

ZigBee-to-TCP/IP Gateway: New Opportunities for ZigBee-based Sensor Networks

Speaker: Alexandra Dmitrienko

Kiel University of Applied Sciences, Germany &
Saint-Petersburg State Polytechnical University, Russia

sdmitrienko@mail.ru, dav@efo.ru



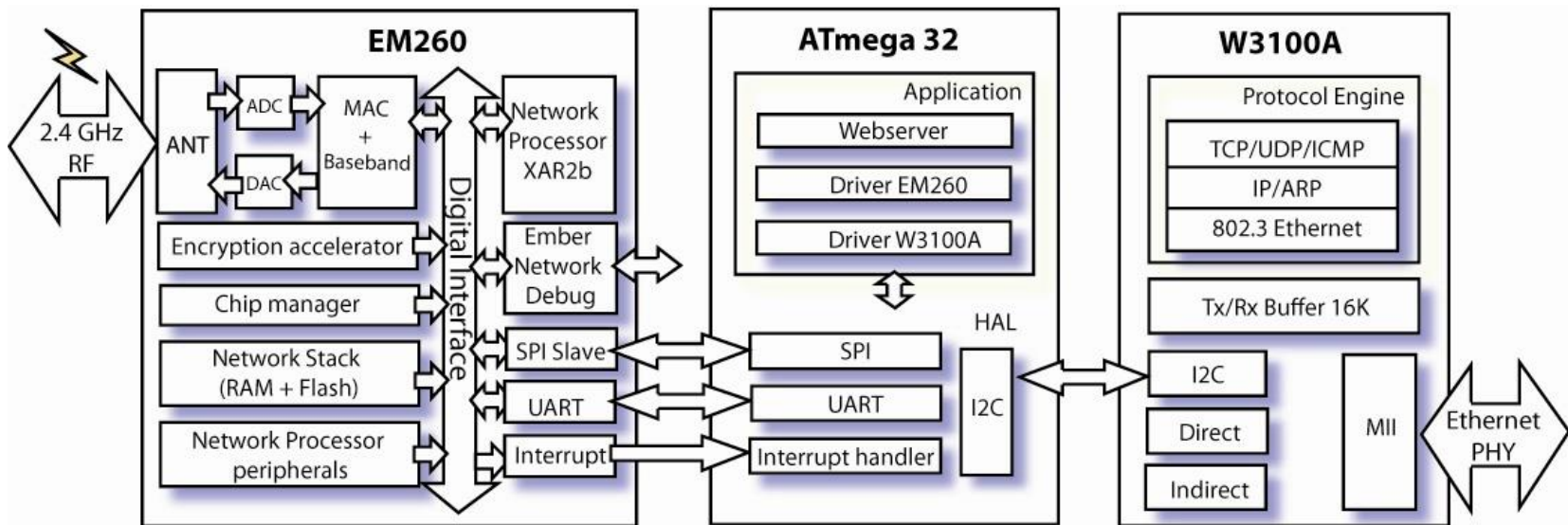
Gateway TCP/IP <->ZigBee: utility

- getting remote access via Ethernet for control ZigBee network from any work station with internet connection
- possibility to upgrade firmware by remote uploading via Ethernet + over-the-air
- increase the size of the ZigBee network by integrating several small networks



Gateway TCP/IP<->ZigBee: structure

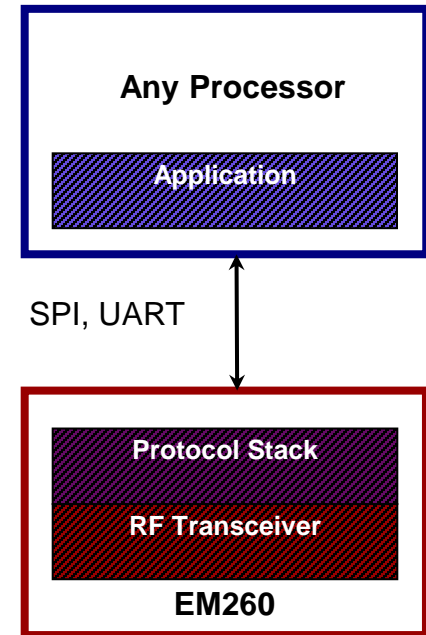
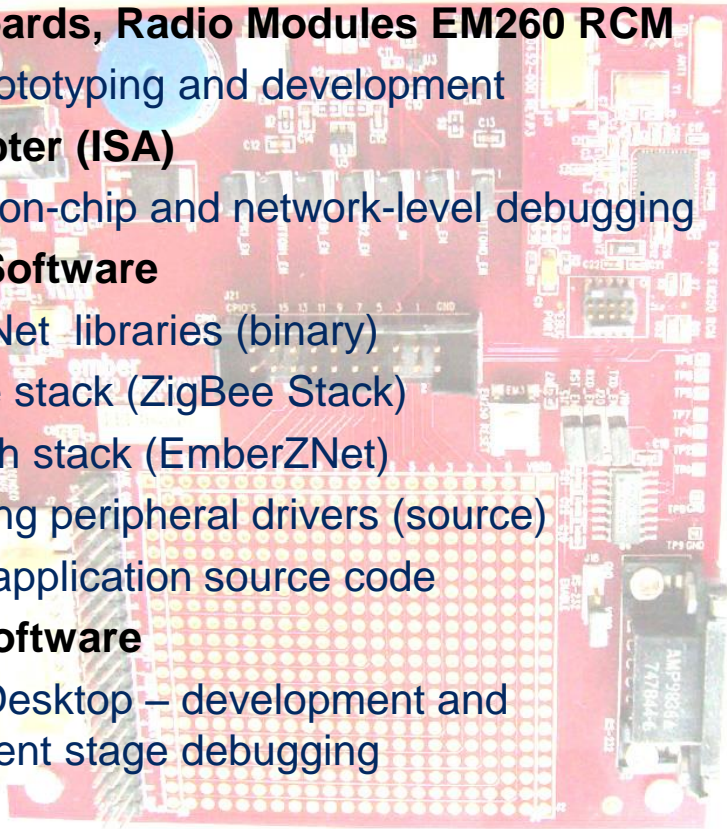
- ZigBee network processor
- Host controller for main application
- TCP/IP network processor



ZigBee network processor: EM260

EM260 Jump Start Kit

- **Breakout Boards, Radio Modules EM260 RCM**
 - Rapid prototyping and development
- **InSight Adapter (ISA)**
 - Enables on-chip and network-level debugging
- **Embedded Software**
 - EmberZNet libraries (binary)
 - Tree stack (ZigBee Stack)
 - Mesh stack (EmberZNet)
 - Supporting peripheral drivers (source)
 - Sample application source code
- **PC-Based Software**
 - InSight Desktop – development and deployment stage debugging

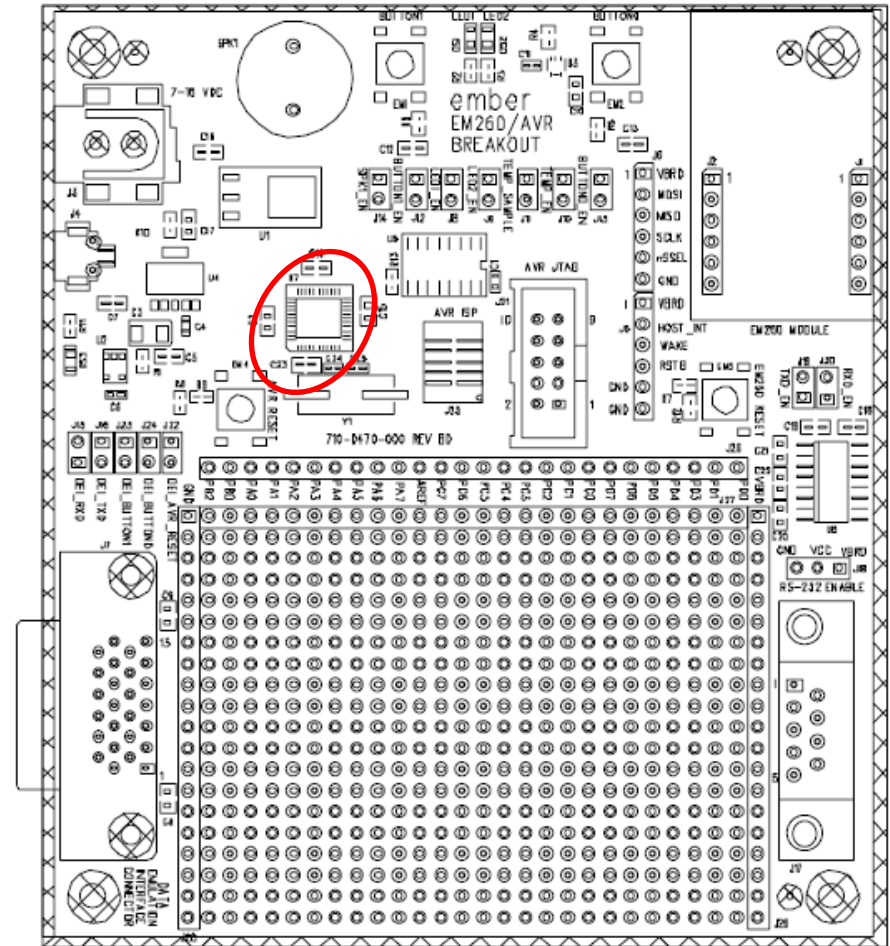


- Can be used with any MCU
- Adding ZigBee to any device is easier
- Self contained, and pre-programmed
- QFN 6mm X 6mm package



Host microcontroller: ATmega32

- 8-bit AVR microcontroller
- 16 MHz
- Low power
- High performance
- 32 K bytes Flash
- 1024 Bytes EEPROM
- 2 K byte Internal SRAM
- JTAG interface for debugging
- Hardware SPI



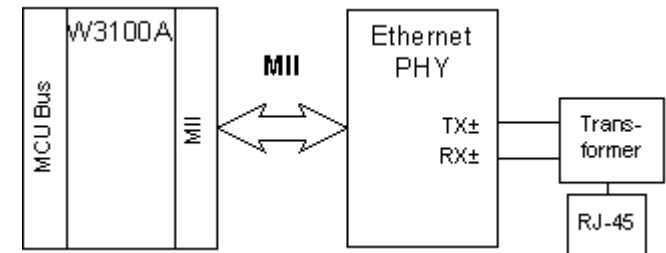
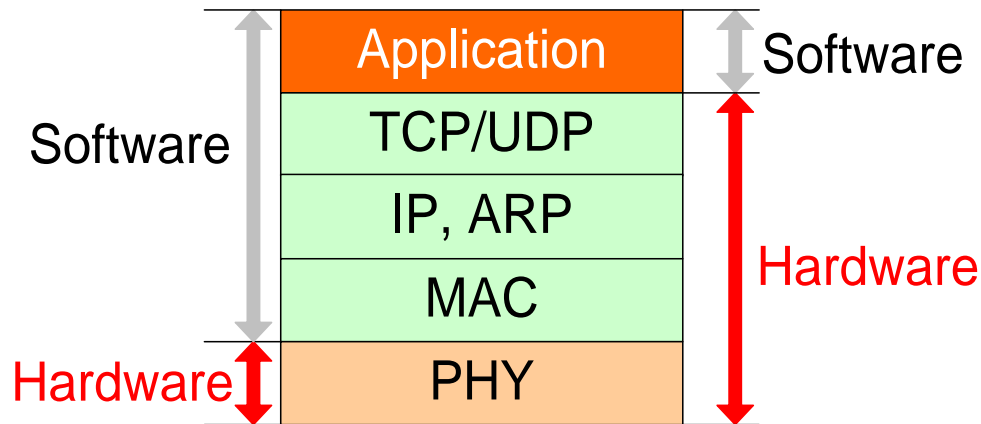
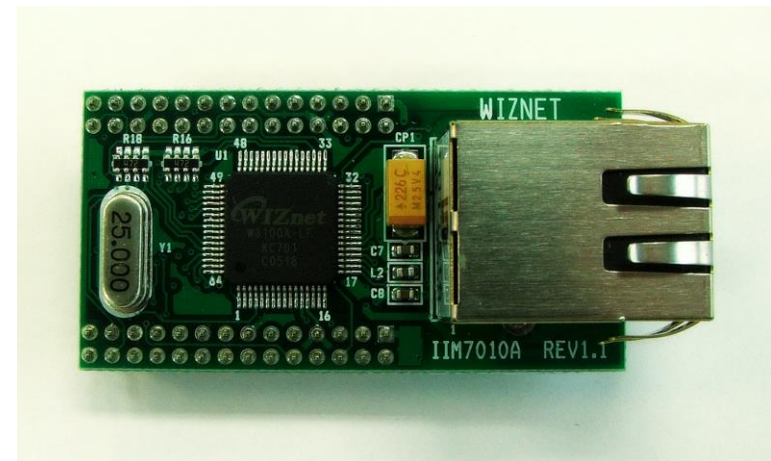
EM260 Breakout Board

TCP/IP network processor: Chip W3100A

- Software TCP/IP solutions: Ubicom, Lantronix, NetSilicon, Rabbit, BECK...
- Hardware TCP/IP solution: WIZnet



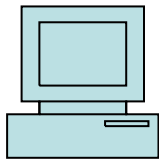
Module NM7010A



Implementation demands

- Configuration TCP/IP parameters
- Configuration ZigBee network options
- Transfer data through gateway

Realization:



Software for PC:
ConfigTool Utility



Firmware for MCU

Firmware for MCU



- Main application routine
- DHCP client
- UDP port for setting IP-options manually
- TCP server for configuration ZigBee network options and sending commands to ZigBee coordinator
- Transferring data between ZigBee and TCP/IP networks
- Driver for W3100A. I2C interface
- TCP server for transmitting data via TCP/IP network
- Driver for EM260. SPI interface
- RS-232 interface
- HAL level (timers, buttons, leds, buzzer, WDT, etc...)



Software for PC: ConfigTool Utility

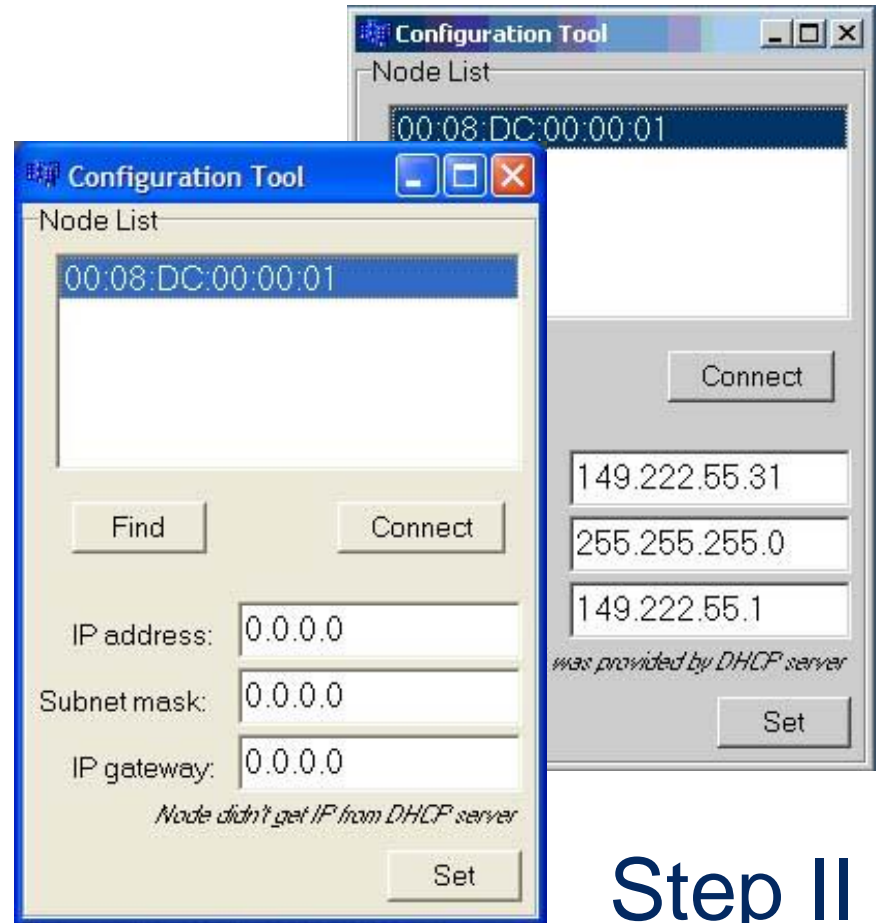


- Main user interface
- UDP port for configuration IP-options
- TCP-client for communication with gateway
- Configuration ZigBee network options
- Sending commands to ZigBee coordinator
- Getting of Coordinator's Binding Table

Utility ConfigTool: setting IP-options



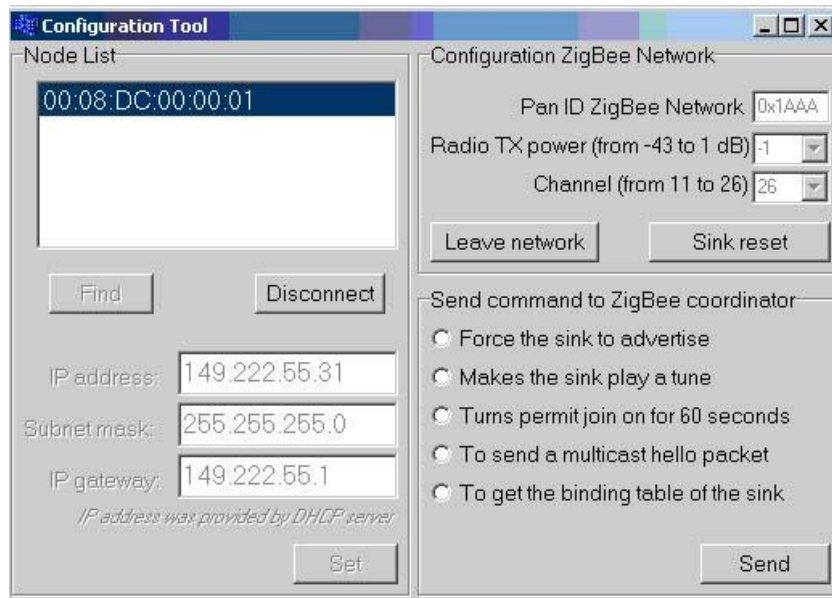
Step I



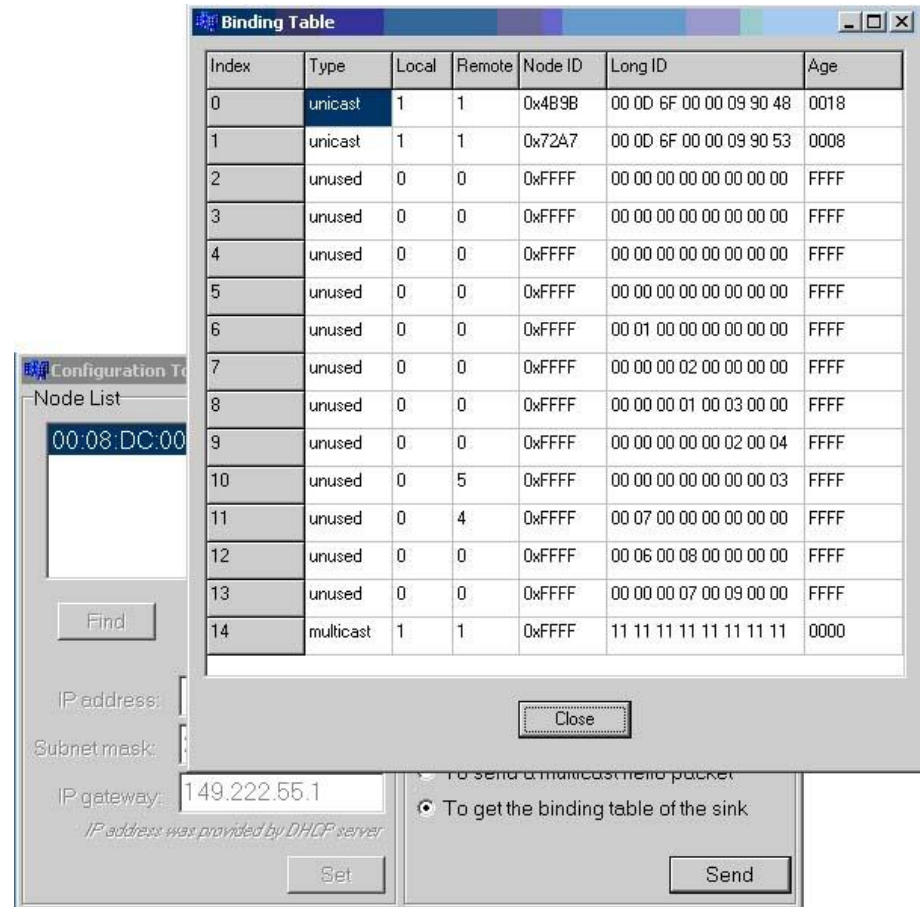
Step II



Utility ConfigTool: ZigBee configuration options

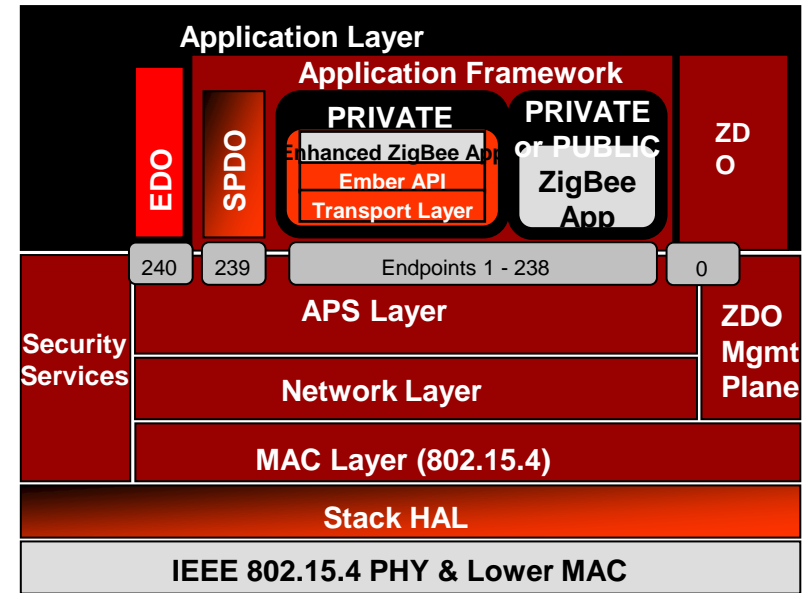


Step III



Transfer data

- Gateway is running in TCP-server mode
- For demonstration transfer data it is possible to use any standard TCP-client program (Telnet, Hyper Terminal)



5 types of messages are available: ■ Application ■ Ember ■ ZigBee

- APS layer ZigBee
- Transport layer EmberZNet
- unicast
 - multicast
 - broadcast
 - datagram
 - sequenced



Transfer data

- TCP-connection to gateway
- Serial connection with router

```

iã³EVENT: ezspUtilInit passed
INIT: sensor app 000D6F000099048
SENSOR APP: joining network - channel 0x1A, panid 0x1AAA
EVENT: stackStatus now EMBER_NETWORK_UP
SENSOR APP: network joined - channel 0x1A, panid 0x1AAA
EVENT: setting multicast binding, status is 0x00
RX [sink advertise] from: 000D6F000099076; processing message
EVENT waiting 05 ticks before reponding
EVENT: sensor set binding to sink [000D6F000099076]
TX [sink ack sensor], status:0x00
RX [sink ready] from: 000D6F000099076; processing message
RX [Unicast message] from: 000D6F000099076; processing message
Message: My unicast message
-

```

```

RX [sensor select sink] from: 000D6F000099048; processing message
GRX [sensor select sink] from: 000D6F000099053; processing message

```

index	type	local	remote	Node id	Long id	Age
0	unicast....	01	01	0x4B9B	00 0D 6F 00 00 09 90 48	000E
1	unicast....	01	01	0x72A7	00 0D 6F 00 00 09 90 53	0002
2	unused....	00	00	0xFFFF	00 00 00 00 00 00 00 00	FFFF
3	unused....	00	00	0xFFFF	00 00 00 00 00 00 00 00	FFFF
4	unused....	00	00	0xFFFF	00 00 00 00 00 00 00 00	FFFF
5	unused....	00	00	0xFFFF	00 00 00 00 00 00 00 00	FFFF
6	unused....	00	00	0xFFFF	00 01 00 00 00 00 00 00	FFFF
7	unused....	00	00	0xFFFF	00 00 00 02 00 00 00 00	FFFF
8	unused....	00	00	0xFFFF	00 00 00 01 00 03 00 00	FFFF
9	unused....	00	00	0xFFFF	00 00 00 00 00 02 00 04	FFFF
10	unused....	00	05	0xFFFF	00 00 00 00 00 00 00 03	FFFF
11	unused....	00	04	0xFFFF	00 07 00 00 00 00 00 00	FFFF
12	unused....	00	00	0xFFFF	00 06 00 08 00 00 00 00	FFFF
13	unused....	00	00	0xFFFF	00 00 00 07 00 09 00 00	FFFF
14	multicast..	01	01	0xFFFF	11 11 11 11 11 11 11 11	0000

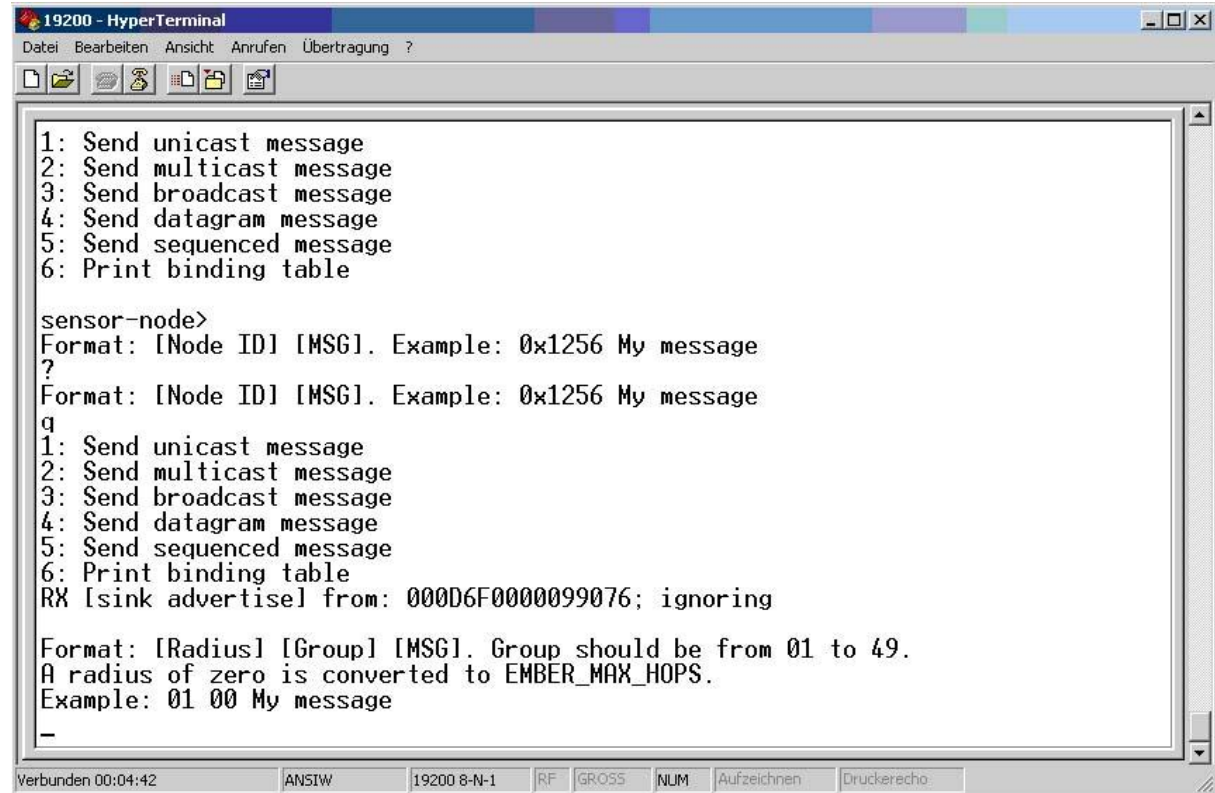
```

1
Format: [Node ID] [MSG]. Example: 0x1256 My message
0x4B9B My unicast message
Unicast message was successfully sent
-

```

Transfer data

- Help messages are available
- For getting help just button '?' should be pressed



```
19200 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
1: Send unicast message
2: Send multicast message
3: Send broadcast message
4: Send datagram message
5: Send sequenced message
6: Print binding table

sensor-node>
Format: [Node ID] [MSG]. Example: 0x1256 My message
?
Format: [Node ID] [MSG]. Example: 0x1256 My message
q
1: Send unicast message
2: Send multicast message
3: Send broadcast message
4: Send datagram message
5: Send sequenced message
6: Print binding table
RX [sink advertisel from: 000D6F0000099076; ignoring

Format: [Radius] [Group] [MSG]. Group should be from 01 to 49.
A radius of zero is converted to EMBER_MAX_HOPS.
Example: 01 00 My message
-

Verbunden 00:04:42 ANSTW 19200 8-N-1 RF GROSS NUM Aufzeichnen Druckerecho
```



Thank you for attention

