator and nodes to a network

This section describes how to prepare the network interface of your computer and how to connect the deRFgateway to the network. Moreover, the setup of the wireless network by powering the nodes (router and end device) and their configuration within the deCON6 application is explained.

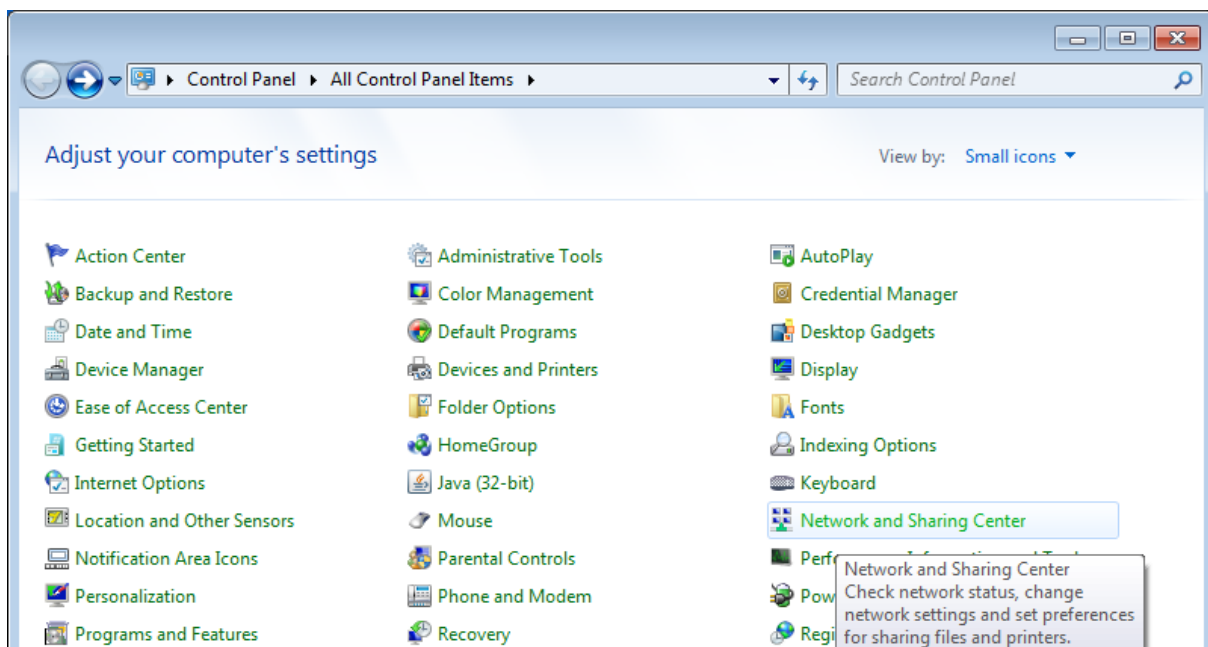
The 6LoWPAN demo application on the coordinator node communicates with a laptop or any other computer using the IPv4 protocol. This ensures that the demo application can communicate with almost any host system. An example with full IPv6 support for the border router can be found among the examples on the development kit CD.

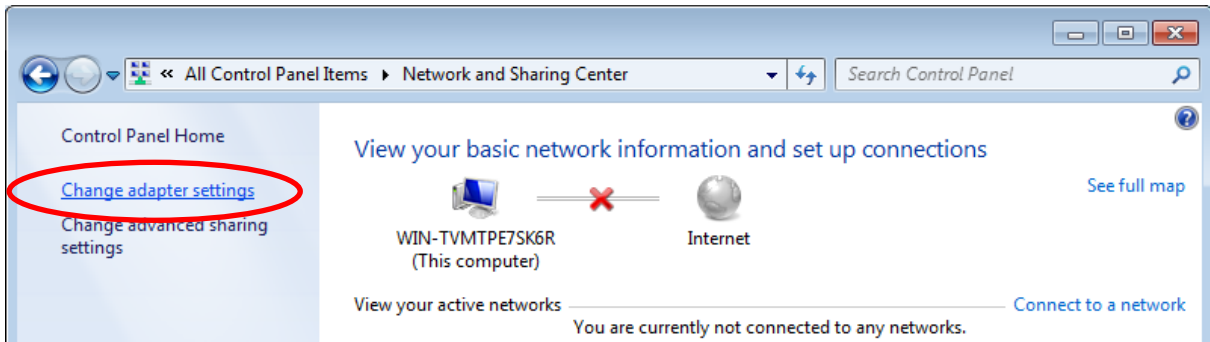
The 6LoWPAN demo application is preconfigured with a fixed IP address. To communicate from a host with the gateway both have to be within the same network segment, that is, both must use an IP address with identical prefix in respect to the network mask. Most host network configurations will not match the IP address segment of the 6LoWPAN demo application and must be changed by the user manually.

5.2.1. Prepare network interface

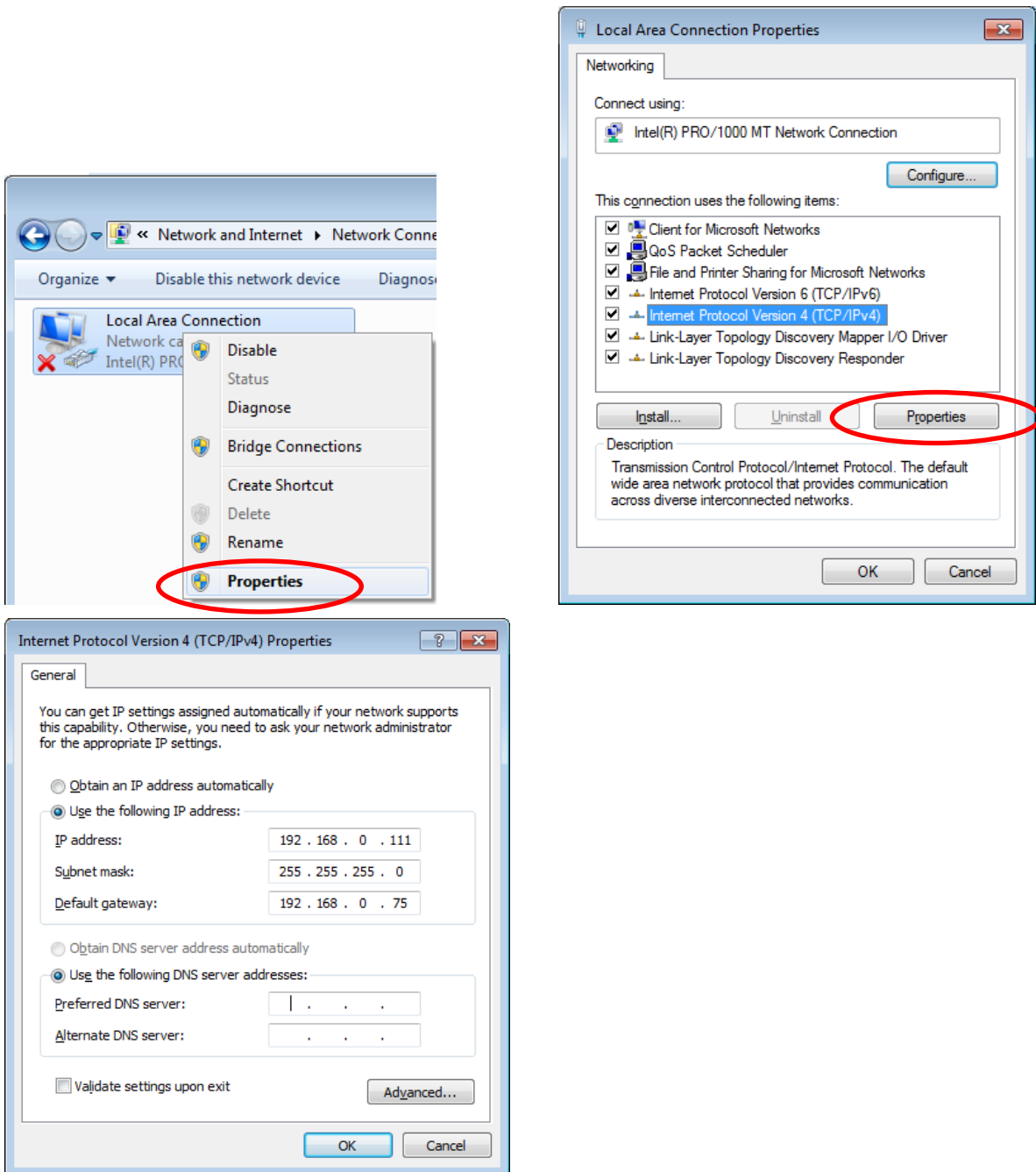
The following explanation and screenshots are valid for Windows 7 systems. Similar steps must be executed for Windows Vista, Windows XP or other operating systems.

- Configure the network interface at the PC and assign it a static IPv4 address; open: Start | Control Panel | All Control Panel Items | Network and Sharing Center
- Click 'Change adapter settings', Local Area Connection, Properties, 'Use the following IP address', see **Figures 16** and **Figures 17**.





Figures 16: Prepare network connection



Figures 17: Configuring network interface on a Windows host



- Click 'Use the following IP address' and insert the IP address 192.168.0.111, the Subnet mask '255.255.255.0'. Leave the Default gateway blank. Then click 'Obtain DNS server address automatically'. Finally click the 'OK' button, the PC applies the settings.

Note: In older version of the deRFdevelopment Kit 6LoWPAN an IP address of 192.168.192.111 might be used.

After the network interface has been configured correctly the deRFgateway can be connected to the PC and powered using the AC adapter of the development kit.

5.2.2. Connect coordinator

To proceed with the coordinator startup, run the following steps.

- Connect the coordinator to the power supply.
- Connect the coordinator to the network interface with the Ethernet cable.

Note: Please remember to either use an Ethernet switch between host and deRFgateway or a cross link cable.

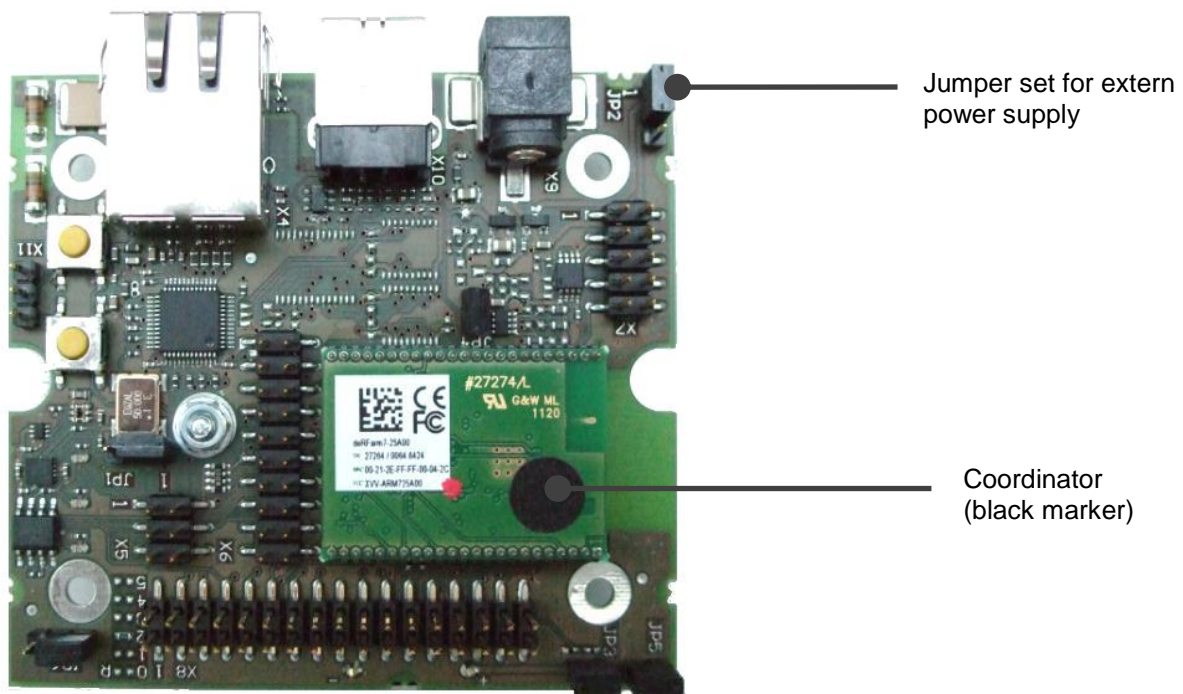


Figure 18: deRFgateway with mounted deRFarm7 as coordinator

Note: Check the correct jumper position, here 'extern power supply', see figure above or Figure 5.

With the deRFgateway powered up and connected to the PC the network connection can be tested using the simple 'ping' command.

- To execute a 'ping' open the console 'Command Prompt': Start | Execute | cmd. The coordinator's IPv4 address is 192.168.0.111

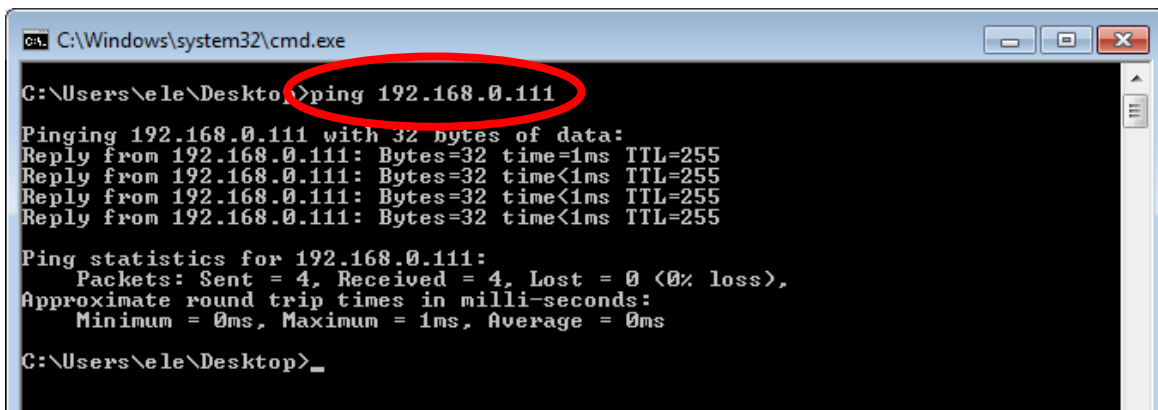


Figure 19: Check network connection

It is recommended to start the deCON6 application prior to powering the wireless nodes in order to check and monitor the setup progress. The wireless network is an ad-hoc network and nodes join the network after power on without any further configuration.

To start the deCON6 application execute the following:

- Start deCON6 via Start | Programs | deCON6.
- Select '192.168.0.111:40000' from 'Select interface' and click 'Connect'.

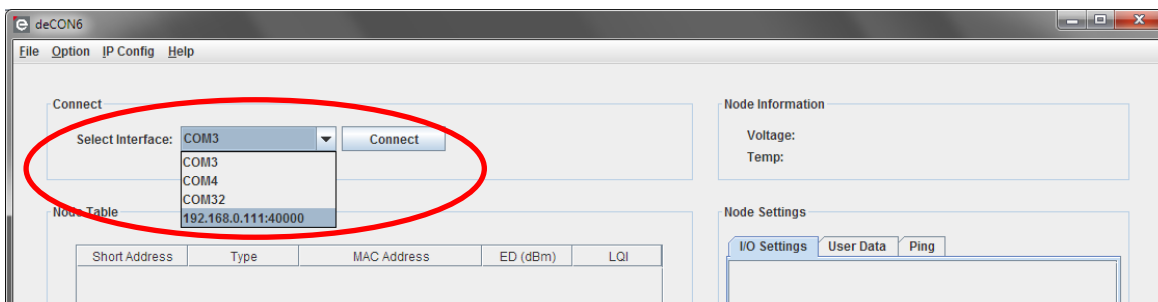


Figure 20: Start deCON6

5.2.3. Power up the nodes

To power up a node check the jumper JP2 setting and set it to battery operation as shown in the figure below or **Figure 5**.

- Insert the batteries on the deRFnode's rear side.

Note: Alternatively the nodes could be powered by connecting them to an USB host. To power a node by USB the correct driver must be installed. If you abort the installation the USB system will not provide power to the node. If you want to use USB powered nodes please refer to USB driver installation in **Section 5.3**. For USB powered nodes the jumper JP2 has to be set to external power supply.

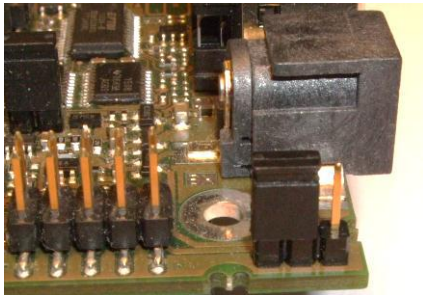


Figure 21: Jumper setting for a battery powered node

After a node is powered it joins the network. It will not appear in the deCON6 application within the 'Node Table' until you press the 'Update Table' button. Once a node is listed in the 'Node Table' you can select the node and check proper communication by running a ping, see **Figure 33**. If the communication fails a 'timeout - no ping response' message appears.

5.2.4. Using deCON6

The following sections explain more detailed how to configure the nodes and get special node information.

All node information displayed in the left part of the application relates to the currently selected node in the node table. That means you can work with only one node at a time.

Whenever you attach a new node to the network or remove nodes from the network or are in doubt if the displayed 'Node Table' is still correct press the 'Update Table' button to re-read the current node table from the coordinator.

- Use the 'Update Table' button to display the connected nodes.

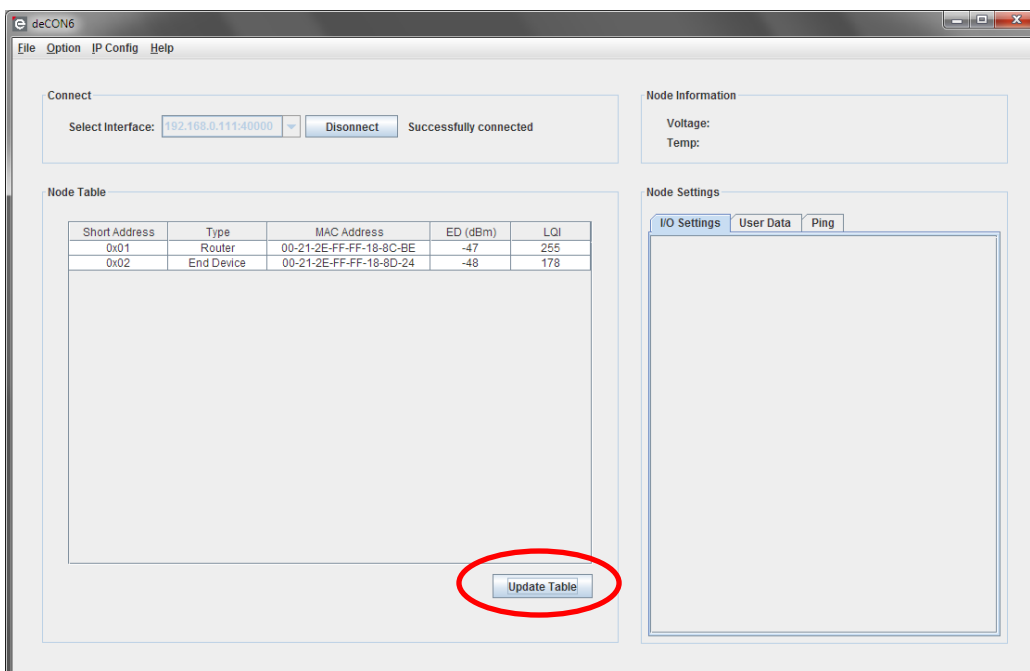


Figure 22: Display connected nodes



When selecting a node the corresponding 'Node Information' in the deCON6 window's right side is displayed.

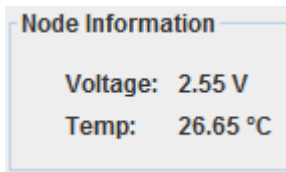


Figure 23: Node information

To interact with a node use the three tabs within 'Node Settings'. Use 'I/O Settings' for direct control over configured GPIO pins, 'User Data' for arbitrary data and 'Ping' for sending a ping command.

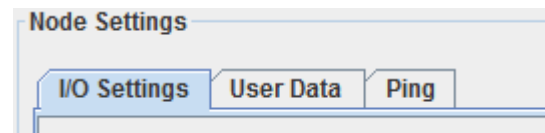


Figure 24: Node Settings

5.2.5. Configure node visualization

The deCON6 application can store an individual configuration for each node. The configuration can be changed at runtime and the names/descriptions can be adjusted by providing own XML files. To generate a configuration follow these steps:

- Select the node you want to edit in the 'Node Table', e.g. 'End Device' as shown in **Figure 25**. The selected node is highlighted and the current configuration displayed. Use 'Edit' to configure the selected node.

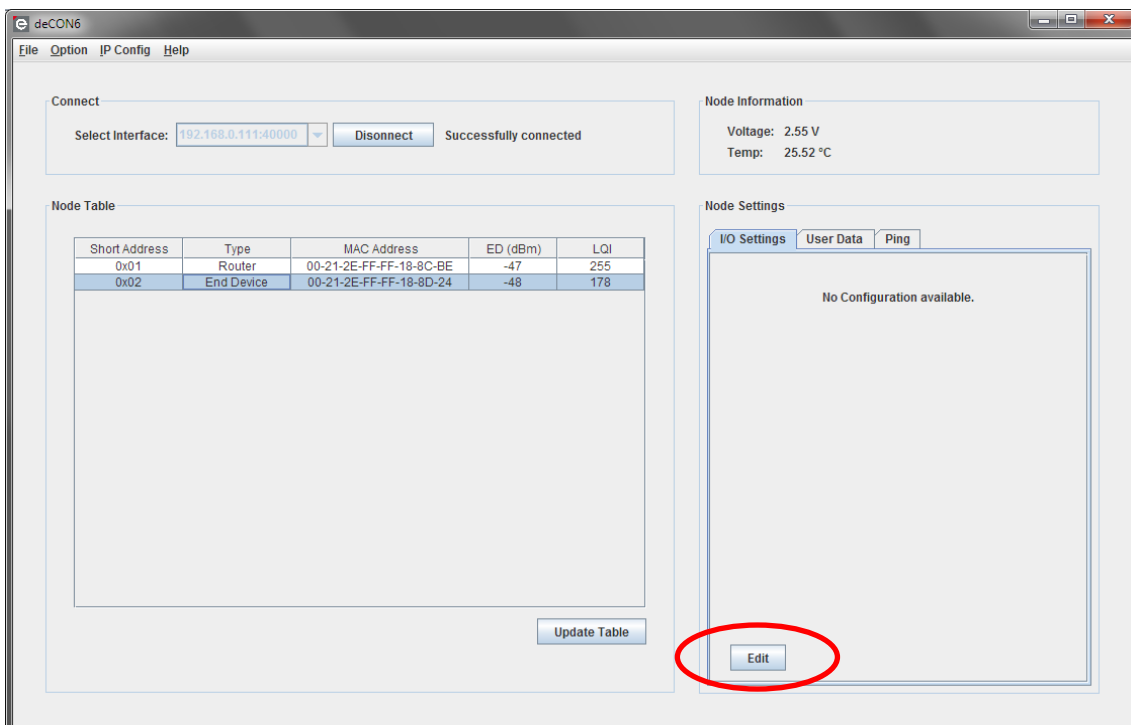


Figure 25: Edit node configuration



The 'Configuration' window appears with the 'Component:' select box in the lower area:

- Select 'deRFnode'.

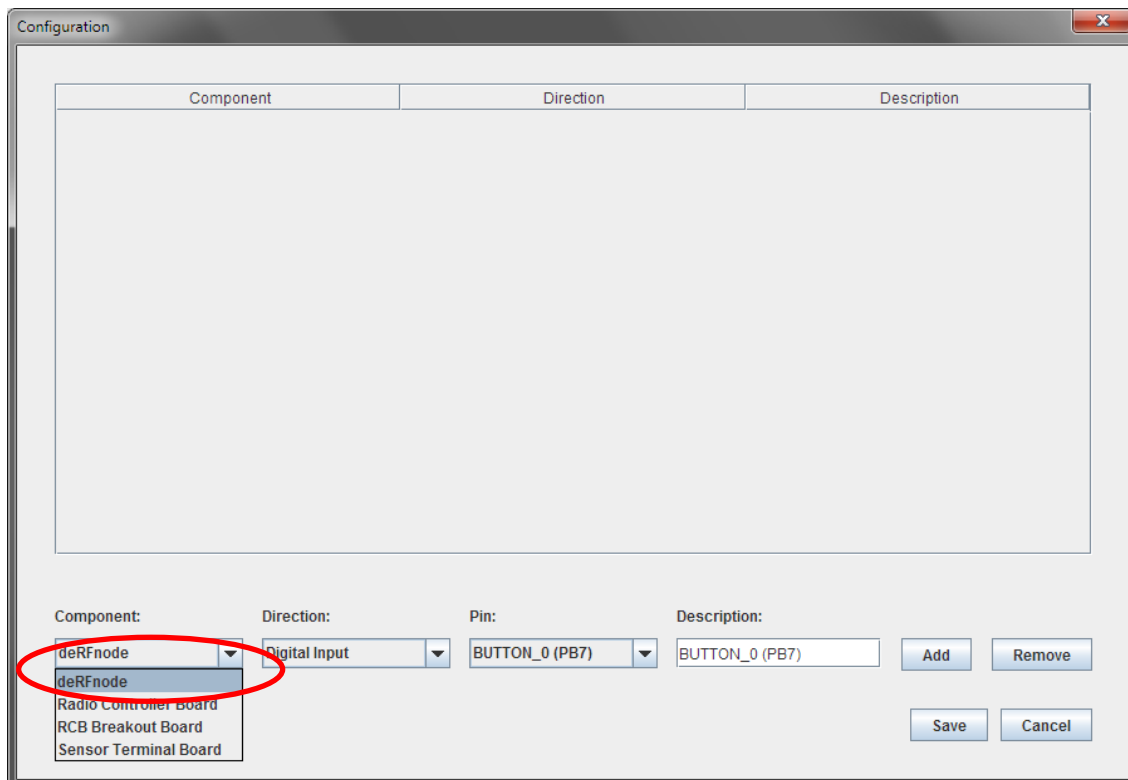


Figure 26: Select component

- With deRFnode selected continue with the 'Direction' box and select Digital Inputs. In the 'Pin' box all available input pins are listed. Select one and press the 'Add' button to add this pin to the 'I/O Settings' panel.
- Repeat this step for any digital input you want to add to the panel.
- Select Digital Outputs from the 'Direction' box. The 'Pin' box now displays all available outputs. Select the output you want to add to the panel and press the 'Add' button.
- Repeat this step for any digital output you want to add to the panel.

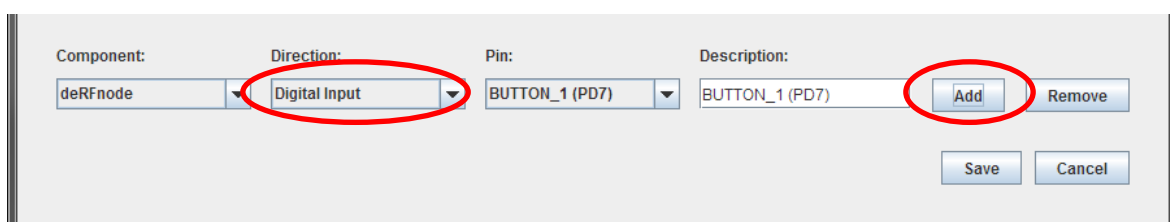


Figure 27: Configure nodes

- Save the node configuration.

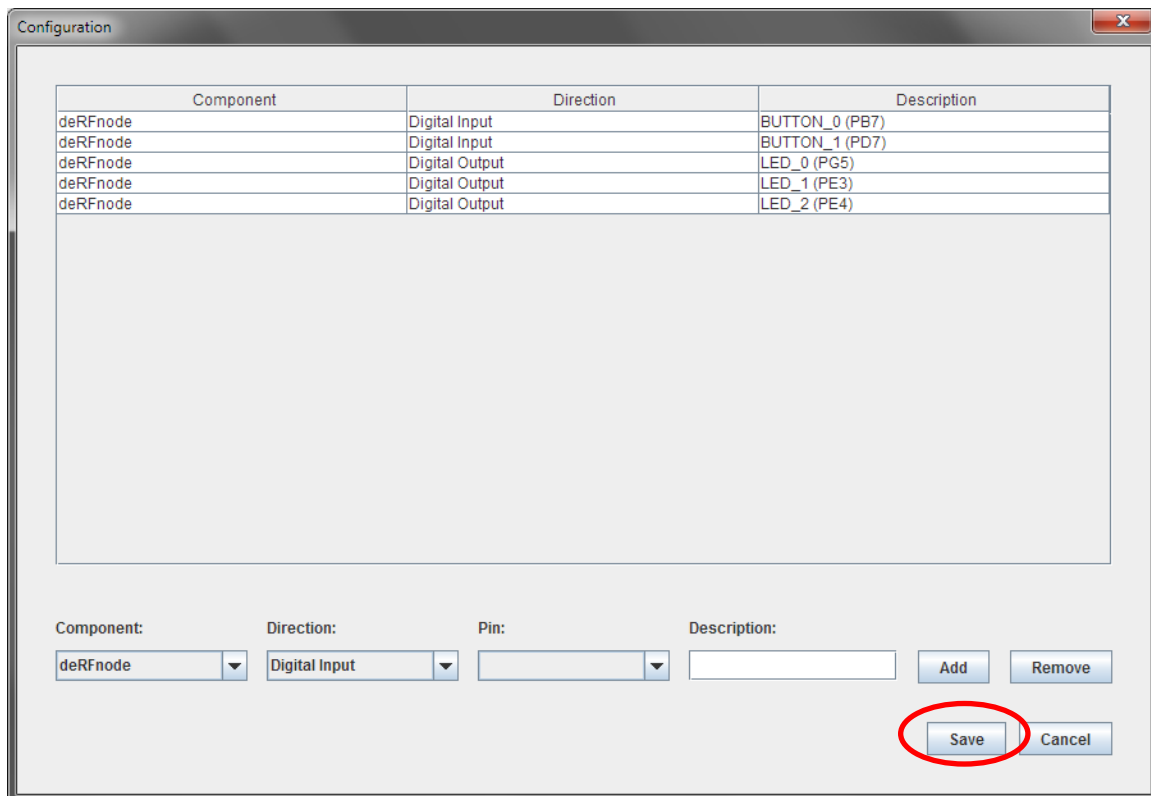


Figure 28: Save node configuration

After configuration you can use the 'I/O Settings' panel to control and visualize the pins.

5.2.6. I/O Settings tab

You can change the configuration of the 'I/O Settings' by pressing the 'Edit' button and add or remove the desired pins.

Note: Please keep in mind that the following explanations refer to the deRFnode platform. The names and functionalities of the pins can differ for other platforms especially for platforms defined by other users.

Within the 'I/O Settings' tab you can:

- Switch on or off the LEDs using the check boxes. Check the LEDs you want to turn on (checked) or off (not checked) and press the 'Send Changes' button.
- See activated inputs by a green highlighted background. Pressing a button on the selected node (e.g. button 2, see **Figure 7**) will highlight the background of the corresponding pin in the 'deCON6' window.

Note: Pressing a button on a node that is not selected in the deCON6 'Node Table' will not be visualized. The deCON6 application discards the received information.

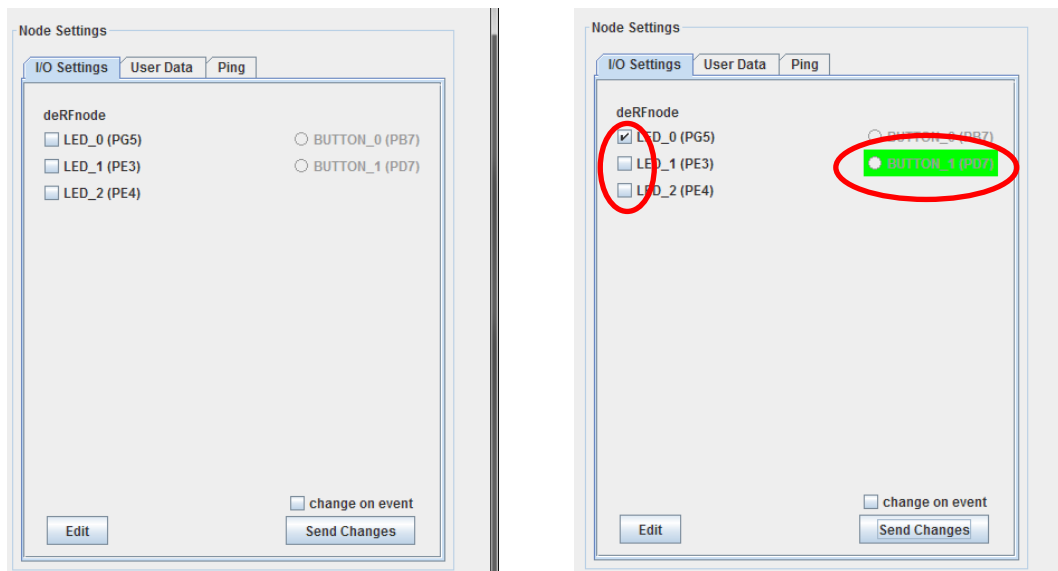


Figure 29: I/O settings tab

Note: If you activate the 'change on event' check box the 'Send Changes' button gets inactive and the deCON6 application communicates any change directly to the selected node making the LEDs switch on or off by simply clicking the check box.

5.2.7. User data tab

You can use the deCON6 application to send user defined data to any node within the 6LoWPAN network. The data is sent to the nodes via the deRFgateway and will appear on the USB interface of the node.

To see the received data from the node you can connect the node via USB to your PC and open a terminal program. Whenever you send data from the deCON6 application to the node the sent data will appear plain text in the terminal window.

Note: Older deRFdevelopment Kits may send user data over the serial interface. In that case the same serial settings can be used but the connection is made via the 6-pin trace header X5. The signals on X5 have TTL levels and need to be translated to RS232 levels using a RS232 level shifter (not an USB-serial-converter). A matching RS232 level shifter can be found on the dresden elektronik homepage (**RS232 Level Shifter**) and a functional compatible USB version too (**RS232-USB Level Shifter**). You can check out **Overview Level Shifter** to get an overview of the available versions.

Execute the following steps to send user data to a node:

- Connect the node to the PC using one of the USB cables. The operating system will ask you for a driver. Please refer to **Section 5.3** for the driver installation.
- Find out the COM port assigned to the deRFnode. Refer to **Section 5.3** for details.
- Open a terminal program. Since the deRFnode USB driver installs a virtual COM port on your PC any terminal program can be used.

Note: If you use your favorite terminal program the interface setup might differ from the description below.



Note: If your PC has no terminal program installed you can use the Tera Term (e.g. from www.h-online.com/download/teraterm-pro.html). Tera Term is a free terminal program that has been used to illustrate this example. To install the Tera Term refer to its documentation.

- Open Tera Term. Tera Term will ask you for a new connection. Select 'Serial' and the COM port for the deRFnode. If you connect several nodes to you PC make sure to select the one you want to use.

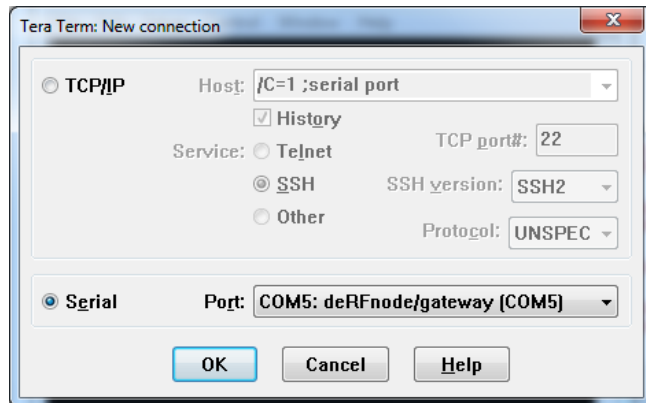
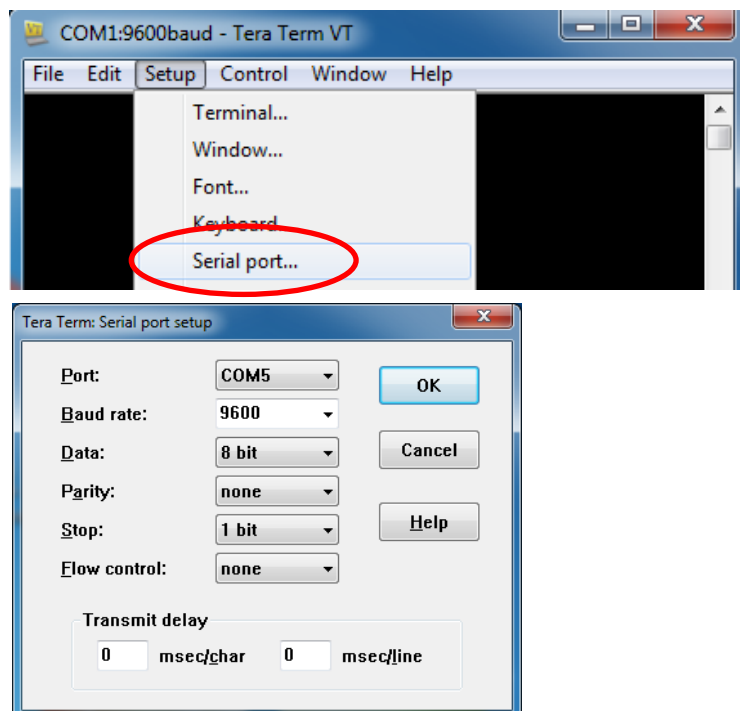


Figure 30: Configure serial port

- Press the 'OK' button to close the window and return to Tera Term.
- Change the serial port settings by selecting the Setup | Serial port... menu entry.



Figures 31: Configure serial port settings



In the above window input the port settings as noted below:

Parameter	Value
Port	According to Device Manager settings
Baud rate	9600
Data bits	8
Parity	none
Stop bits	1
Flow control	Off

Table 2: Port settings

- Within deCON6 select the node you want to send data to and enter arbitrary data into the edit window of the 'User Data' tab. Press the 'Send Data' button. For further details please refer to the 6LoWPAN software package description [12].

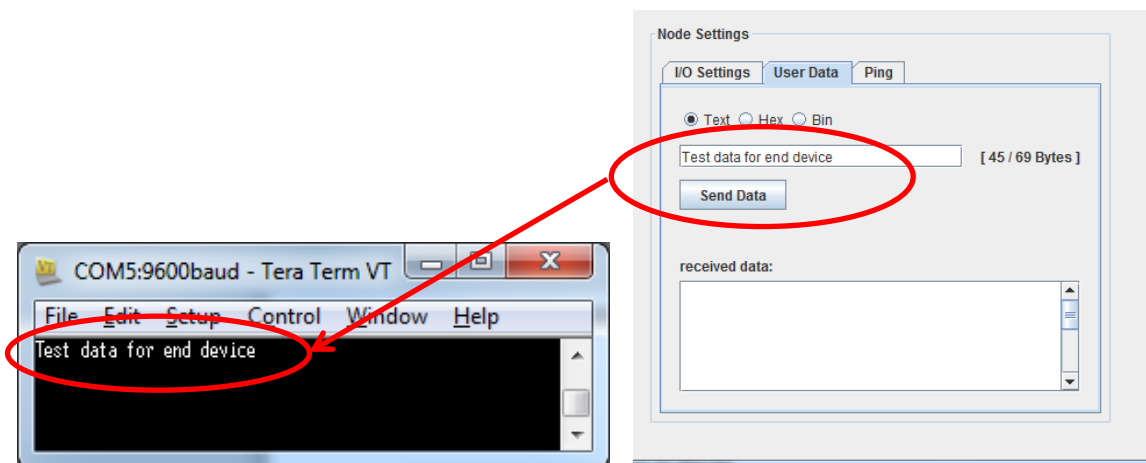


Figure 32: User Data tab

User data can be plain text, hexadecimal or binary data. Click the corresponding radio button. The number of bytes is limited by the size of a single 6LoWPAN packet as deCON6 will only send one packet per 'Send Data' click. Therefore, the maximum payload size is 69 bytes.

5.2.8. Ping tab

Use the 'Ping' tab to send a ping command to the selected node. This is a simple connection check.

To execute a ping do the following:

- Select the node in the 'Node Table'
- On the 'Ping' tab select a maximum response time.
- Press the 'Ping' button.

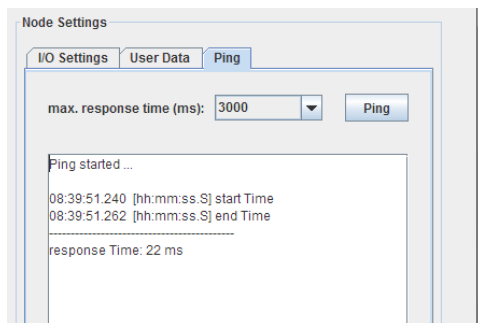


Figure 33: Ping tab

You will see the send and receive time and the calculated response time.

Note: In a wireless environment the response times can vary. In multi-hop situations the response will take longer as the packet has to pass multiple nodes.

5.2.9. Show network configuration

You can visualize the current network topology by selecting 'Show Network' from the 'Option' drop down menu of the deCON6 application.

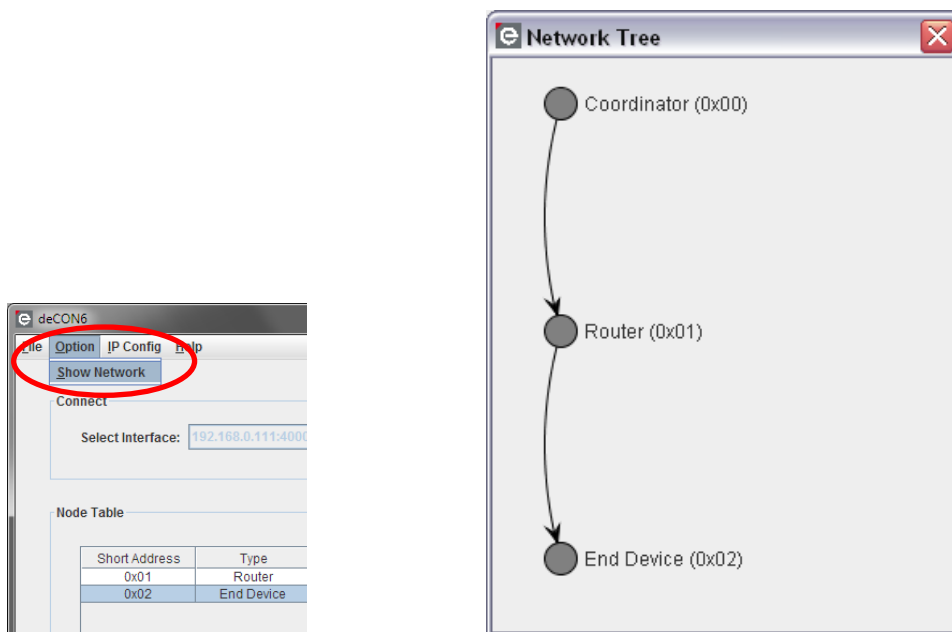


Figure 34: Current network configuration

5.2.10. Monitoring node activity

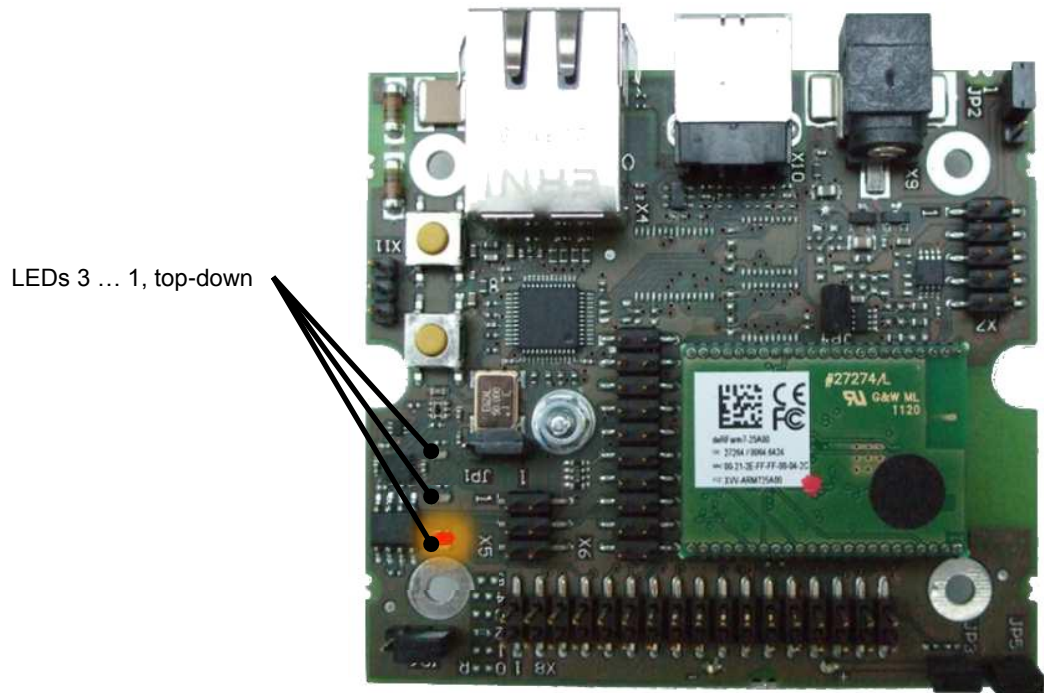


Figure 35: coordinator with illuminated LED 1

LED	Description
LED 1	Power indicator, permanently illuminated
LED 2	Toggles in case of receiving RF packets (in older versions of the deRFdevelopment kit this LED has no functionality)
LED 3	Toggles in case of receiving Ethernet packets (in older versions of the deRFdevelopment kit this LED has no functionality)

Table 3: Coordinator

LED	Description
LED 1	Controlled by the deCON6 application
LED 2	Controlled by the deCON6 application
LED 3	Controlled by the deCON6 application

Table 4: Router, end device



5.3. USB driver installation

If you connect the deRFgateway-1TNP2/deRFnode-2TNP2 for the first time to the PC or if you change the USB port, you are requested to install USB device drivers. Windows asks you if it should connect to Windows Update to find a suitable driver. Select 'No, not this time' and continue with 'Next >':

USB Driver installation under Windows XP



Figure 36: Installation wizard (1)

The next step is to manually choose the directory where the driver files are located. To do so, please select 'Install from a list or specific location (Advanced)':



Figure 37: Installation wizard (2)



All necessary files can be found in the '\Driver' directory on the delivered CD. Also check for a possible newer version on the dresden elektronik website under <http://www.dresden-elektronik.de/funktechnik/service/download/driver/?L=1#c3079>.

Attention: Do not get confused, if the installation procedure starts twice. This is necessary since both drivers for generic USB access and drivers for a virtual COM port have to be installed.

USB Driver installation under Windows 7 Editions

Connect the deRFgateway/deRFnode to the PC. Windows 7 will not install a driver for it and mark the device with an exclamation mark in the 'Device Manager'. You have to choose the driver manually. To do so, please open:

- Control Panel | System | Device Manager

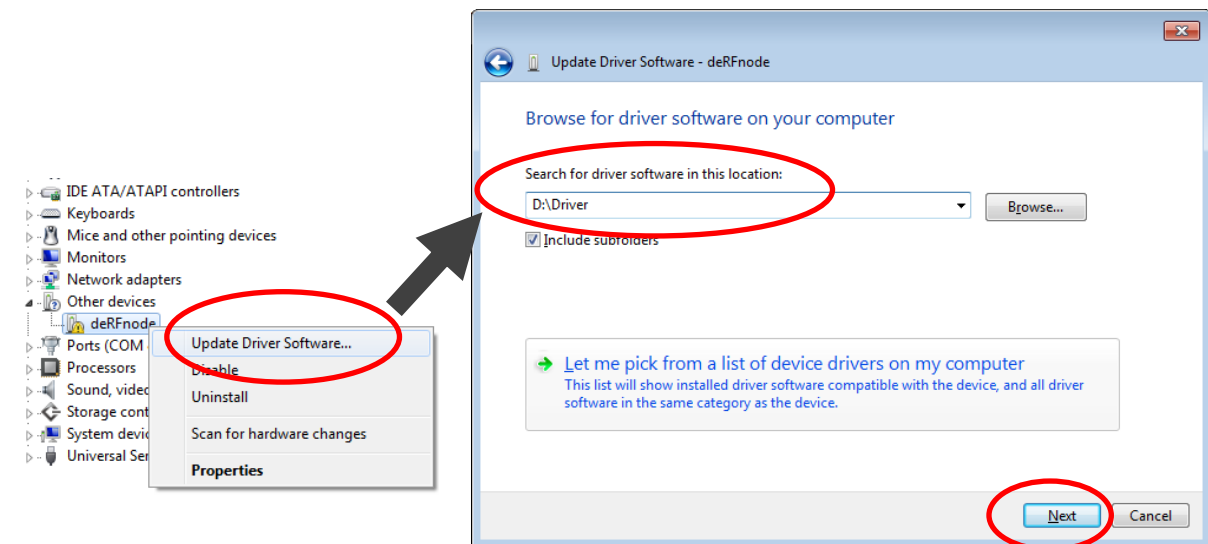


Figure 38: Installation on Windows 7 Editions

All necessary files can be found in the '\Driver' directory on the delivered CD. Also check for a possible newer version on the dresden elektronik website under <http://www.dresden-elektronik.de/funktechnik/service/download/driver/?L=1#c3079>.

Virtual COM port

A virtual COM port is automatically assigned to the newly installed USB device. To find out the COM port given to your device open:

- Control Panel | System | Device Manager

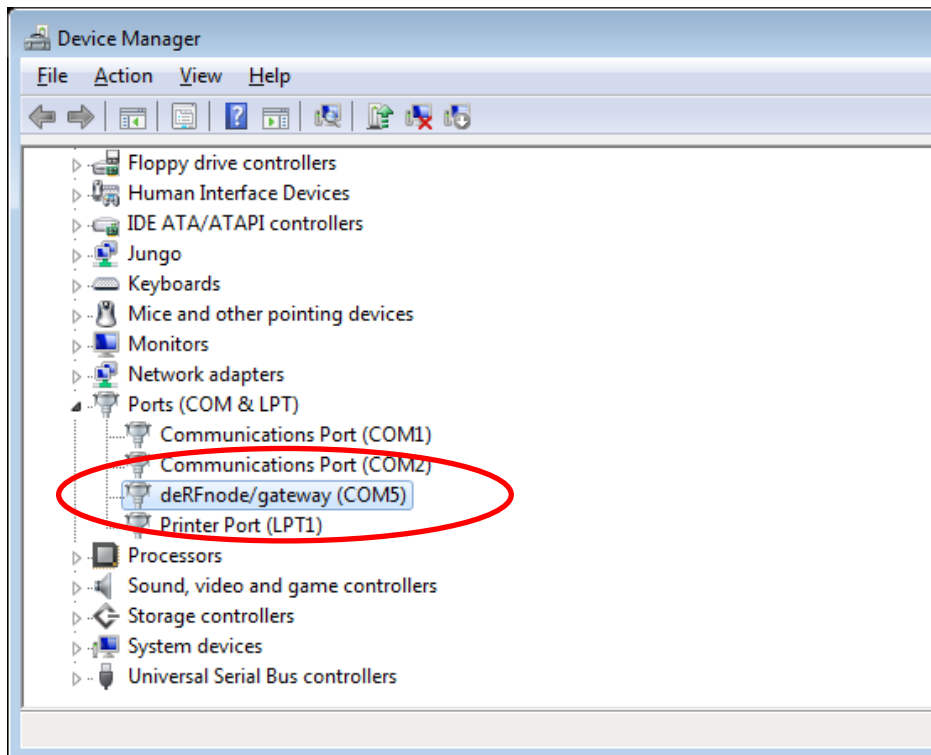


Figure 39: Assigned COM port



6. Developing custom network applications and programming

This is section an introduction to the available native examples, the MAC-based examples and the 6LoWPAN examples. This section also explains how to develop custom applications with the 6LoWPAN stack from dresden elektronik [12]. Furthermore the connection of additional sensors to the nodes is described.

The section includes instructions for compiling, debugging and flashing firmware for AVR-and ARM based microcontrollers.

6.1. Software packages

This first section introduces the application examples from the deRFnative and deRFmacAddon package developed by dresden elektronik.

Prototyping, developing, and debugging custom applications based on dresden elektronik's the 6LoWPAN stack are described in **Section 6.1.3**. The latter section also includes examples that demonstrate how to use a 6LoWPAN wireless sensor network and communicate with the nodes in IPv4 and IPv6 enabled environments. For IPv4 operations **Section 6.1.4** demonstrates how to operate a 6LoWPAN network inside an IPv4 network.

6.1.1. Native examples

The provided native examples demonstrate how to evaluate the deRFnode board features such as the available sensors, buttons and LEDs, see **Figure 7**. The native examples are running locally; radio transmission is not involved.

However, a suitable IDE, a development tool chain and a programmer are required. Example IDEs, tool chains and programmers are described in **Section 6.2.1** for AVR and **Section 6.2.2** for ARM boards.

Native application examples

Each of these applications demonstrates a special feature of a certain piece of hardware of dresden elektronik radio modules and related base boards.

In general all examples behave identically on the different deRF platforms e.g. deRFnode and deRFgateway. However, not all examples are available for each platform and module family. The examples are compiled standalone, that is, they do not require any stack installation. To keep the application code simple and easy to understand some examples use additional source code library modules e.g. from the AT91Lib for ARM or a TWI library for sensor access. In this case, the source code is included in the native examples package.

An overview of the applications available in the deRFnative package is given below:

deRFnative_Button Platform: AVR, ARM7

This example shows how to change the LED status if a button is pressed.

deRFnative_I2C_EEPROM Platform: AVR

The example code simply writes and reads a test pattern to/from the serial EEPROM available on deRFmega128 evaluation radio modules. As this EEPROM is only available on those evaluation modules it is not available for OEM modules, ARM7 based modules and USB radio sticks.



deRFnative_I2C_Sensors

Platform: AVR, ARM7

The application demonstrates the native access to the temperature, light and acceleration sensors available on the deRFnode platform. It reads out the actual values from each sensor and restarts if a button is pressed.

deRFnative_LED

Platform: AVR, ARM7, SAM3

This example changes the LED status cyclically by a timer.

deRFnative_LowPower

Platform: AVR, ARM7

This application requires a deRFnode. It demonstrates how the device may be switched between high power consumption and sleep mode. Initially the device polls the sensors in a given interval. If the button is pressed the device enters sleep mode. If the button is pressed again the device wakes up and continues reading the sensors.

deRFnative_MotionDetection_Blink

Platform: AVR, ARM7

This application uses the acceleration sensor of the deRFnode or deRFgateway. This example turns the device into something like an alarm trigger. Once activated, the LED starts to blink if the board is moved. This alert must be reset manually.

deRFnative_MotionDetection_Trace

Platform: AVR, ARM7

This application uses the acceleration sensor of the deRFnode or deRFgateway in a motion tracking way. It configures the motion detection threshold of the acceleration and uses the interrupt to read the sensor and print trace messages as long the board is moved.

deRFnative_SPI_Flash

Platform: AVR, ARM7

This application uses the on-board serial flash of a deRFnode or deRFgateway. At application start, the flash is erased. Then a random pattern is written to it and read back. Finally both read and written data are compared.

deRFnative_SRAM

Platform: AVR

This example only works with a Sensor Terminal Board and shows the access to the 32KB external SRAM of the STB. It supports RCBs only.

deRFnative_Temp

Platform: AVR

This application demonstrates the native access to the temperature sensor on the micro controller or on the platform. It supports RCBs, the Sensor Terminal Board and deRFmega128 radio modules. The application reads the value from the temperature sensor and prints the result to terminal. Button presses restart the procedure.

If you want to use the temperature sensor which is provides on deRFnode or deRFgateway platform, please see example application 'deRFnative_I2C_Sensors'.

deRFnative_USB

Platform: AVR, ARM7, SAM3

The native example demonstrates the USB access. Each byte which has been read from the USB channel is sent back with a prefix.

For deRFmega128 modules USB works in conjunction with an FTDI chip. This is provided on either the Sensor Terminal Board or the deRFnode for AVR.

For deRFarm7 boards the native USB interface is used. This requires a deRFnode for ARM or a deRFgateway. For the deRFsam3 boards the native USB interface is used too.



For further description details please refer to the documentation on the delivered CD:

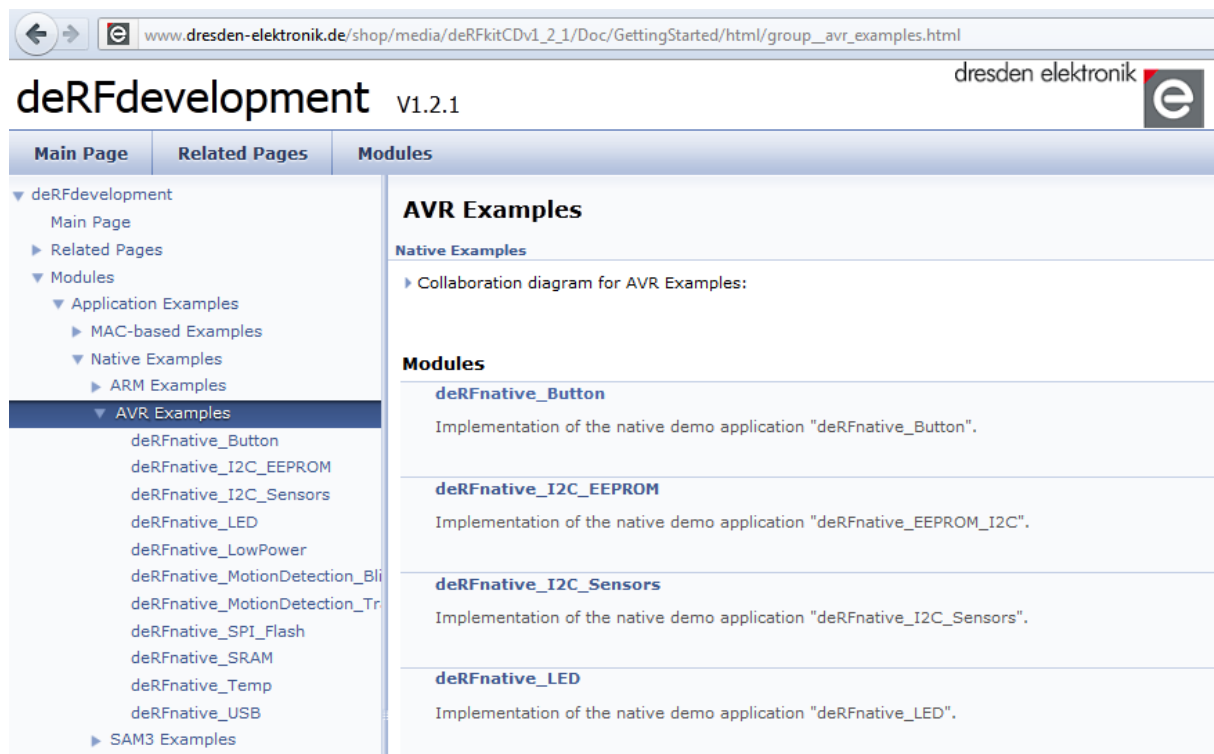


Figure 40: deRFgettingStarted

Use the 'Main Page' tab:

deRFdevelopment | Modules | Application Examples | Native Examples - select the desired platform and then click on the example.

Or use the 'Modules' tab by clicking on it; then click on the example you want to know details about.

Installation

To install the native examples, please run the self-extracting archive `deRFnativeExamples_vx_xx.exe` from the delivered CD, directory `Packages\deRFnativeExamples`, and specify any directory of your choice where you wish the files be extracted to.

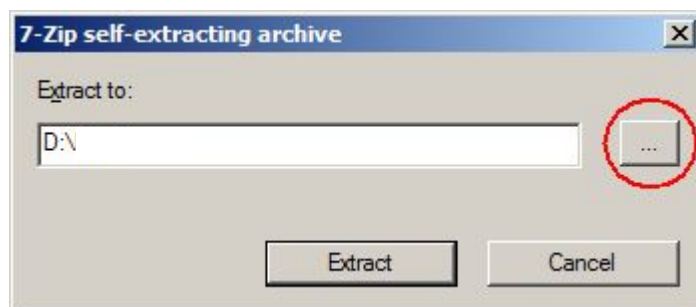


Figure 41: self-extractor running



All examples are already pre-compiled. So you may directly proceed with flashing the firmware into your target.

To select a pre-compiled binary chose your application, than your target platform and within the target platform directory open the 'GCC\bin' sub-directory. Here you will find the according binary either in .HEX-format for AVR or in .BIN format for ARM. For flashing the firmware to your device, please refer to **Section 6.2**.

6.1.2. MAC-based examples

Working with the MAC-based examples requires the availability of the IEEE 802.15.4 MAC Software Package, a suitable IDE, a tool chain and a programmer. Please refer to **Section 6.2.1** and **Section 6.2.2** for examples.

The MAC examples are based on the IEEE 802.15.4 MAC Software Package provided by Atmel Corporation. The MAC layer simplifies the programming procedure by providing all the required radio functions and a lot of example applications for the different features of Atmel radio transceivers. The deRFmacAddon extends Atmel's software package with the deRFnode board features and additional example applications.

Overview IEEE 802.15.4 MAC

The IEEE 802.15.4 standard defines the protocol and compatible interconnection for data communication devices using low data rate, low power and low complexity, short-range radio frequency transmissions in a wireless personal area network (WPAN). Atmel developed the MAC stack software for different target platforms (microcontroller and board) and transceivers.

IEEE 802.15.4 MAC Software Package features:

- Allows a highly flexible firmware configuration to adapt to the application requirements
- Supports different microcontrollers and platforms/boards
- Supports different IEEE 802.15.4 based transceivers and single chips
- Allows easy and quick platform porting
- Provides project files for two supported IDEs (IAR Embedded Workbench[®], Atmel AVR Studio[®])
- Supports star networks and peer-to-peer communication
- Supports non-beacon and beacon-enabled networks

The IEEE 802.15.4 MAC Software Package comes with the complete source code. The license conditions only limit its usage to a combination with any Atmel radio hardware.

MAC-based application examples

The IEEE 802.15.4 stack is designed with different abstraction layers e.g. Platform Abstraction Layer (PAL), Transceiver Abstraction Layer (TAL). Examples exist for each layer, that is, a TAL example only uses the TAL layer demonstrating very basic radio communication without MAC functionalities. The MAC examples use the MAC and TAL layer demonstrating different types of sensor networks. All example applications are structured according to the highest stack layer used (e.g. for TAL examples the TAL layer, for MAC examples the MAC layer).



After installing the IEEE 802.15.4 Mac Stack and the deRFmacAddon you can find the example applications under <MAC-root-directory>\Applications in the respective sub-directory of the highest stack layer.

BSP_Poll_All_Sensors Highest stack layer: BSP

This example shows how to use the Board Support Package for deRFnode and deRFgateway development boards. It accesses all sensors (temperature, ambient light and acceleration) and prints the read values to a serial link.

deRFsimple_Button Highest stack layer: PAL

This application toggles LED_0 and LED_1 every time a button is pressed by using PAL layer resources.

deRFsimple_USB Highest stack layer: PAL

This code shows how to read and write bytes from/to USB interface. To test it, simple open any terminal program on the virtual COM port that is assigned to your Sensor Terminal Board or deRFnode or deRFgateway (baud rate 115 kbit, 8 data bits, no parity, 1 stop bit). The application waits for an arbitrary character to be sent. If the application receives this character the actual temperature is read and the value sent over the USB interface to the terminal program. With ATmega128RFA1 modules the internal temperature sensor is used and with RCBs the temperature sensor from the Sensor Terminal Board is selected.

LowPower_Sensor Highest stack layer: TINY_TAL

This application demonstrates how to power down the deRFnode platform and therefore enable battery powered operation. Cyclically it wakes, determines its current operating conditions (battery voltage, environment temperature, luminosity, acceleration) and transmits these over the air. As receiver an USB dongle is used to decode and print the received messages.

Wireless_UART_2 Highest stack layer: TINY_TAL

The example implements an alternate Wireless UART, based on the TINY_TAL Example 'Wireless_UART'. It uses a timer to improve over-the-air-performance through sending data block-wise instead of transmitting frames upon every single byte received over the serial connection.

For more details please refer to the documentation on the delivered CD, see **Figure 40**.

Note: In later revisions of the IEEE 802.15.4 MAC Stack Atmel removed the TINY_TAL layer thereby removing some application as well.

Attention: The deRFmacAddon must match the version of the IEEE 802.15.4 MAC Stack from Atmel. Mixing different versions can lead to none functional dresden elektronik examples as well as none functional stack examples. The matching version is provided on the kit CD.

MAC stack setup and installation

The MAC stack that matches the deRFmacAddon can be found on the development kit CD in the directory '\Packages\MAC'. It is also strongly recommended to read the file IMPORTANT_README_FIRST.txt in the root of the Software Package!



deRFmacAddon

The deRFmacAddon is a software product developed by dresden elektronik. So only dresden elektronik is responsible for it. If you run into any trouble please write to support@dresden-elektronik.de.

The deRFmacAddon package adds hardware support for dresden elektronik platforms to the IEEE 802.15.4 MAC Stack. It also adds some example applications.

Installation

You can get the IEEE 802.15.4 MAC Stack from the Atmel homepage (http://www.atmel.com/tools/IEEE802_15_4MAC.aspx) or from the delivered CD (directory: Packages\MAC).

Software	Description
	IEEE 802.15.4 MAC (30MB, updated June 2012) version 2.8.0
	IEEE 802.15.4 MAC (17.7MB, updated October 2011) version 2.7.1
	IEEE 802.15.4 MAC (43MB, updated August 2011) version 2.6.1

The CD always contains the MAC stack version matching the deRFmacAddon. Beware of getting a newer version from the Atmel homepage with more advanced features which might be incompatible with the deRFmacAddon.

While running the installer **please choose a top-level-directory** as installation directory e.g. C:\Mac as otherwise the path length may cause trouble. This installation location is referred to as <MAC-root-directory> from now on.

Note: Always execute the MAC stack setup first.

Note: Choose a directory with a short name e.g. C:\Mac.

After the MAC Stack installation you can proceed with the installation of the deRFmacAddon. The addon is also provided on the delivered CD under 'Packages\MAC'.

To install the deRFmacAddon run the installer deRFmacAddon_for_Atmet_MAC_vx_x_x_Win32_Vx_xx.exe and choose the directory where you have installed the MAC Stack before.



Figure 42: Self-extractor

Confirm overwriting and/or replacing files.



Note: Always install the deRFmacAddon into the same directory as the MAC Stack.

Note: The 'Confirm File Replace' dialog is a good indicator that you have chosen the right directory.

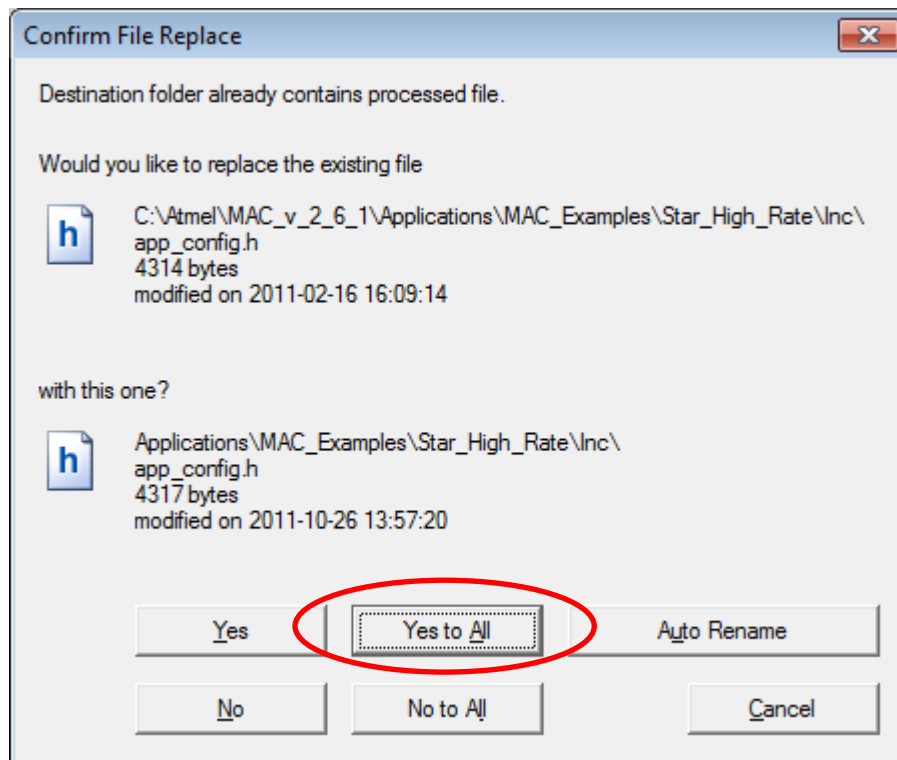


Figure 43: Confirm file replace

All examples come already pre-compiled. So you may directly proceed with flashing them into your target.

You can access the pre-compiled binary of your chosen example application by first opening the <MAC-root-directory>\Applications directory and opening within it the respective highest stack layer. Now go into the directory of the application and open your target platform and go deeper to the respective 'GCC\bin' sub-directory. Here you will find the pre-compiled binary either in .HEX-format for AVR or in .BIN format for ARM. For flashing the firmware to your device, please refer to **Section 6.2**.

If you want to modify an example act analogously as explained for the Eclipse Integration of the MAC stack as explained in **Section 6.2.2.5**.

6.1.3. 6LoWPAN-based examples

You can find the 6LoWPAN-based application examples designed by dresden elektronik after installing the deRF6LoWPANAddon from Packages\6LoWPAN in the <MAC-root-directory>\Applications\SLOW sub folder.

There are three example applications available:

- Control4SLOW is the demo application for the development kit and demonstrates a GUI based control of 6LoWPAN sensors using a special dresden elektronik protocol.
- TCPPEcho integrates the 6LoWPAN sensors into an IPv4 environment by adding Network Address Translation (NAT) functionalities to the gateway.



- UDPEcho allows direct communication with all wireless nodes in an IPv6 enabled environment. All communication uses UDP protocol as the name already suggests.

Control4SLOW

Highest stack layer: ROM

The Control4SLOW example application demonstrates how to use a 6LoWPAN wireless sensor network to control and monitor an arbitrary environment. The device types router and end device are available as controllable nodes and the coordinator as the border router to the wired Ethernet. All router and end devices connect to the coordinator via radio link and the coordinator communicates to the Java application deCON6. The communication between the coordinator and the Java application is based on the IPv4 protocol using a TCP connection. The IPv4 settings are adjustable for deCON6 and for the coordinator.

The wireless communication uses 6LoWPAN and UDP connections.

TCPPEcho

Highest stack layer: ROM

Many environments are based on IPv4 infrastructure and do not have real IPv6 connectivity. As 6LoWPAN wireless sensor networks are based on IPv6 addresses no direct communication to the sensor nodes is possible. The aim of the TCPPEcho example is to enable the communication with individual wireless nodes within those environments by providing a similar mechanism as Network Address Translation.

The border router (coordinator) supports both IPv4 and IPv6 connections. It uses the IPv6 protocol on the wireless side and the IPv4 protocol on the wired side. The TCPPEcho application opens a new TCP port on the wired side any time a new 6LoWPAN node joins the wireless network. Due to this mechanism each 6LoWPAN node is accessible via a dedicated TCP port. The coordinator does not perform a network address translation but a technique which could be called PAT (port address translation). With this PAT all wireless IPv6 nodes are accessible by single IPv4 address from the IPv4 network.

Please refer to **Section 6.1.4** for further details about the PAT mechanism.

UDPEcho

Highest stack layer: ROM

This application allows the direct communication with any 6LoWPAN node within an IPv6 enabled environment. As the name suggests an UDP connection to the target node is established and all transmitted data passes through the border router that acts simply as a gateway between the wired and the wireless world.

All examples are already precompiled so you might proceed by directly flashing the required example onto the target nodes.

Installation

The 6LoWPAN addon from dresden elektronik is based on the Atmel IEEE 802.15.4 MAC Stack (http://www.atmel.com/tools/IEEE802_15_4MAC.aspx) and the deRFmacAddon. That means that you need to install three packages to achieve a functional 6LoWPAN stack. Each addon extends the functionality of the IEEE 802.15.4 MAC Stack and the installation must follow this order:

Atmel IEEE 802.15.4 MAC Stack, deRFmacAddon, deRF6LoWPANAddon.



The addons are self-extracting archives and have to be extracted to the location where the IEEE 802.15.4 MAC Stack was installed. Please refer to **Section 6.1.2** for the installation of the IEEE 802.15.4 MAC Stack and the deRFmacAddon before proceeding with the deRF6LoWPANAddon.

Note: Always install the deRFmacAddon prior to the deRF6LoWPANAddon. Otherwise the deRFmacAddon will destroy required files of the deRF6LoWPANAddon.

After the IEEE 802.15.4 MAC Stack and the deRFmacAddon are installed you can proceed with the installation of the deRF6LoWPANAddon. The addon is also provided on the delivered CD under Packages\6LoWPAN.

To install the deRF6LoWPANAddon, open the self-extracting installer deRF6LoWPANaddon_For_Atme1_MAC_v2_6_1_Win32_Vx_x.exe and choose the directory where you have installed the IEEE 802.15.4 MAC Stack. Confirm overwriting files, if requested.

Select the directory you like to extract to:

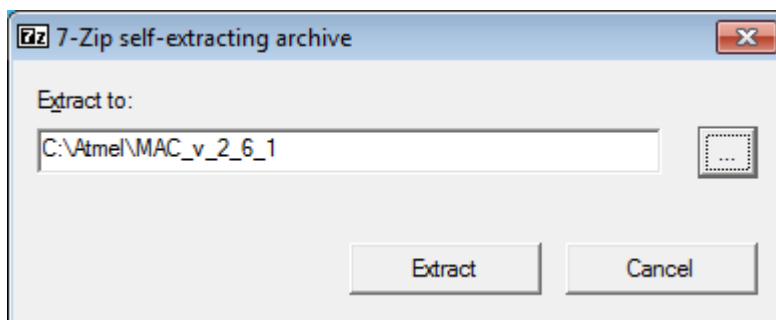


Figure 44: Self-extractor

Confirm to replace existing files.

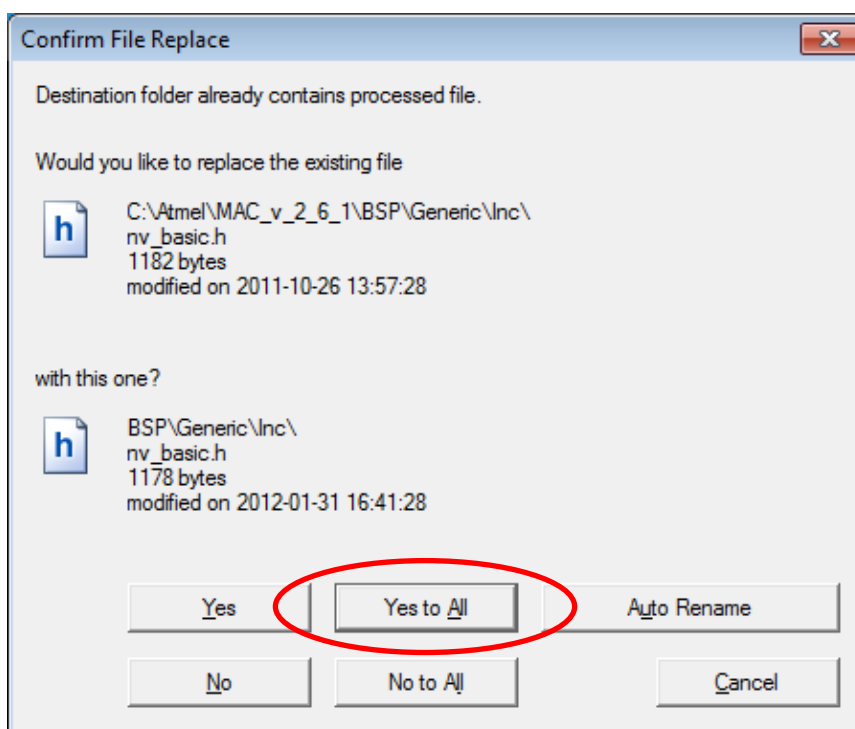


Figure 45: Confirm file replace



For installation details please refer to [12], 6LoWPAN Developer and User Guide/BHB, Section 4, Addons from dresden elektronik.

6.1.4. IPv4 to IPv6 gateway example

Using the TCPPortEcho example from the 6LoWPAN deRFdevelopmentKit a 6LoWPAN radio network can be operated inside of an IPv4 network giving you access to each node from the IPv4.

Figure 46 shows a 6LoWPAN radio network and the border router connected to some IPv4 network. The border router generates and manages the IP sub-net that contains all wireless nodes and acts as gateway to the global IP infrastructure. The border router forwards data from the outside to the wireless nodes and vice versa.

With this PAT mechanism the user does not need to know the individual IPv6 address of a wireless radio node. Only the IPv4 address of the border router and the port for the desired node are required.

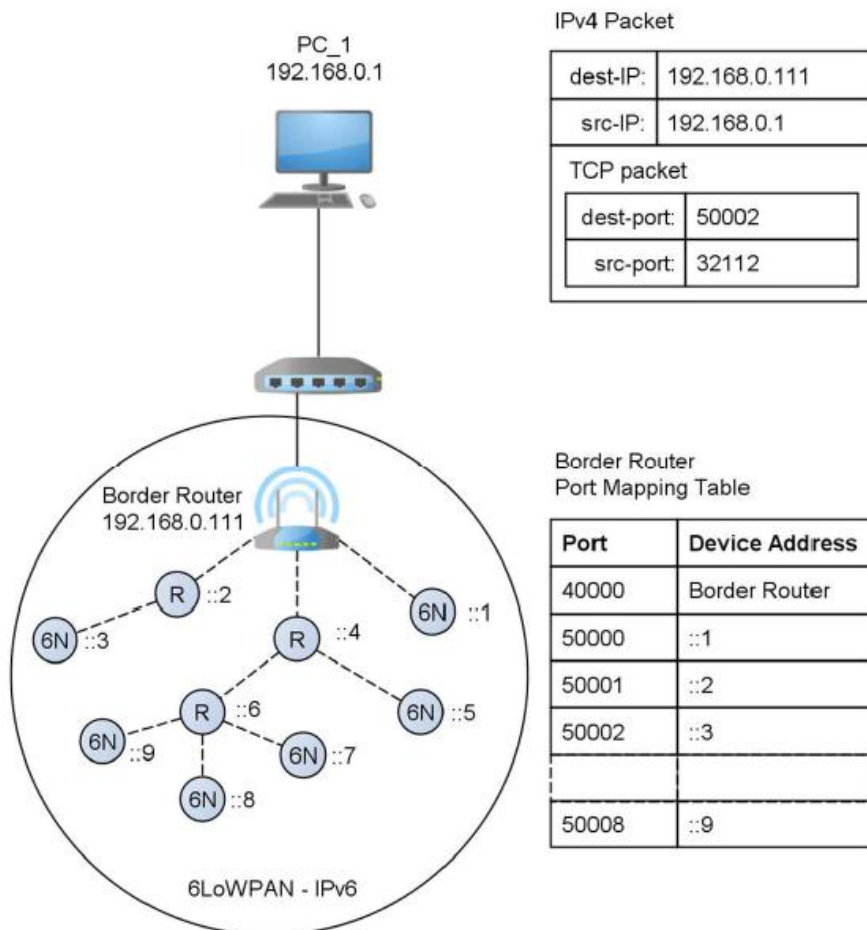


Figure 46: 6LoWPAN radio network connected to an IPv4 network

To communicate with a wireless node open a TCP connection to the border router. On success you can send and receive data as with any other Ethernet connection.

Note: Please keep in mind that the communication over the wireless link is only UDP based, so data losses are possible as you only have a TCP connection to the border router not the wireless node itself.



The port numbers are assigned in the order nodes join the network, that is, the first node that joins gets port 50000, the second one 50001 and so on. If a node leaves the network the assigned port is still available at the IPv4 side but data exchanges will fail. If the node joins again it gets the same port assigned and data exchange can continue as before. However, power-cycling the border router will result in a completely new assignment of the ports and it is very likely that the nodes will get new ports assigned.

Note: If you open a TCP port on the wired interface and this fails it is likely that no node has joined and this port number is not yet assigned (e.g three nodes joined the network and you try to open port 50003).

A simple tool to establish an IPv4 connection to the different nodes and send/receive data is provided on the development kit CD. The application is called deIP-Client and requires a Java Runtime Environment.

To install a Java Runtime Environment or ensure that the correct version is already installed on the PC please refer to **Section 5.1**.

To install the deIP-Client please execute the following steps:

- Run the deIP-Client setup from the delivered CD:
\\Packages\6LoWPAN\deRF_IPClient_Setup.exe
- Follow the given instructions.
- Start the deIP-Client by clicking the Windows Start icon | Programs | deIP-Client.

Input as 'Remote IP' the IPv4 address of the border router and as 'Remote Port' the port of the node you want to communicate to e.g. 50000 for the first node that joined the network.

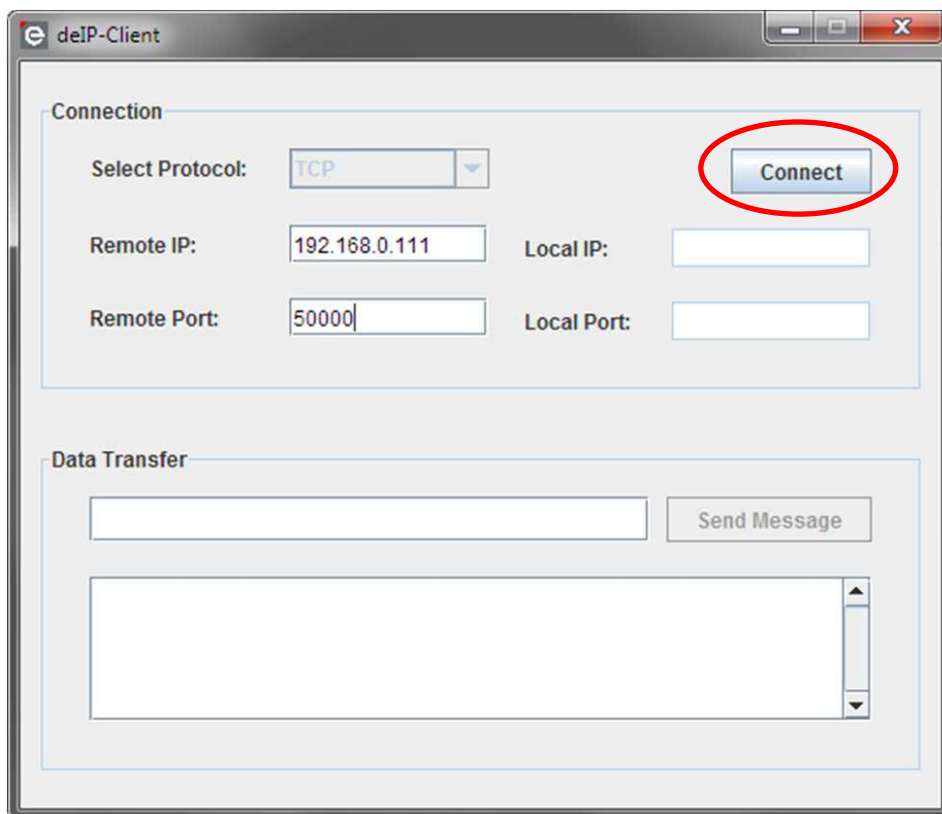


Figure 47: deIP-Client



Press the 'Connect' button to establish the connection. The 'Local IP' and 'Local Port' fields will show the IPv4 address and port the host uses for this connection. An example for a successful connection is shown in **Figure 49**.

If the client cannot connect to the requested port an error message will appear after a couple of seconds.

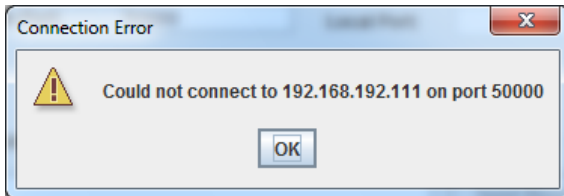


Figure 48: Connection error

After the connection has been established any message can be send to the node using the 'Data Transfer' (1) input field. Input any data into the input field and press the 'Send Message' button. The node receives the data and sends the same data back to the originating IP address. The response will appear in the box (2) below the input field.

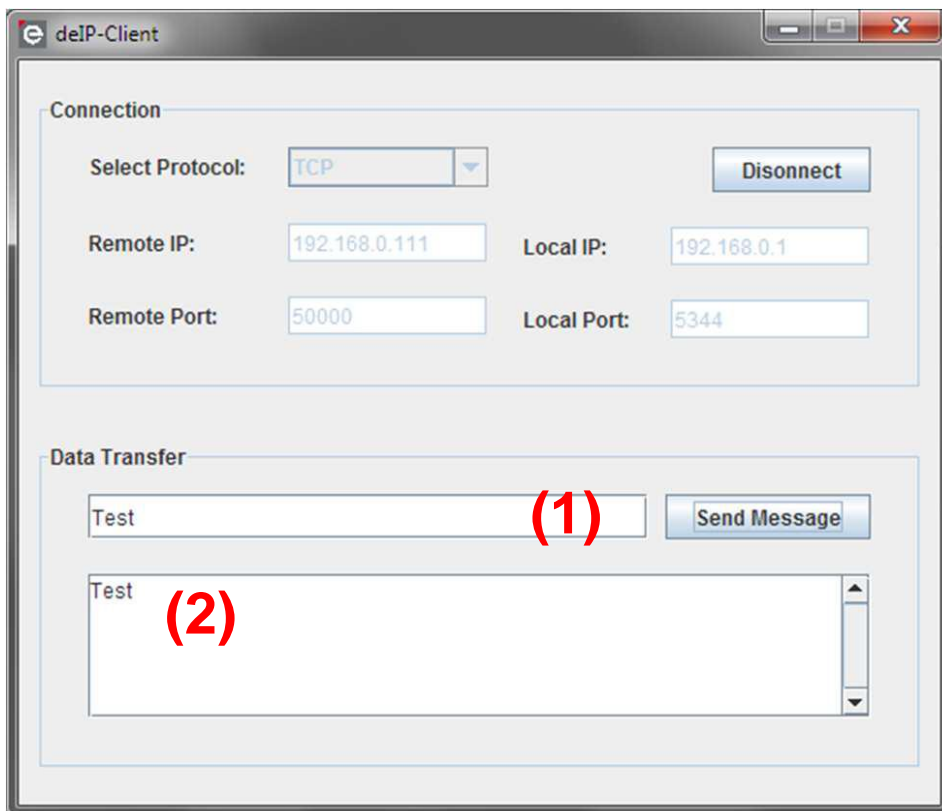


Figure 49: Using deIP-Client to send and receive data

You can run multiple instances of the deIP-Client to communicate with more than one node at the same time.

For more information please refer to the latest documentation available at dresden elektronik software section (<http://www.dresden-elektronik.de/funktechnik/products/software>).



6.2. Development tools

A development tool chain consists of:

- An integrated development environment (for example Atmel Studio® or Eclipse), where sample applications may be modified, compiled, and debugged,
- a corresponding compiler tool chain (for example AVRGCC or YAGARTO), which provides everything necessary to compile application source code into binary images, and
- a programming device (for example, JTAG) which may be used to program and debug the application on a target platform

Atmel Studio can be used to develop and debug applications for AVR-based platforms. In Atmel Studio each application has a corresponding project file identified by the *.atsln extension. A project file contains the necessary information about build configuration.

The Eclipse IDE can be used to develop and debug applications for almost any processor type. Due to Eclipse not being able to debug AVR processors using Atmel programmers dresden elektronik recommends the Eclipse IDE mainly for ARM based developments that is for the deRFarm7, deRFsam3 modules and deRFusb sticks. However, AVR developments can be carried out in Eclipse too but always need to be programmed using either a command line tool like avrdude or the already mentioned Atmel Studio. For more details and how to work with the Eclipse IDE please refer to **Section 6.2.2.5**.



6.2.1. AVR Development

AVR development is carried out in the Atmel Studio. The IDE supports AVR programmers and debuggers from Atmel and represents a full development environment for AVR. An Atmel Studio installation comes with the latest supported AVR and ARM tool chains for Atmel processors so no separate tool chain installation is required. It is even possible to use the installed tool chains with other IDEs like Eclipse. For all features of Atmel Studio and supported programmers please refer to Atmel's homepage.

To get a deeper insight into the programming options for AVR and ARM devices, the required tools and processes please refer to [9]. Please note that the Atmel Studio can be used for AVR and ARM based developments.

6.2.1.1. Setup IDE

The Atmel Studio is a free IDE from Atmel Corporation. To install it, follow these steps:

1. Run the Atmel Studio Setup file from the delivered CD (\3rdParty\Atmel) or download the latest version of Atmel Studio from the Atmel homepage:

<http://www.atmel.com/tools/studioarchive.aspx?tab=overview>

When downloading the Atmel Studio, the user can either choose between a full packet combined with the .Net Studio or download the Atmel Studio without the .Net Framework (if it is already installed on the PC).

Software	Description
	Atmel Studio 6.1 update 1.1 (build 2674) Installer (595MB, updated June 2013) This installer contains Atmel Studio 6.1 update 1.1 with Atmel Software Framework 3.9.1 and Atmel Toolchain. This update adds support for the SAM D20 and SAM4N families of ARM-based microcontrollers.
	Atmel Studio 6.1 update 1.1 (build 2674) Installer - Full (806MB, updated June 2013) This installer contains Atmel Studio 6.1 update 1.1 with Atmel Software Framework 3.9.1 and Atmel Toolchain. This update adds support for the SAM D20 and SAM4N families of ARM-based microcontrollers. This installer also contains MS Visual Studio Shell and .NET 4.0. Select this installer if you need to install Atmel Studio on a computer not connected to the internet.

Optionally the Eclipse IDE may be used in parallel or instead of the Atmel Studio.

1. Download the Eclipse IDE for C/C++ Developers at www.eclipse.org/downloads. It comes as a ZIP archive.
2. Extract it to a directory of your choice. An installation is not required.



6.2.1.2. Tool chain setup for AVR-based MCUs

This section describes the tool chain setup for AVR-based MCUs required for compiling and debugging. The different dresden elektronik products use the following MCUs:

- ATmega1281 (RCBs)
- ATmega2561 (RCBs)
- ATmega128RFA1 (RCB 6.3, deRFmega128 modules)
- ATmega256RFR2 (deRFmega256 modules)

Note: This installation is only required if you use the older AVR Studio or Eclipse IDE as the new Atmel Studio already installs the tool chain.

To install the WinAVR GNU compiler (version 2010/01/10) please start the installer from the delivered CD (3rdPartyWinAVR).

This provides the GNU compiler collection tool chain (GCC version 4.3.3) as well as any other required utilities like make or rm (remove, by UNIX).

6.2.1.3. Suggested programmers for AVR-based microcontrollers

dresden elektronik suggests to use one of the following programmer devices:



Figure 50: AVR JTAGICE mkII



Figure 51: AVR Dragon

Notes:

1. Even if there are many other programmers on the market, they are not listed above since dresden elektronik does not support these types.
2. JTAG Programmers for ARM devices cannot be used to program AVR devices and vice versa.
3. A programmer is not contained in any development kit. It has to be purchased separately.

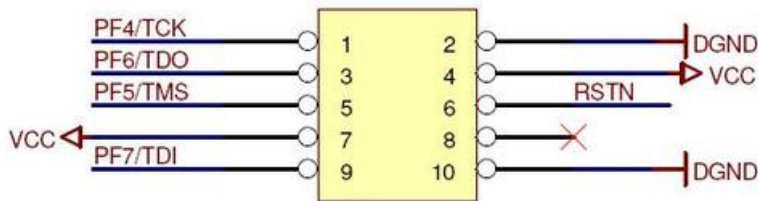


Figure 52: Pinout for 10-pin AVR JTAG header

6.2.1.4. Connecting the target

To program a device, the programmer must be connected to the respective JTAG header and the target needs to be powered.

On the different dresden elektronik products several JTAG programming headers are available:

- 10-pin AVR-JTAG header
- 10-pin AVR-JTAG mini header (pin-compatible)

Note: Not all boards (e.g. RCBs, deRFmodules) have a JTAG header. In those cases an additional board may be required for providing this header (e.g. Sensor Terminal Board, deRFnode, deRFbreakout Board).



Figure 53: AVR JTAGICE mkII connected to deRFmega128 plugged onto a deRFnode

Note: The blue cable is not required for programming. It connects to a dresden elektronik USB level shifter stick to a PC for serial debug output. Although not required, a trace (debug output) connection is largely helpful for any embedded development.



6.2.1.5. Open a project in Atmel Studio

To open a project with Atmel Studio the according *.atsln file has to be opened. This can be done by simply double-clicking on it. The following pictures show the Atmel Studio views for the different steps using the UDPEcho application as an example.

The project files are shown in the project explorer and a build process can be started from the 'Build' menu of the Atmel Studio - choose Build | Rebuild All.

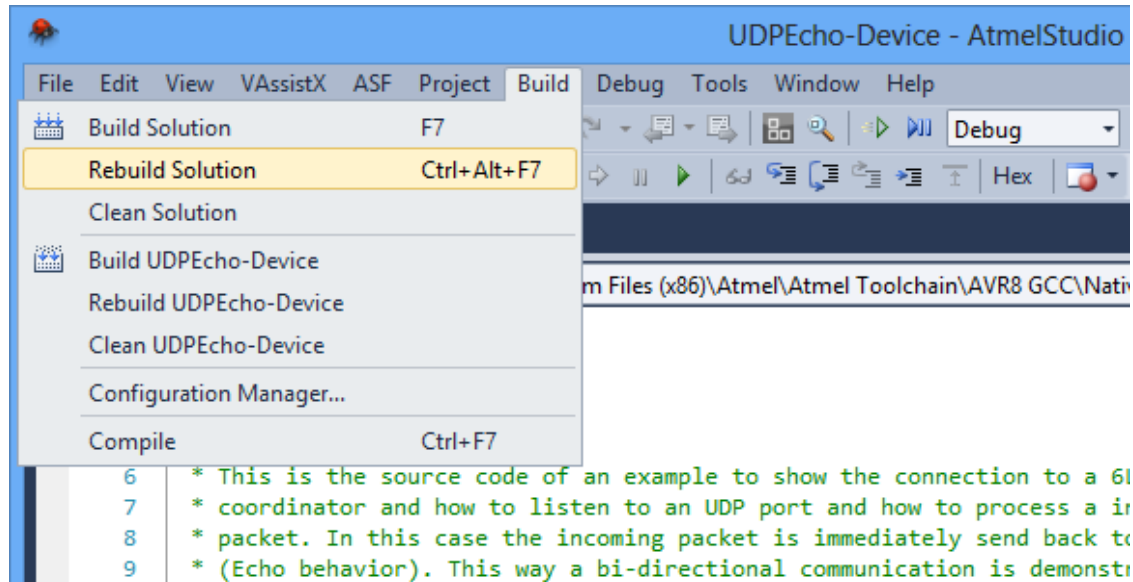


Figure 54: Rebuilding the project

6.2.1.6. Downloading the application

In the next step the device and Atmel Studio have to be prepared for downloading and debugging. The Atmel Studio has to be told the kind of programmer and AVR device:

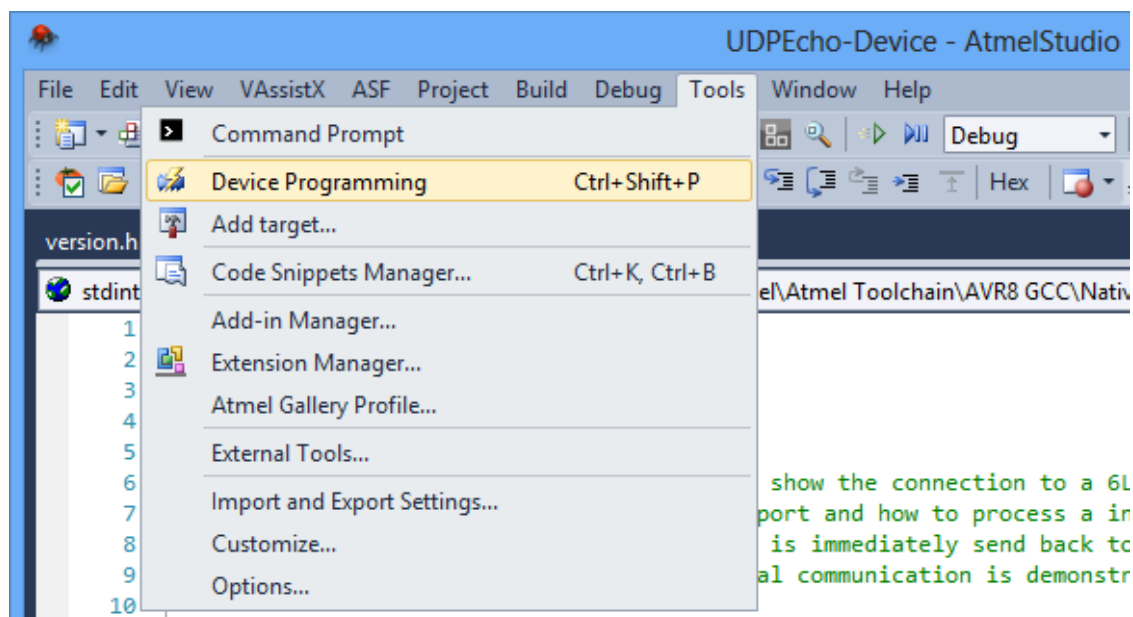


Figure 55: Preparing the connection to the AVR programmer

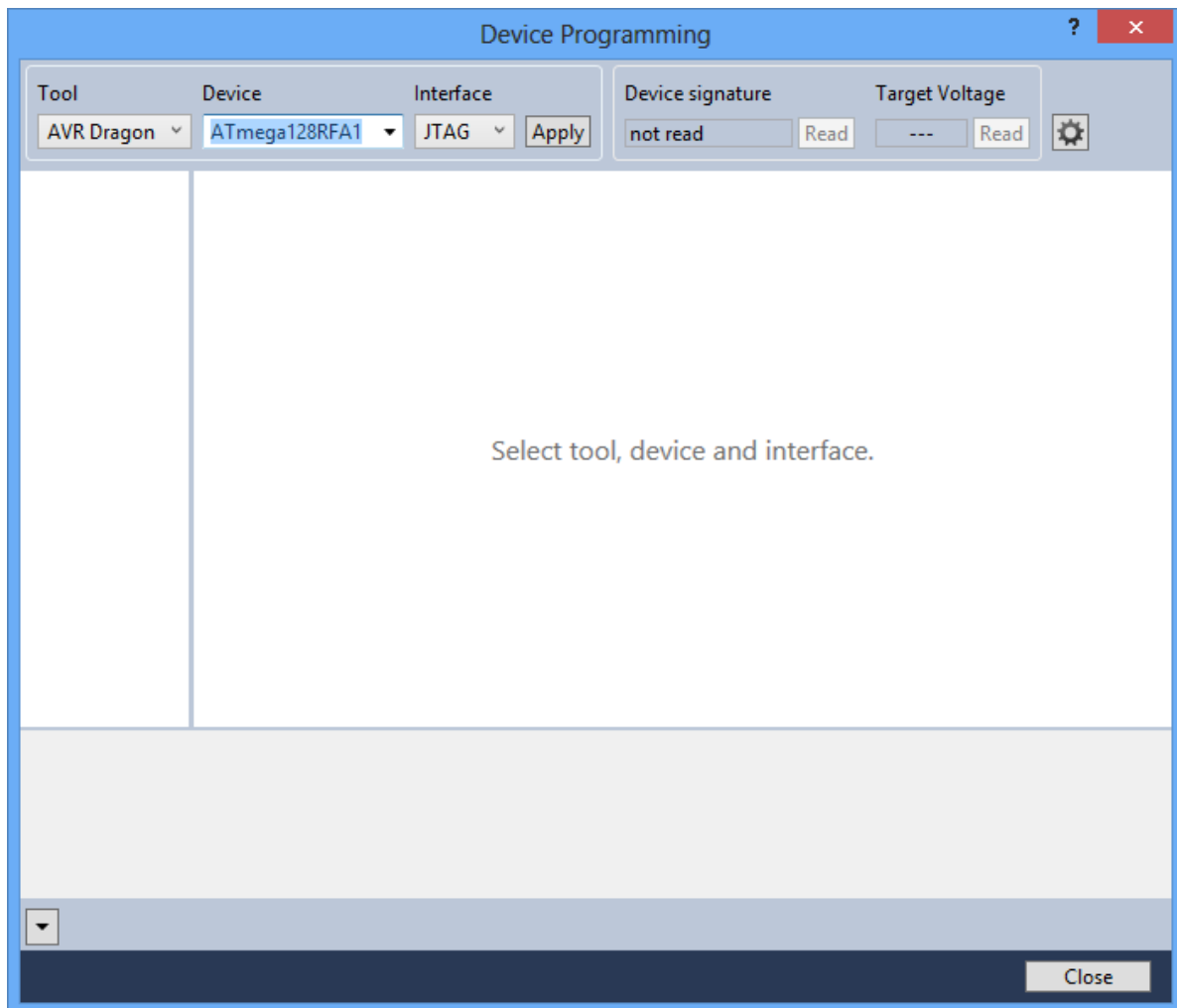


Figure 56: Choosing programmer, device and connection (JTAG)

For programming a suitable programmer must be chosen. The AVR Dragon or Atmel JTAG-ICE mkII are recommended. By connecting a programmer to the PC, the Atmel Studio recognizes it and lists it under 'Tool'. After selecting the appropriate programmer the programming device must be chosen. In this case it is the ATmega128RFA1. Selecting JTAG as interface and pressing 'Apply' will open the available options, as can be seen in the next figure, for programming and reading the target device.

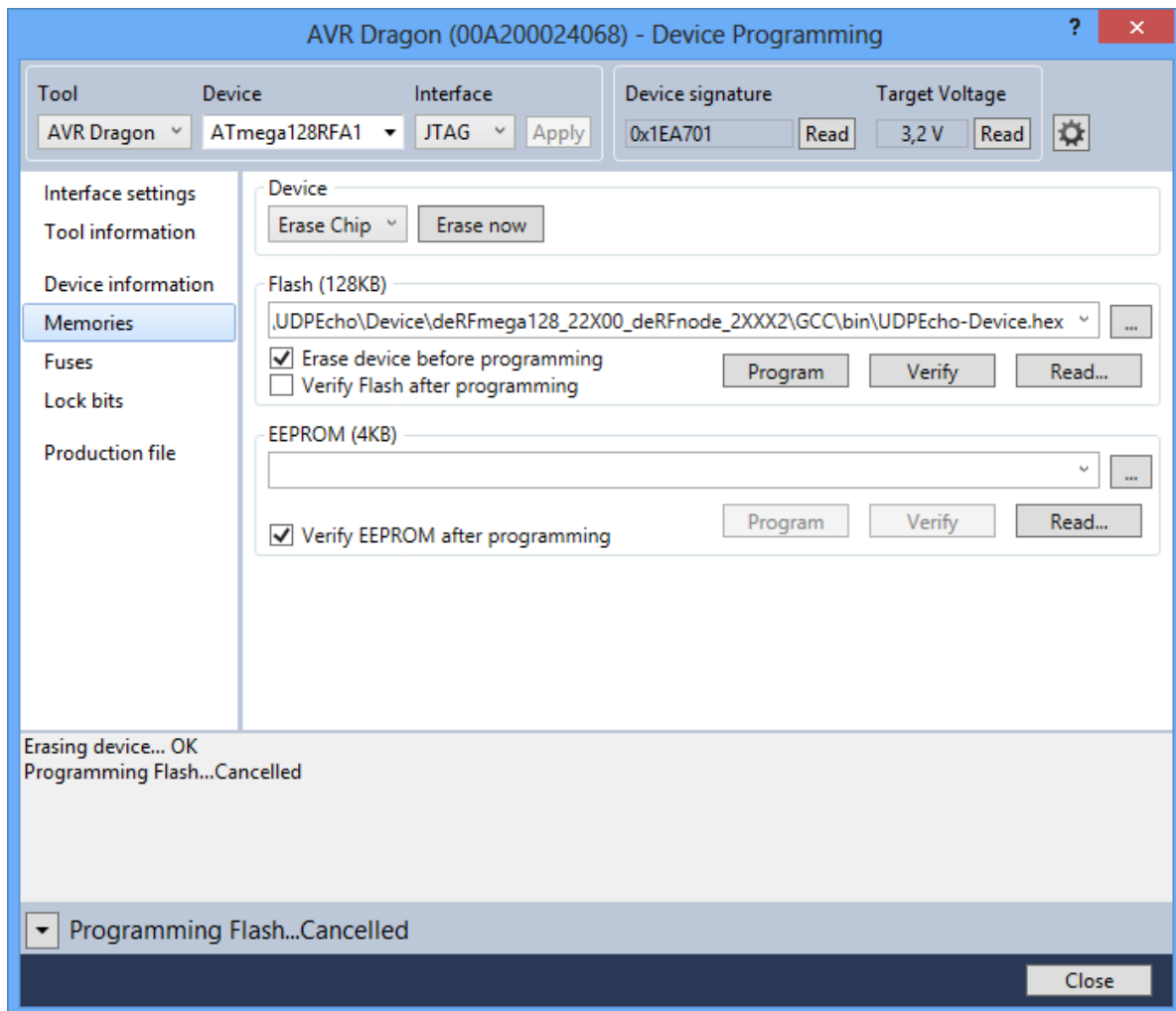


Figure 57: The programming interface

Under the 'Flash' Field the device programming file can be chosen. Pressing the '...' button will open a file selection dialog to select a firmware file. Pressing the 'Program' button will download the application to the target device.

Note: If programming fails, ensure device and programmer are powered as well as device drivers have been installed properly. Also check if the JTAG connector is plugged in the right direction.

Subsequently some important so called 'fuses' of the AVR device have to be set. The fuses are configuration options and determine the way the AVR is generating its CPU clock, getting its program memory contents and so on. The fuse settings can differ for each application, so please refer to the application Readme.txt for the correct fuse settings.

Attention: Do not experiment with the fuses as you might destroy important information on the target (e.g. erase the EEPROM content by disabling the EESAVE fuse) or even lock yourself out from any further programming actions (e.g. disabling the JTAGEN fuse).

The fuse settings in **Figure 58** are valid for each 6LoWPAN demo application.

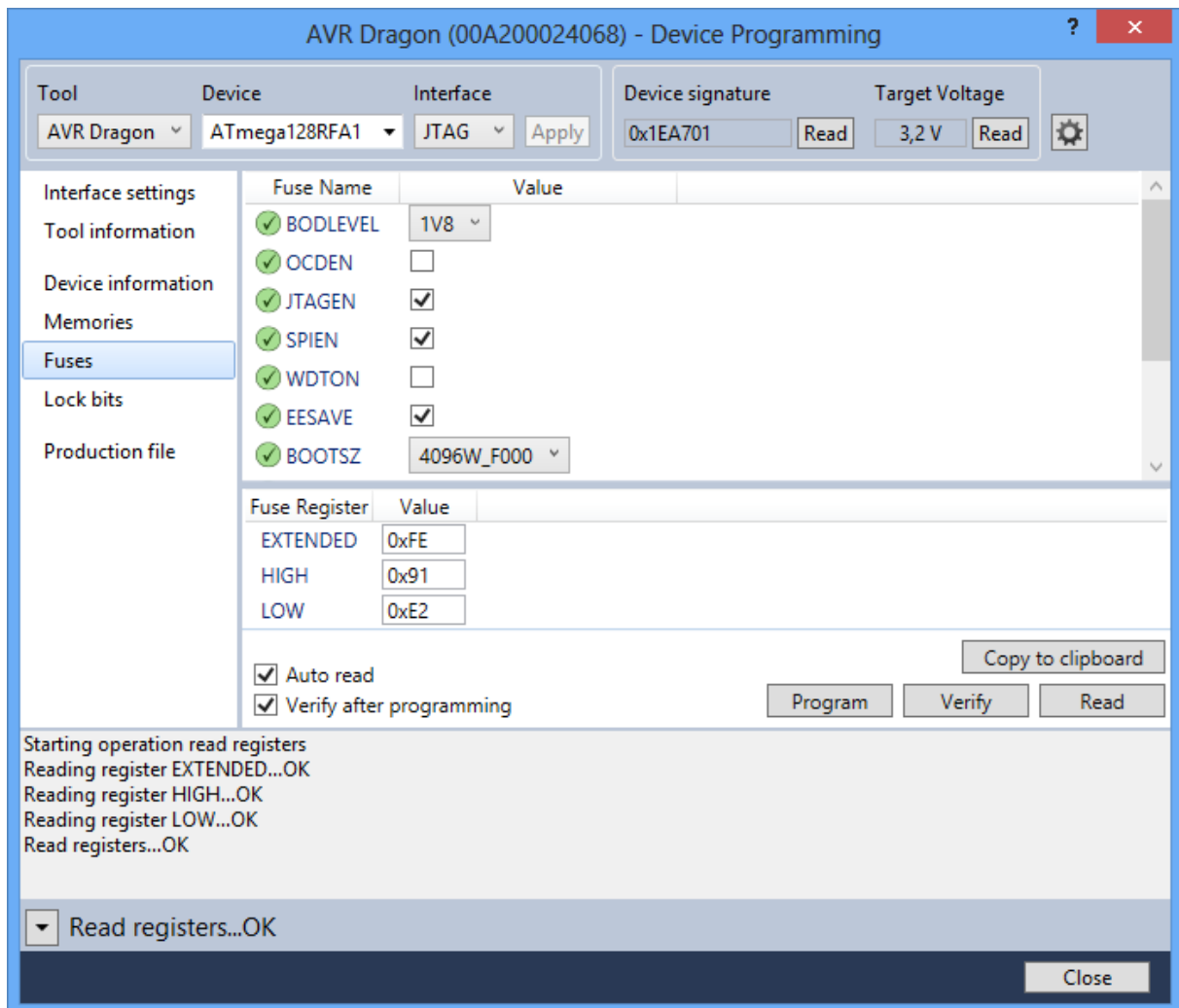


Figure 58: Setting fuses using the Fuses view

Select the required fuse options in the window and press the button 'Program' to set the fuses in the AVR device.

After the fuses are programmed verification is recommended. This can be done automatically by checking the 'Verify' option. If activated (as in **Figure 58**) the 'Reading register ...' informs about the successfully programmed fuses.

If you do not intend to debug your application, the programming is finished and your application will start each time you power up the device.

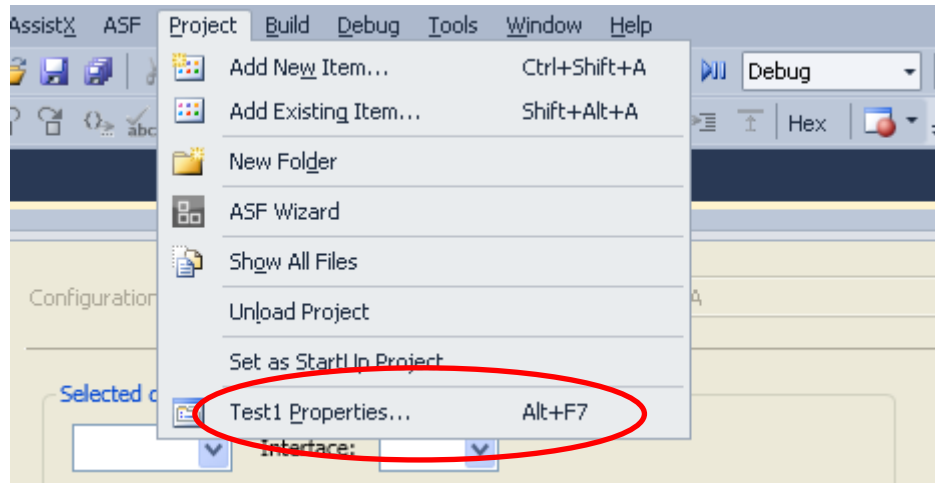
If you need to debug your application the programming step can be omitted since the Atmel Studio can do this – and by default does – as the first part of the debugging session.

6.2.1.7. Debugging the application

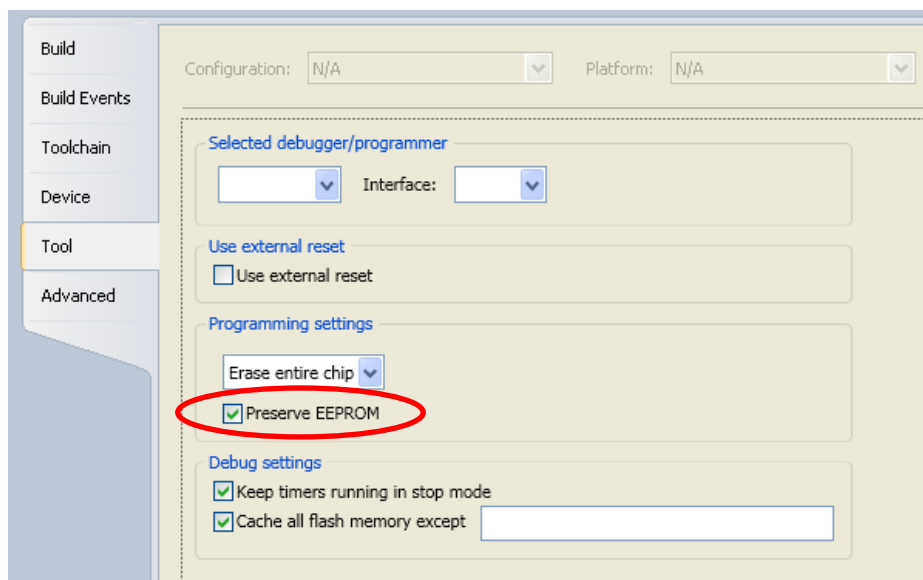
The selection of the programmer and target device is already done, so a debugging session is immediately possible. To be able to debug, the binary and ELF files must be in the same location where the project file (*.atsln) is located.



Attention: Atmel Studio has a default setting of programming before debugging and erasing the EEPROM content in that process. As the EEPROM contains valuable manufacturer data the nodes need to successfully initialize the 6LoWPAN stack, it is highly recommended to set the 'Preserve EEPROM' check box in the tool settings. You can reach the project settings via the Project menu selecting the Properties entry.



Within the properties view select Tool and in the section 'Programming settings' you can now activate the 'Preserve EEPROM' check box.



Do not forget to save the new settings.

To start your debugging session follow the steps in the pictures below:

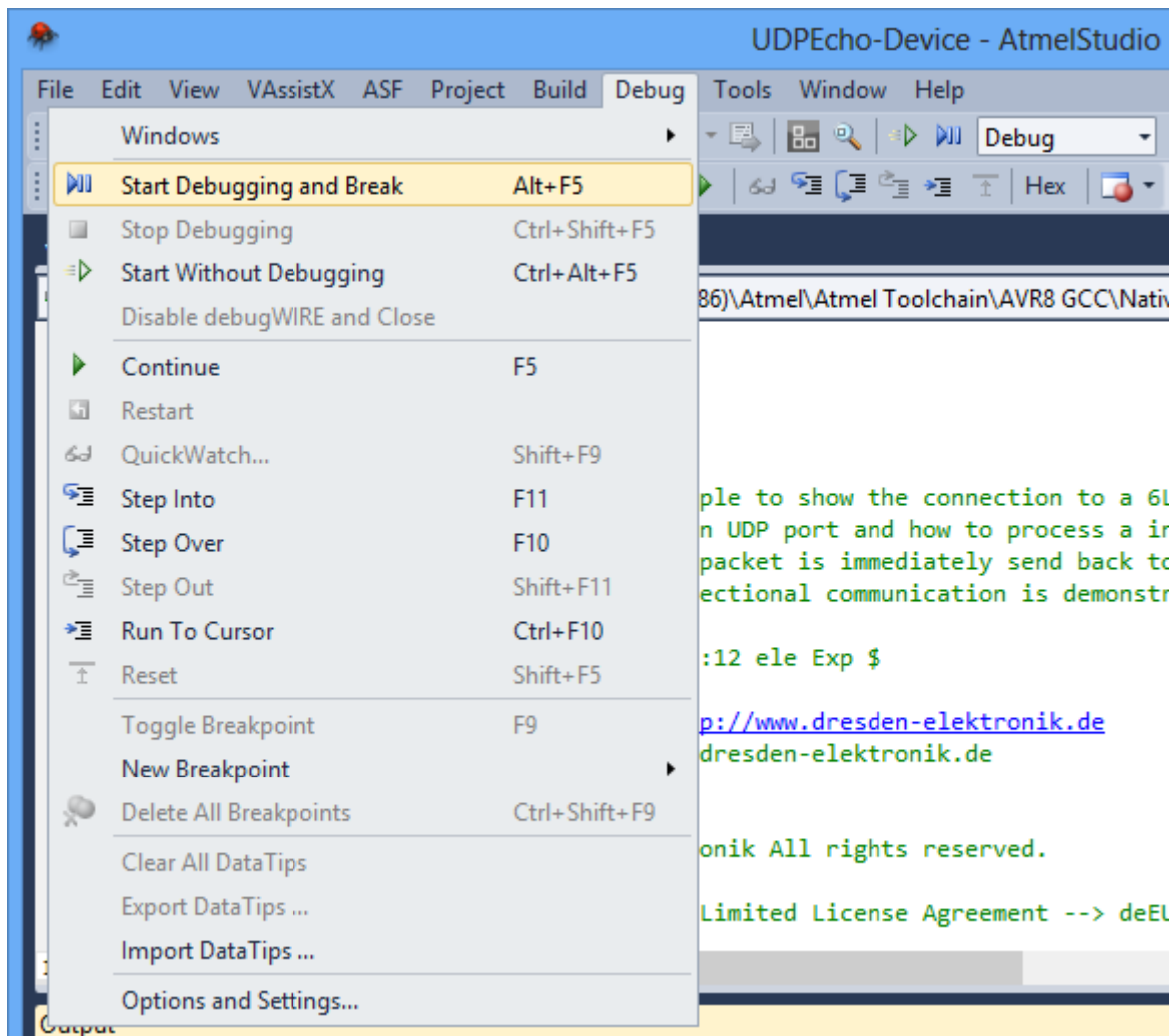


Figure 59: Start debugging session

After the debugging has been started, the target device will start execution until a breakpoint is reached. A breakpoint could easily be inserted by right clicking into the source code and select 'Insert Breakpoint' under the menu item 'Breakpoint' (as demonstrated in **Figure 60**).

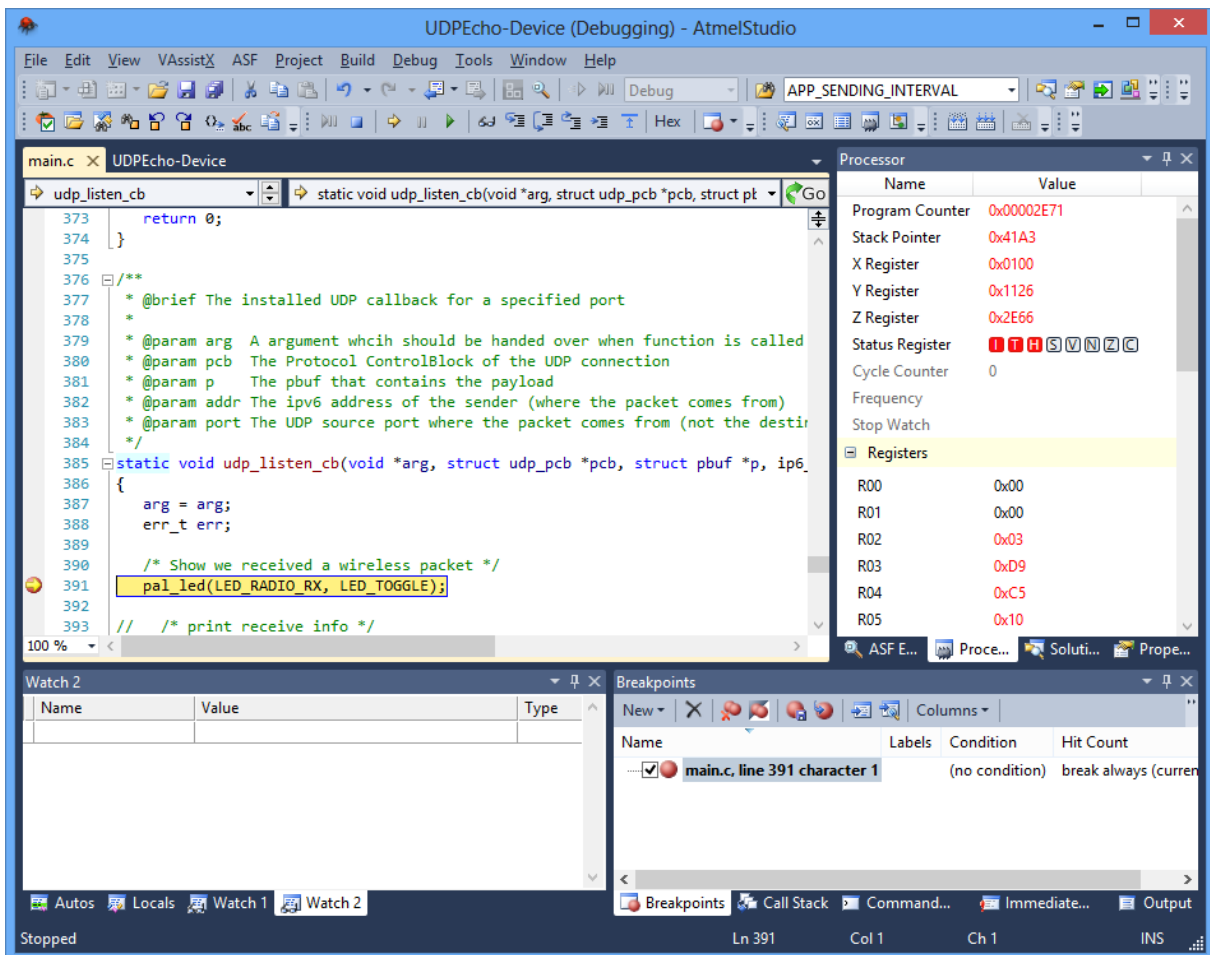


Figure 61: Debugging breakpoint execution

6.2.1.8. Basic Troubleshooting

In case of any operational problem with your setup, please check the following:

1. Check the power first, and make sure that all of your equipment is properly connected.
2. Verify that the PC conforms to the minimum system requirements.
3. Verify that the PC USB or UART interface is working and that the correct drivers are installed.
4. Check that you have set up the hardware according to specific kit instructions.
5. Make sure you have programmed the right images and set the correct fuse values. Resetting the node may be required.



6.2.2. ARM Development

Developing custom applications and debugging them requires an appropriate IDE. The Eclipse IDE is recommended. Further on compiler and supplementary tools are needed. Finally an appropriate programmer has to be chosen.

How to work with the Eclipse IDE together with more information about the other prerequisites is described in the following sections.

6.2.2.1. Setup IDE

The Eclipse IDE setup is quite simple and involves the following steps:

1. Download the Eclipse IDE for C/C++ Developers at www.eclipse.org/downloads. It comes as a ZIP archive.
2. Simply extract it to a directory of your desire. An installation is not required.

Debug Plugin

Debugging is an optional feature. If required and when working with the Eclipse IDE, it is recommended to install the Zylin plugin.

1. Start the Eclipse IDE
2. Go to the 'Help' Menu, select 'Install New Software ... '.
3. Click the 'Add' button, enter the URL <http://www.zylin.com/zylincdt> in the location field, assign a name and confirm the dialog with 'OK'.
4. In the parent window, enable the checkbox for 'Zylin Embedded CDT', continue with a click to 'Next'. After a dependency check the download will start.
5. Wait until the download completes, then accept the license conditions and confirm the security warning.
6. Wait until the installer completes.
7. Confirm to restart the IDE for applying changes.

The second step is to install an application which interfaces the eclipse IDE with the debugger/programmer. In this case, it is required to download and to install the Segger Tools.

Flash Tool

Uploading the firmware to the target requires additional software. When using the SAM-ICE programmer as suggested SAM-BA is required. The latest version can be downloaded without registration from

<http://www.atmel.com/tools/ATMELSAM-BAIN-SYSTEMPROGRAMMER.aspx>

Afterwards start the installer and follow the given instructions.



6.2.2.2. Tool chain setup for ARM-based MCUs

The following section describes how the tool chain has to be set up that is required for compiling and debugging firmware for the following microcontrollers.

- ARM7S
- ARM7X
- Cortex M3 ('SAM3')

Due to licensing issues it is not possible to redistribute all needed tools on the current CD. Missing components must be downloaded directly from their producer.

Compiler/Supplementary Tools

To set up the arm-none-eabi based YAGARTO GNU ARM tool chain (version 2011/03/28); please start the installer directly from the delivered CD. It can be found in the directory '3rdParty\Yagarto'.

The tool chain consists of the following parts:

- Binutils-2.20.1
- Newlib-1.19.0
- GCC-4.6.0
- GDB-7.2

It is also recommended to install the YAGARTO Tools. This installer is also available on the delivered CD in the directory '3rdParty\Yagarto'.

The YAGARTO tools provide typical UNIX-like commands such as make, sh, touch, uname and more.

6.2.2.3. Suggested programmer for ARM-based microcontrollers

dresden elektronik suggests to use the following programmer device:



Figure 62: Atmel SAM-ICE

The proposed device enables programming of firmware files as well as debugging applications.

Notes:

1. Even if there are many other programmers on the market, they are not listed above since they are unsupported.
2. JTAG Programmers for ARM devices may not be used to program AVR devices and vice versa.
3. A programmer is not contained in any development kit. It has to be purchased separately.
4. For programming deRFusb radio dongles, a SAM-ICE adapter may additionally be required.
5. Some SAM3-based deRFusb radio dongles do NOT have a JTAG connector soldered. In this case, JTAG programming/debugging requires the connector to be soldered manually. Pure firmware programming may be done over the USB interface.

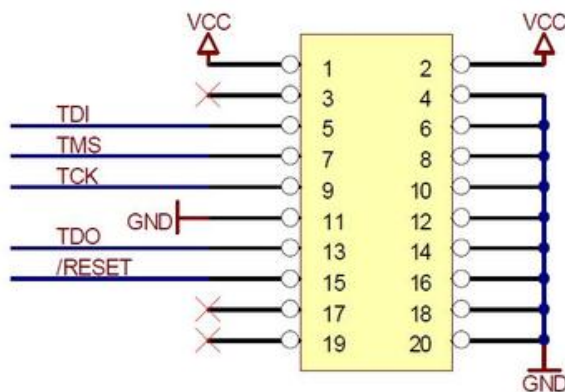


Figure 63: Pinout for 20-pin ARM JTAG header

6.2.2.4. Connecting the target

To program a device, the programmer must be connected to the respective JTAG header and the target needs to be powered.

On the different dresden elektronik products, several JTAG programming headers are available:

- 10-pin combined JTAG/Debug header,
- 20-pin ARM-JTAG header

Note: Not all boards (e.g. deRFmodules) have a JTAG header. In those cases an additional board may be required for providing this header (e.g. deRFnode, deRFgateway, deRFbreakout Board).

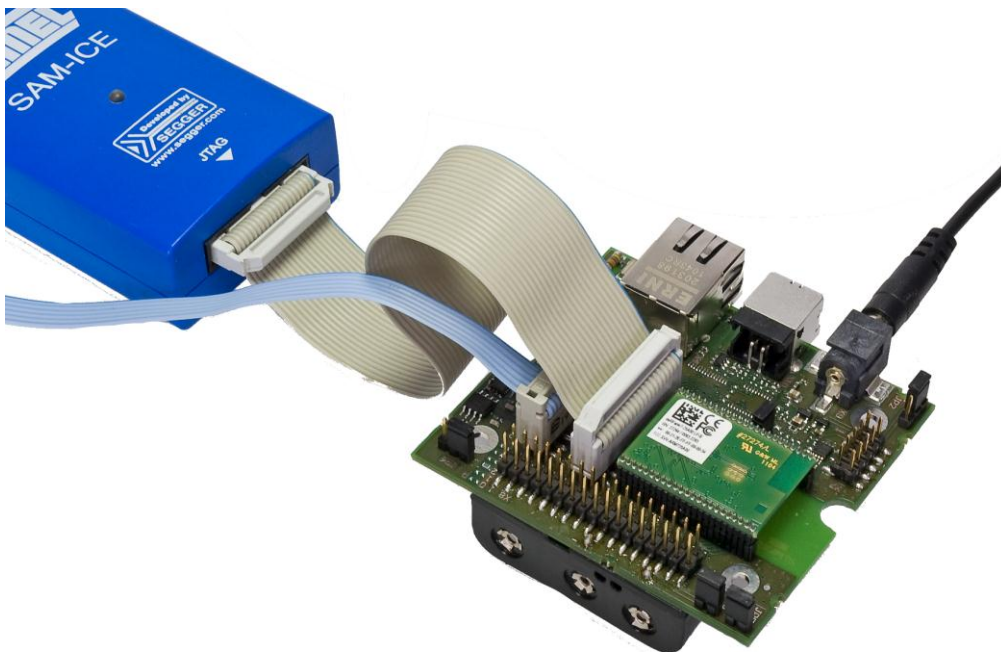


Figure 64: SAM-ICE connected with deRFarm7 plugged onto a deRFgateway

Note: The blue cable is not required for programming. It connects to a dresden elektronik USB level shifter stick to a PC for serial debug output. Although not required, a trace (debug output) connection is largely helpful for any embedded development.

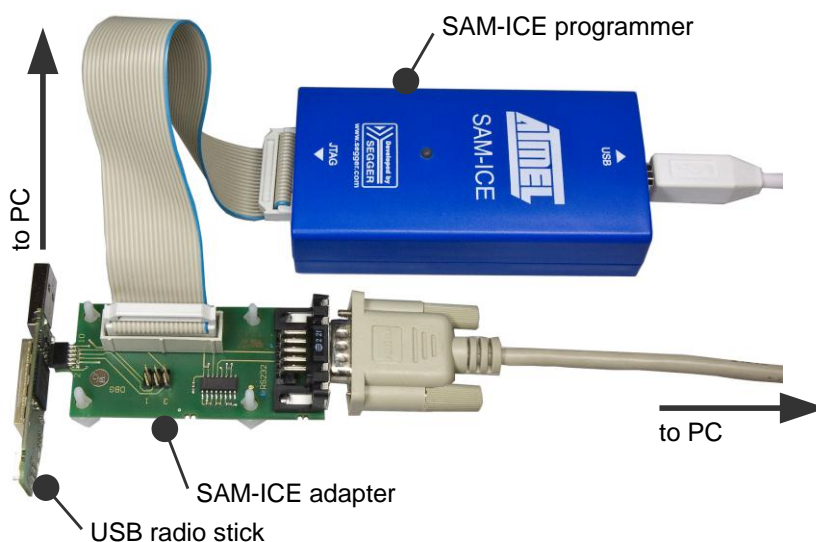


Figure 65: USB radio stick with SAM-ICE adapter and programmer

More technical background is given in [9]; it is described in detail how to change the firmware on SAM3S based USB sticks from dresden elektronik.



6.2.2.5. Importing a project into Eclipse

Setting up the Eclipse IDE is described in **Section 6.2.2.1**. Setting up the tool chain for ARM-based MCUs is described in **Section 6.2.2.2**.

This section describes how to import a package such as the IEEE 802.15.4 MAC Stack into the Eclipse workspace. This import requires the following steps:

1. Choose a workspace.
2. Create a new project.
3. Import the IEEE 802.15.4 MAC Stack into the workspace.
4. Create build targets.
5. Compile the application.

After unpacking the Eclipse IDE and starting it for the first time a workspace location has to be chosen. If you check the 'Use this as the default and do not ask again' option this dialog will not appear again but you still can switch workspaces via the Eclipse menu commands.

- Open the Eclipse IDE.
- Choose a workspace directory.

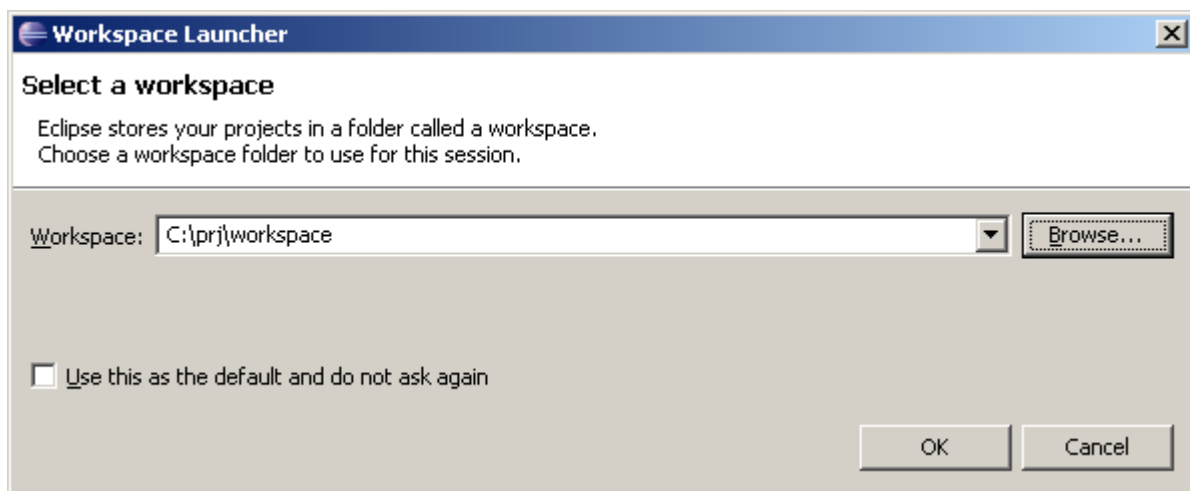


Figure 66: Workspace Launcher

To create a new project either use the File menu or right click into the Project Explorer 'New Project | C Project' and chose in the following dialog 'Makefile Project | Empty Project | -- Other Toolchain --'.

- Create a new project.

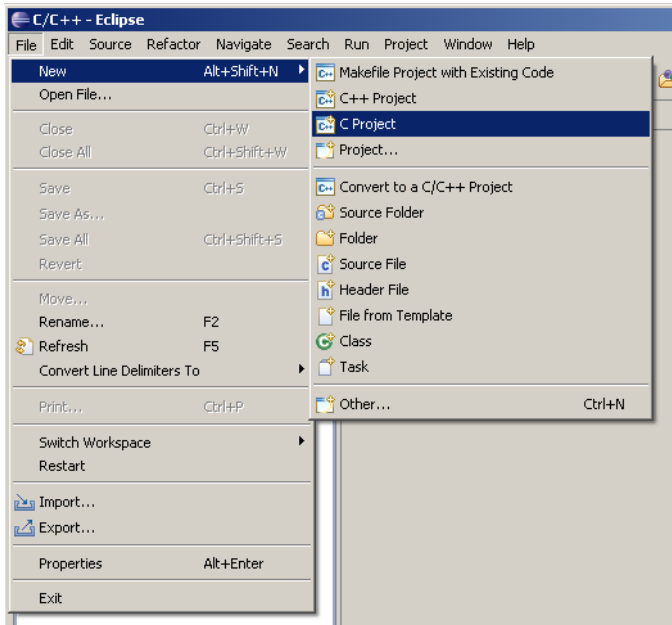


Figure 67: New C project

- Assign any project name such as 'MAC Stack'.
- Check 'Use default location' to create the new directory in your previously chosen workspace.
- Choose 'Makefile Project' as Project type.
- Select '-- Other Toolchain --'.
- Click 'Finish' to create the empty project.

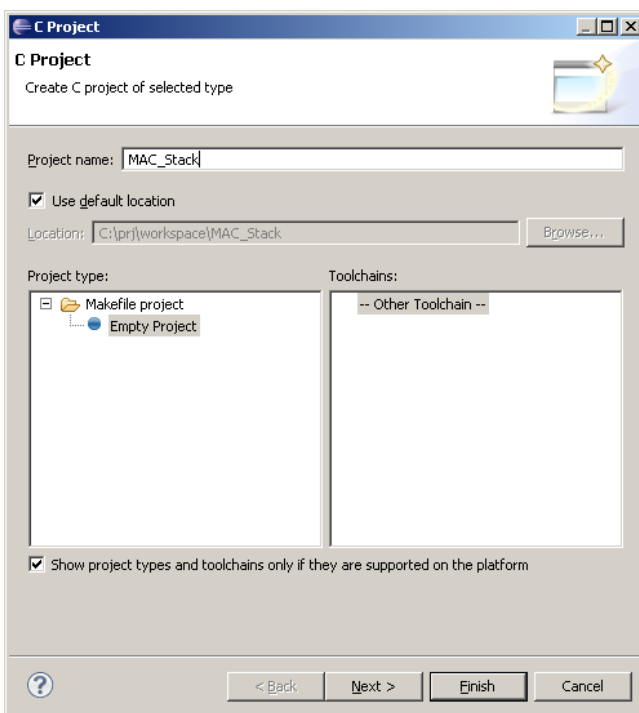


Figure 68: Create a new C project



- Import the IEEE 802.15.4 MAC Stack files by clicking 'File | Import'.

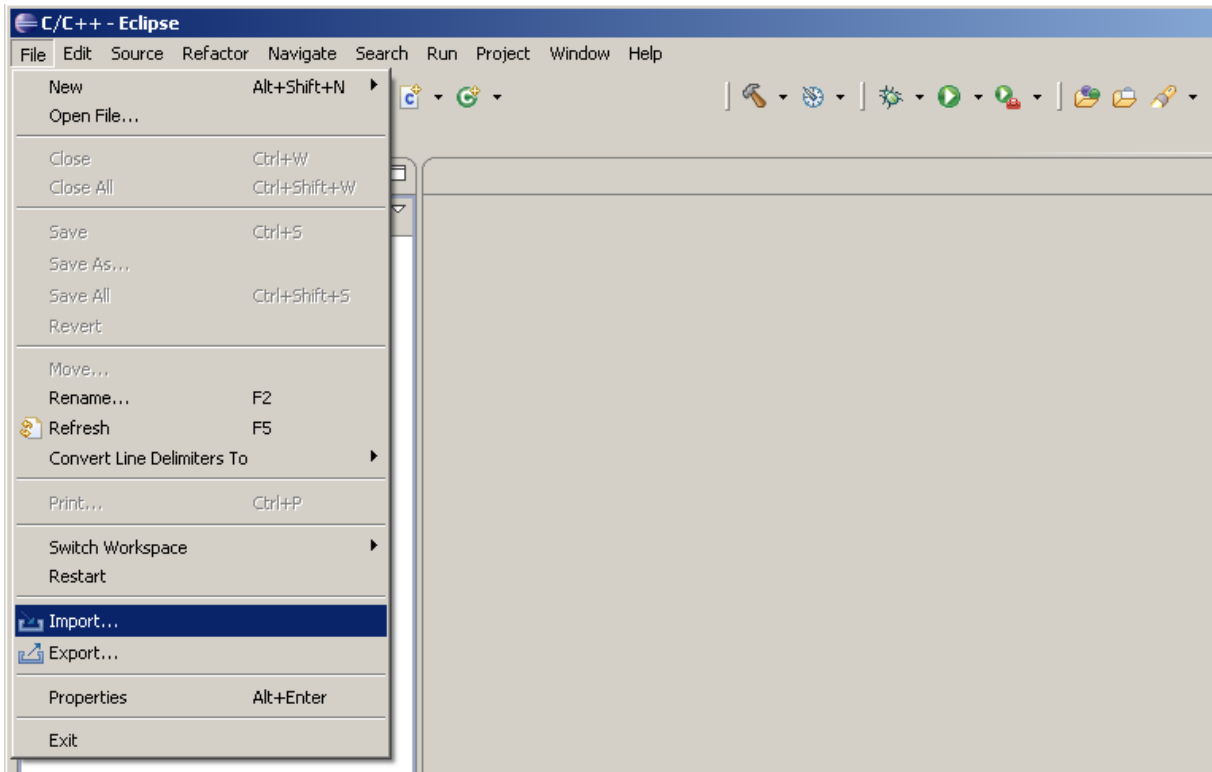


Figure 69: Import files into the project

- Choose 'File System' as import source.

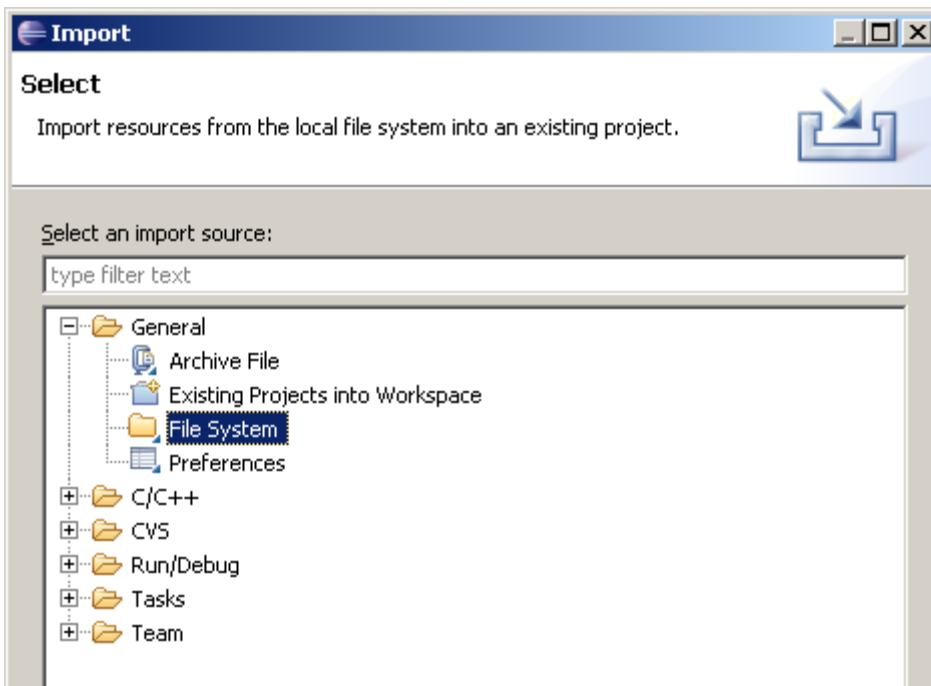


Figure 70: Choose the import source

Press 'Next' and click on 'Browse...' in the next dialogue.

- Select the path to the IEEE 802.15.4 MAC Stack <MAC-root-directory>.
- Press Finish to import the source files to your local workspace.

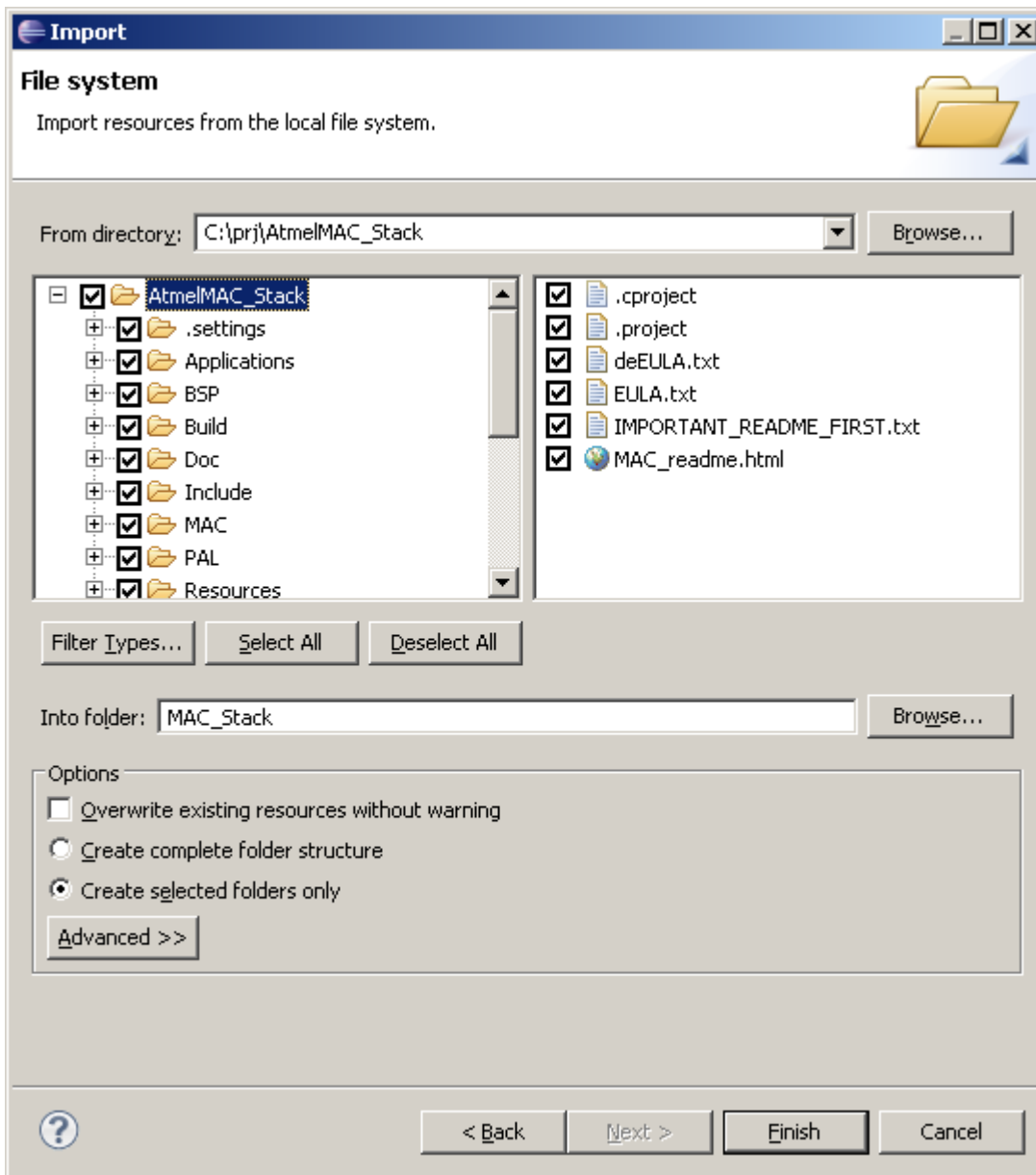


Figure 71: Select the folder that contains the source files you want to import



In the next step the actual binary will be built.

- Create a new make target by switching to the 'Make Target' view on the right-hand side of the Eclipse window.
- Browse to the application you want to build for example the 'Simple Sensor Network' which can be found for ARM under 'Applications | MAC_Examples | Basic_Sensor_Network | AT86RF231_AT91SAM7X512_deRFarm7_25X00_deRFgateway_1XXX2'
- Browse to the folder 'GCC' where the Makefile is located

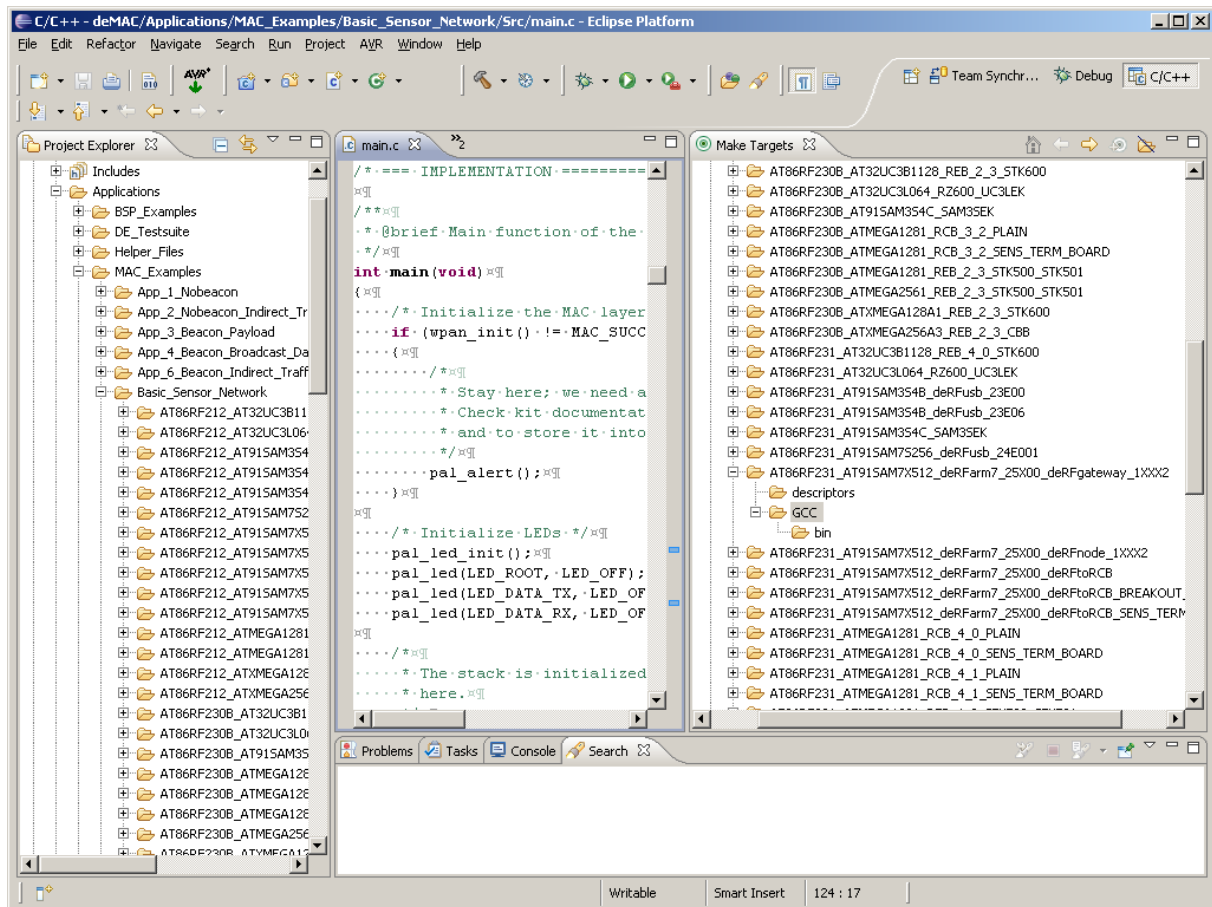


Figure 72: Switch to 'Make Target' View and locate to your application

- Right click on 'GCC' folder and select 'New...'
- Within the appearing dialog type "all" into the field 'Target name'.
- Repeat the previous two steps with Target name "clean".

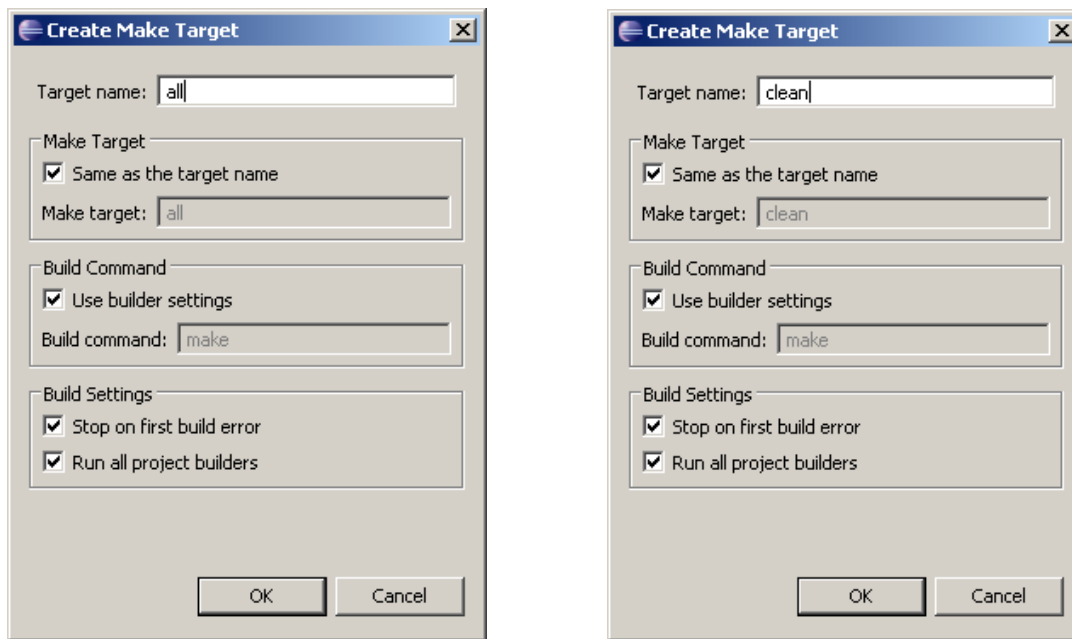


Figure 73: Target name 'all' and 'clean'

- Double-click on 'all' to build the project. During the build process you can see in the console which projects are being compiled and linked and how large they are.

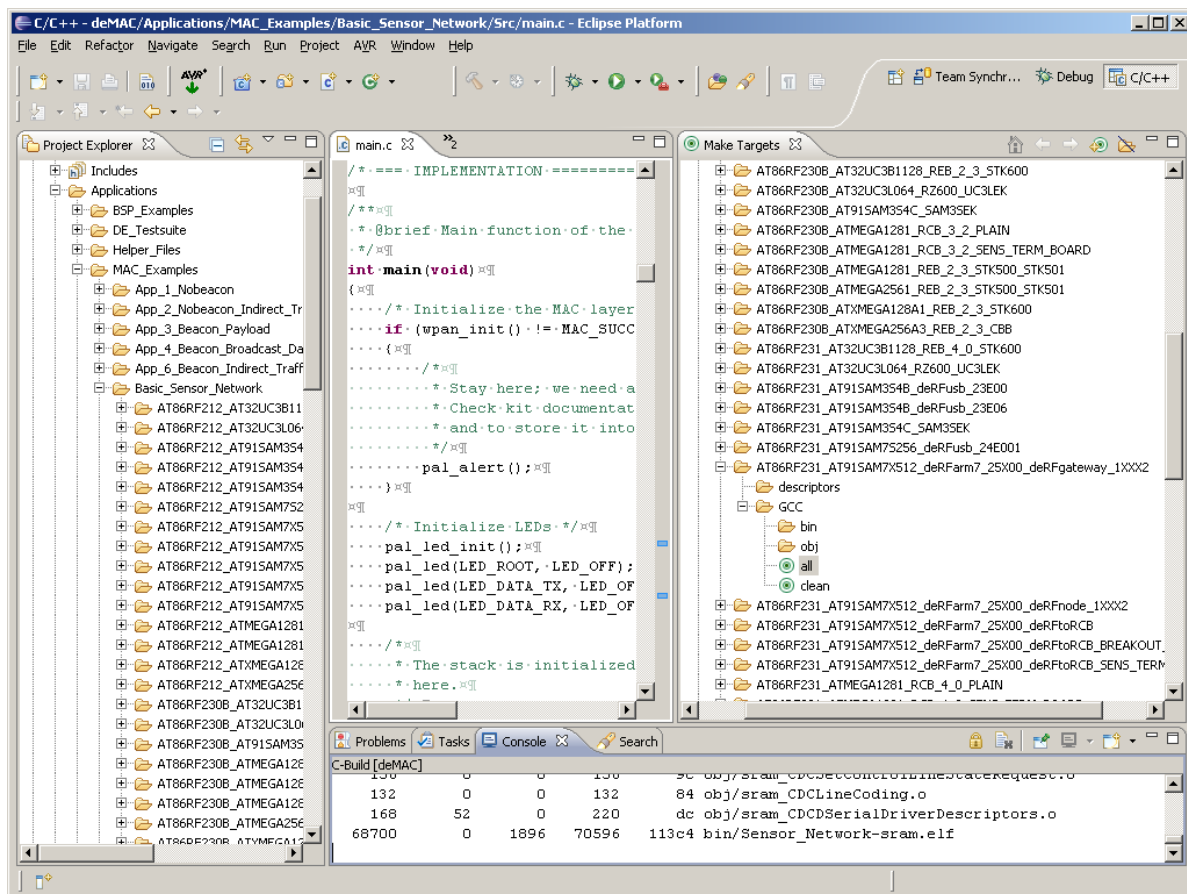


Figure 74: Finished ARM build process



-
- The build process is the same either for ARM and AVR processors.
 - The 'make' program is doing all the build magic. You just need a properly installed AVR and/or ARM tool chain.
 - To build another application just locate to the corresponding 'GCC' folder, generate 'all' and 'clean' target and double-click 'all'.
 - The 'clean' target deletes the binaries and objects generated.
 - With the build successfully done you are now ready for downloading the application to your target.



7. Analyzer

The analyzer variants of the deRFusb-23E00 or deRFusb-13E00 USB radio sticks are optimally matched for Perytons professional network and protocol analysis software or Luxoft's free Bitcatcher.

7.1. Perytons Analyzer

The USB radio stick in the dresden elektronik development kit 6LoWPAN 2.4 GHz is already programmed with a firmware for the Perytons analyzer.

To get the free 30-day trial it is necessary to click onto Perytons website;
<http://www.perytons.com/downloads>

Here it is required to enter the contact information (email address, name, phone number and organization) and press 'SUBMIT'. A link to the download will then be sent to your email address.

With the valid 30-day trial license you can monitor network structures as well as observe data flows and runtime performance in detail without additional effort. Especially during the development phase this protocol analysis software will serve as an essential and valuable tool. Compared to other analyzer tools the dresden elektronik USB radio stick facilitates synchronous sniffing of all 16 channels. The analyzer is operating with an accuracy of 1 μ s.

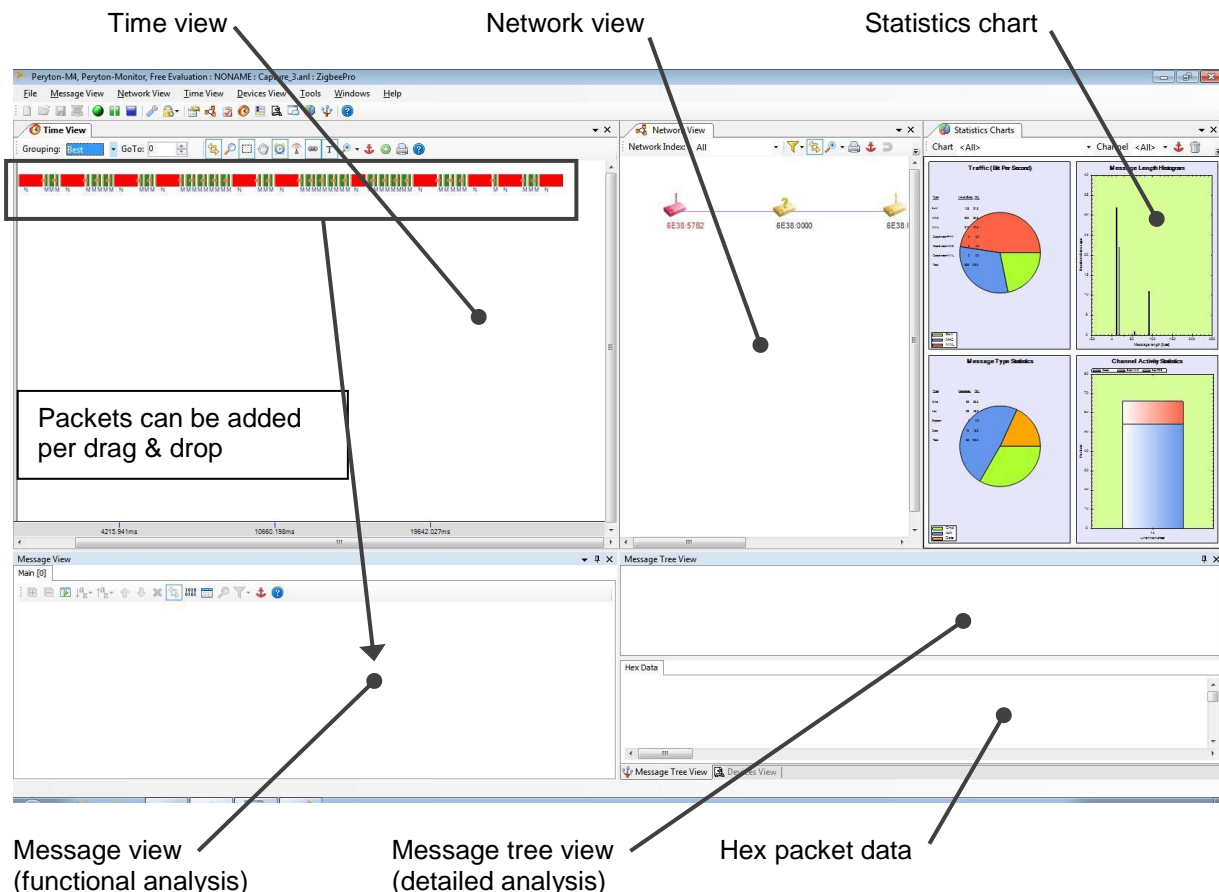


Figure 75: GUI analyzer overview



The analyzer software of Perytons supports a number of standard protocols such as for example ZigBee, ZigBee RF4CE, 6LoWPAN/ZigBeeIP, IEEE 802.15.4 MAC or proprietary protocols. For more information about the analyzer GUI please refer to the Perytons™ Protocol Analyzer Overview [13].

All three Perytons software versions are supported for the IEEE 802.15.4 radio range:

Version	Description	Number of required USB sticks
Peryton-S	Single-channel analyzer	1
Peryton-D	Like S with antenna diversity	2
Peryton-M	Multi-channel analyzer	Channels + 1

Start the Perytons Analyzer via the Start menu 'Perytons | Perytons'.

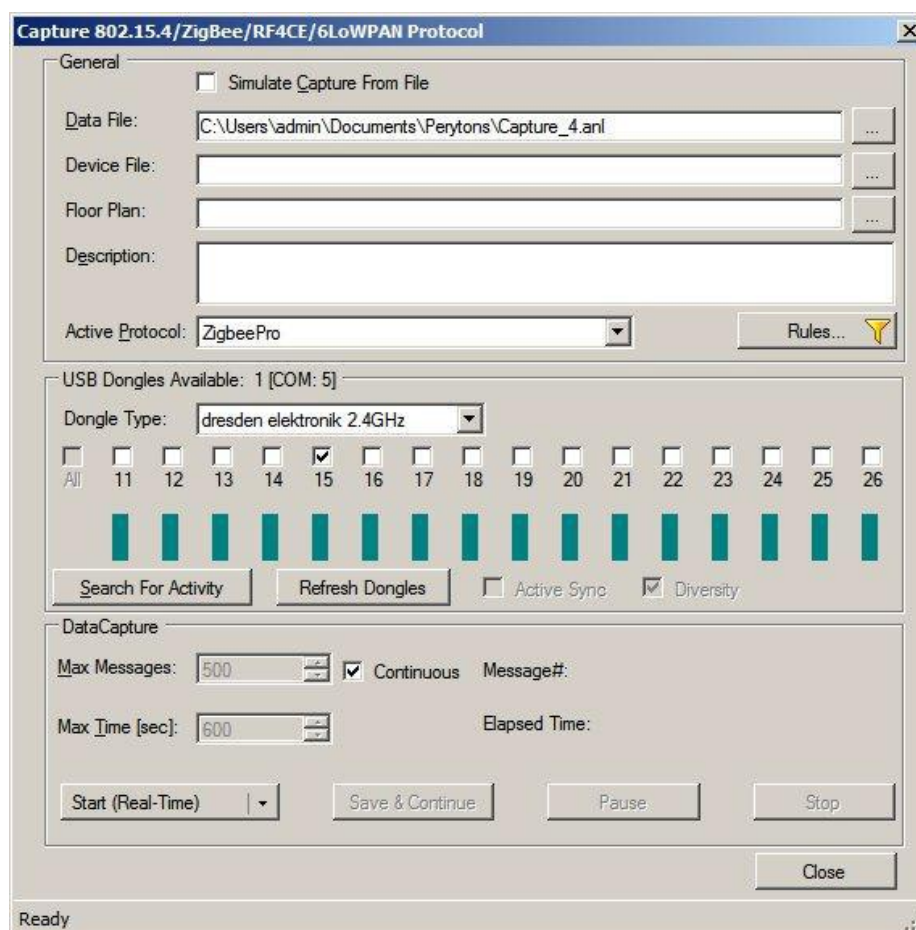


Figure 76: Starting the analyzer

In the appearing capture dialog choose the 'dresden elektronik 2.4GHz' as dongle type. Then do the following steps:

1. Select the channel.
2. Set up the protocol (e.g. 6LoWPAN or ZigBee Pro).
3. Start the session by pressing 'Start (Real-Time)'.



Perytons evaluation version 4.0 includes:

- The Perytons-M4 Analyzer - Evaluation version 4.0 (50Mbyte) for IEEE 802.15.4 - 2006, ZigBee Home Automation (HA), ZigBee Commercial Building Automation (CBA), ZigBee Smart Energy (SE), ZigBee Telecom Application (TA), ZigBee Health Care (HC), ZigBee RF4CE (GDP, ZRC, ZID, Z3D), ZigBee Light Link (ZLL), ZigBee Retail (ZRS), ZigBee IP and 6LoWPAN.
- Support of a variety of off-the-shelf Hardware vendors for 2.4 GHz and sub-GHz capture.
- Sample capture files.
- Help files and User Manual.
- A free 30-days license.

7.2. BitCatcher

Alternatively available for network and protocol analysis are the deRFusb23E00 and deRFusb13E00 USB radio sticks. These analyzer variants are optimally matched for Luxoft's easy-to-use network and protocol analysis software BitCatcher.

BitCatcher allows the monitoring of complex network structures as well as the observation of data flows and runtime performance in detail. The analyzer sticks are available as a single-channel sniffer for the sub and 2.4 GHz frequency range.

The BitCatcher comes with the preloaded wireless protocol descriptions for IEEE 802.15.4 MAC, ZigBee PRO, ZigBee Smart Energy, and ZigBee Light Link and proprietary applications.

The BitCatcher network analyzer software package is available for free download at www.luxoft.com/bitcatcher. BitCatcher runs on both Windows and Linux operating systems and required drivers are included in the package.



8. Revision notes

Up to now for the individual components of the deRFdevelopmentKit technical problems, malfunctions or any other critical issues are not known.

Concerning software packages and applications also possible restrictions, comments or hints are given within this document.

9. References

- [1] Datasheet Radio modules deRFarm7-25A00|25C00.pdf
- [2] User Manual Radio Modules: deRFarm7-x5A0x-x5C0x-BHB-en.pdf
- [3] Datasheet Radio modules deRFmega128-22A00|22C00.pdf
- [4] User Manual Radio Modules: deRFmega128-22A0x-22C0x-BHB-en.pdf
- [5] Datasheet deRFnode-2TNP2.pdf
- [6] Datasheet deRFgateway-1TNP2.pdf
- [7] User Manual: deRFnode_deRFgateway-BHB-en.pdf
- [8] Datasheet deRFusb-23E00_JTAG.pdf
- [9] User Manual: deRFsoftware_Programming_Manual-BHB-en.pdf
- [10] User Manual USB Radio Sticks: deRFusb-BHB-en.pdf
- [11] Atmel AVR2033: SAM-ICE Adapter - Hardware User Manual
- [12] User Manual 6LoWPAN, Developer and User Guide: 6LoWPAN-BHB-en.pdf
- [13] Perytons™ Protocol Analyzer Overview
<http://www.perytons.com/files/PerytonsAnalyzerOverview.pdf>



dresden elektronik ingenieurtechnik gmbh
Enno-Heidebroek-Straße 12
01237 Dresden
GERMANY

Phone +49 351 - 31850 0
Fax +49 351 - 31850 10
Email wireless@dresden-elektronik.de

Trademarks and acknowledgements

- IEEE 802.15.4™ is a trademark of the Institute of Electrical and Electronics Engineers (IEEE).
- ZigBee® is a registered trademark of the ZigBee Alliance.
- Atmel® Atmel logo and combinations thereof, AVR™, ARM™ and BitCloud® are registered trademarks or trademarks of Atmel Corporation or its subsidiaries.
- Perytons™ is a trademark of the Perytons Ltd.
- Windows® and others are registered trademarks or trademarks of Microsoft Corporation in U.S. and or other countries.

All trademarks are registered by their respective owners in certain countries only. Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

Disclaimer

This note is provided as-is and is subject to change without notice. Except to the extent prohibited by law, dresden elektronik ingenieurtechnik gmbh makes no express or implied warranty of any kind with regard to this guide, and specifically disclaims the implied warranties and conditions of merchantability and fitness for a particular purpose. dresden elektronik ingenieurtechnik gmbh shall not be liable for any errors or incidental or consequential damage in connection with the furnishing, performance or use of this guide.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's personal use, without the written permission of dresden elektronik ingenieurtechnik gmbh.

Copyright © 2013 dresden elektronik ingenieurtechnik gmbh. All rights reserved.