ZigBeeNet™ Software 1.0

Application Note

Serial Bootloader

Executive Summary

The Application Note provides quick instructions on using Serial Bootloader. The dedicated equipment and targeted applications are outlined in this document. Also the manual contains the comprehensive information on command line options and practical details concerning software employment.

Related documents:

[1] ZigBeeNet™ Product Datasheet. MeshNetics Doc. M -252~08

[2] ZigBit™ OEM Modules. Product Datasheet. MeshNetics Doc. M-251~01

[3] ZigBit™ Development Kit 2.0 User's Guide. MeshNetics Doc. S-ZDK-451

[4] ZigBit Ethernet Gateway 1.0 User's Guide. MeshNetics Doc. P-ZGTW-451

[5] AVR Studio User Guide. Available in HTML Help with the product.
http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

[6] JTAGICE mkII Quick Start Guide.
http://www.atmel.com/dyn/resources/prod_documents/doc2562.pdf

Overview

Serial Bootloader is a software utility designed to program an application code into WSN nodes through USB or RS-232 port, without using JTAG. It also provides the possibility to set up the network parameters for each node without affecting its firmware image. Serial Bootloader is a part of ZigBeeNet Software [1].

Serial Bootloader supports programming of MeshNetics ZigBit OEM modules [2] installed on Meshbean2 board [3], ZigBit Ethernet Gateway [4] or any another custom device. In order to be programmed by means of Serial Bootloader device should be connected to PC through USB or RS-232 port.

Serial Bootloader consists of two parts: a PC console application for Windows platforms and bootstrap code residing in the MCU.

To use Serial Bootloader you should download the bootstrap code, and set up device fuse bits when needed by means of JTAG. Each MeshBean2 board is delivered with fuse bits set up and the bootstrap preloaded to ZigBit MCU. You have to set up any other device using ZigBit as described above.

Setting Fuse Bits

To enable programming of a node by Serial Bootloader fuse bits are to be set up for ZigBit as follows: **`0xFF, 0x9C, 0x62`**.

In order to ensure these fuse bits are set, check ON the following options in Fuses Tab using AVR Studio [5]:

```
Brown-out detection disabled; [BODLEVEL=111]
JTAG Interface Enabled; [JTAGEN=0]
Serial program downloading (SPI) enabled; [SPIEN=0]
Boot Flash section size=1024 words Boot start
address=$FE00;[BOOTSZ=10]
Divide clock by 8 internally; [CKDIV8=0]
Int. RC Osc.; Start-up time: 6 CK + 65 ms; [CKSEL=0010
SUT=01]
Boot Reset vector Enabled (default address=$0000);
[BOOTRST=0]
```

Uncheck the rest of options, and write the fuse bits to device. Make sure the above hex value string appears at the bottom of Fuses Tab.

It is recommended to use Atmel JTAGICE mkII emulator [6] together with AVR Studio to set up fuse bits.

IMPORTANT NOTE:

Be careful using JTAG for setting up the fuse bits! If you set wrong fuse bits by JTAG, your device would not work.

Setting MAC Address

To communicate within WSN network each node must be identified with a unique MAC address. In general, MAC address can be specified for a node in the following way:

1) by hardware pre-configuration

2) by loading an image file containing MAC address to a node

3) by means of Serial Bootloader command options.

Command Options

Serial Bootloader accepts the following options:

```
bootloader –p port_number [–f file_name] [-b baud_rate] [-h]
[-s bootstrap_size] [-M MAC address] [-C Channel mask]
[-P PANID]
```

The table below contains the description of the options.

Option	Description	Default Value
-p port	COM port	
-f file_name	Name of Motorola SREC file	
-b baud_rate	Baud rate in bits per second (1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200)	38400
-h	Hardware flow control, if used	None
-s size	The size of bootstrap code, in words (512, 1024, 2048, or 4096)	1024
-M MAC	MAC address in HEX format to be assigned to the node	
-C channel mask	Channel mask in HEX format to be assigned to the network	
-P PANID	PANID in HEX format to be assigned to the network	

The order you use the options above doesn't matter. Any of them can be omitted except for `-p` option.

IMPORTANT NOTES:

Serial Bootloader is designed so that, if `-M`, `-C` or `-P` option is present in the command line, the corresponding parameter stored in the EEPROM would be overwritten. Furthermore, the relevant value(s) which was set inside the downloaded image file will be ignored.

If –f option is not specified (i.e. without image file download) you can use any other command option to change the EEPROM settings of the node without interfering with the application code downloaded before.

Usage Examples

```
bootloader –f wsndemo.srec –p COM5 –M 1 –C 100000 –P 5320
```

The above command demonstrates how to load the WSN Demo image into a node connected to PC via COM5. The following parameters are assigned:

MAC address = `0x1`
Channel mask = `0x100000`
PANID = `0x5320.`

Serial Bootloader can be used apart from downloading any image:

```
bootloader –p COM5 –M 2 –C 100000 –P 5320
```

The above command is used to assign the following parameters to a node without affecting an image:

MAC address = `0x2`
Channel mask = `0x100000`
PANID = `0x5320.`

To set baud rate, flow control mode or the bootstrap code size to the default value omit the corresponding option in command line.

Programming WSN Nodes

In order to program a wireless device using Serial Bootloader, do the following:

1. Connect a device to your PC via USB or RS-232 port following the operating instructions. If USB connection is used make sure that USB to UART Bridge VCP Driver is installed as described below.
2. Run Serial Bootloader, specifying the image file, COM port and the optional keys in the command line (if required).
3. Press reset button on the device
4. Release reset button on the device. Serial Bootloader will be waiting for approximately 30 seconds for the button to be released. If this does not happen, programming would be aborted.

Serial Bootloader indicates the operation progress. Once loading is finished successfully, the device would be restarted automatically. If loading fails, Serial Bootloader would indicate the reason. In rare cases, loading process could fail due to communication errors between the device and the PC. If this happens, try to repeat programming or try to use normal RS-232 port instead of USB. If loading still fails, the previous code programmed into the device could be corrupted, and the device should be reprogrammed again.

Image File Formats

Serial Bootloader recognizes image files in Motorola hexadecimal format, also known as SREC format. Such file names have the `.srec` extension. Motorola SREC files for Serial Bootloader contain both flash memory and EEPROM images.

A user's application developed with AVR Studio can be converted into SREC format using the `AVR-objcopy` utility so it becomes downloadable via serial booting process.

Installation

Serial Bootloader is supplied as a stand-alone utility. It does not require any special installation. Just copy the program from your distribution set to the desired location on your PC.

In order to use USB port for serial connection between a device and PC you need to install a virtual COM port driver. CP210x USB to UART Bridge VCP Driver compatible with different Windows versions can be downloaded from the manufacturer's website:

http://www.silabs.com/tgwWebApp/public/web_content/products/Microcontrollers/USB/en/mcu_vcp.htm.

Install the VCP driver kit from Silicon Laboratories and connect a device to USB port. Once Windows detects new hardware, the driver installation wizard appears. Follow on-screen instructions. When installation is completed, make sure the driver is installed successfully and the new COM port is present in the hardware list. For that purpose invoke the Device Manager:
`Start/Control Panel/System/Hardware/Device Manager`
and check `'Ports (COM&LPT)'` section.

Restrictions

There are some minor restrictions on software downloadable by serial booting process. Serial Bootloader cannot rewrite the upper 2 KB of memory starting from `0xFC00` address, because the bootstrap code resides in that area.