A (Short) Introduction to ZigBee

Christian Trödhandl

Embedded Computing Systems Group
Vienna University of Technology
Europe

April 19, 2007
Outline

1. Introduction

2. Protocol Overview
   - Node Types
   - Network Topology
   - Protocol Stack Architecture
   - Communication between Nodes

3. Atmel ZigBee Nodes
   - Hardware
   - Library

4. Example Application
Outline

1. **Introduction**

2. **Protocol Overview**
   - Node Types
   - Network Topology
   - Protocol Stack Architecture
   - Communication between Nodes

3. **Atmel ZigBee Nodes**
   - Hardware
   - Library

4. **Example Application**
The ZigBee standard for wireless sensor networks

- Application domain: Personal Area Network (PAN).
- Short range operation, low cost sensors, low power consumption.
- Topology: Star or Peer-to-Peer.
- Access control: Beacon or CSMA/CA.
- Reliable data transfer.
- Data rates: 250 kb/s (2450 MHz band), 40 kb/s (915 MHz), 20 kb/s (868 MHz).
Outline

1 Introduction

2 Protocol Overview
   - Node Types
   - Network Topology
   - Protocol Stack Architecture
   - Communication between Nodes

3 Atmel ZigBee Nodes
   - Hardware
   - Library

4 Example Application
Full Function Device (FFD): Can communicate with every type of device. A FFD can operate in three different modes:
Different Node Types in a ZigBee Network

- Full Function Device (FFD): Can communicate with every type of device. A FFD can operate in three different modes:
  - PAN Coordinator: Sends beacon frames, provides routing information, manages short, network-specific addresses

- Reduced Function Device (RFD): Can only talk to a single FFD.
Different Node Types in a ZigBee Network

- **Full Function Device (FFD):** Can communicate with every type of device. A FFD can operate in three different modes:
  - **PAN Coordinator:** Sends beacon frames, provides routing information, manages short, network-specific addresses.
  - **Coordinator:** Acts as router.
  - **Normal device.**
- **Reduced Function Device (RFD):** Can only talk to a single FFD.
Different Node Types in a ZigBee Network

- Full Function Device (FFD): Can communicate with every type of device. A FFD can operate in three different modes:
  - PAN Coordinator: Sends beacon frames, provides routing information, manages short, network-specific addresses
  - Coordinator: Acts as router.
  - Normal device.

- Reduced Function Device (RFD): Can only talk to a single FFD.
Different Node Types in a ZigBee Network

- **Full Function Device (FFD):** Can communicate with every type of device. A FFD can operate in three different modes:
  - PAN Coordinator: Sends beacon frames, provides routing information, manages short, network-specific addresses
  - Coordinator: Acts as router.
  - Normal device.

- **Reduced Function Device (RFD):** Can only talk to a single FFD.
In the star topology, the PAN coordinator chooses a unique (within its radio sphere of influence) PAN id. All attached nodes can only talk to the central PAN coordinator.
Within a peer-to-peer topology, each FFD can communicate with any other device within its range. A RFD may only communicate with a single FFD at a given time.
Multi-Cluster Network

Larger networks may be established by forming multi-cluster topologies. Each cluster has a single cluster head that is responsible for coordination within the cluster.
The IEEE 802.15.4-2003 standard describes the physical and MAC layer.
The IEEE 802.15.4-2003 standard describes the physical and MAC layer.

ZigBee builds on the IEEE standard and defines the network and application layer.
The IEEE 802.15.4-2003/ZigBee Protocol Stack

- The IEEE 802.15.4-2003 standard describes the physical and MAC layer.
- ZigBee builds on the IEEE standard and defines the network and application layer.

![Diagram of the ZigBee Protocol Stack]

- **Application**
- **ZigBee**
- **IEEE 802.15.4**

  - **Physical (PHY) Layer**: 2.4 GHz Radio, 868/916 MHz Radio
  - **Media Access Control (MAC) Layer**
  - **Network (NWK) Layer**
The IEEE 802.15.4-2003 standard describes the physical and MAC layer.

ZigBee builds on the IEEE standard and defines the network and application layer.
The IEEE 802.15.4-2003/ZigBee Protocol Stack

- The IEEE 802.15.4-2003 standard describes the physical and MAC layer.
- ZigBee builds on the IEEE standard and defines the network and application layer.
Physical (PHY) Layer

The physical layer is responsible for:

- Activation/Deactivation of transceiver.
- Channel selection, assessment.
- Transmission and reception of packets.
- Frequency bands: 2.4 GHz (worldwide), 868 MHz (EU), 916 MHz (US)
The following services are provided by the MAC layer:

- Beacon management
- Channel access
- Guaranteed Time Slot (GTS) management
- Frame validation
- Acknowledgment
- Association, disassociation
The lower level of the ZigBee protocol builds on the MAC layer of IEEE 802.15.4.  

- Topology specific routing  
- Security  
- New device configuration  
- Network startup
**Network (NWK) Layer (cont.)**

- Joining/leaving a network
- Addressing
- Neighbour discovery
- Route discovery
- Reception control
The application layer provides the following services:

- Maintain tables for binding
- Fragmentation, reassembly and reliable data transport
- Provide communication endpoints for the application
- Discovering devices and application services.
- Initiating/responding to binding requests between endpoints
Addressing

- Each ZigBee node has a unique 64 bit MAC address
- Additionally the Coordinator maintains a table to map the 64 bit addresses to network-specific 16 bit addresses
- Within each node, the application can define up to 240 Application endpoints.
The IEEE 802.15.4 standard describes the CSMA/CA mechanism to access the wireless channel:

- A device that wishes to transmit data frames waits for a random backoff.
- If the channel is clear after the backoff, the data is transmitted.
- If the channel is busy, the device waits for another random period.
- (Optional) Acknowledgment frames are sent immediately after the corresponding data frames without using the CSMA/CA mechanism.
Beacon Mode

The network may also utilize the so-called beacon mode:

- A round (superframe) is divided into 16 equally sized slots.
- Coordinator regularly sends beacon frames in the first slot.
- The beacon frames are used to synchronize the attached devices, identifies the PAN, and describes the superframe structure.
- Any device that wishes to send data uses the CSMA/CA mechanism, but aligns the sent frames to the slots.
- The PAN coordinator may assign guaranteed time slots (GTS) to devices for low-latency or fixed data bandwidth.
- Up to 7 GTS can be allocated in this way at the end of the superframe.
Frame Types

The IEEE standard defines four different frame types:

- A beacon frame: Sent by the coordinator to announce the network and contains the superframe structure.
- A data frame: Used for data transfer
- An acknowledgment frame: To confirm the successful reception of a frame.
- A MAC command frame: For handling MAC peer entity control transfers.
Service Primitives

Defines the communication between different layers of the protocol:

- **Request**: Passed from user to the underlying layer to initiate a service.
- **Indication**: To indicate an internal event that is significant to the user.
- **Response**: To complete a procedure invoked by an Indication primitive.
- **Confirm**: Passed to the user application to convey the results of a previous service request.
Service Primitives (cont.)

Initiating Device (Appl, NWK) → Request → Confirm → Lower layer (MAC, PHY) → Indication → Response → Responding Device (Appl, NWK)
Outline

1. Introduction

2. Protocol Overview
   - Node Types
   - Network Topology
   - Protocol Stack Architecture
   - Communication between Nodes

3. Atmel ZigBee Nodes
   - Hardware
   - Library

4. Example Application
The Atmel ATAVRRZ200 demonstration kit contains the following components:

- Five ZigBee nodes, each equipped with an ATmega1281 8 bit AVR microcontroller and an AT86FR230 ratio controller.
- Three buttons and three LEDs on each board.
- Additional LCD board.
- Power supply over battery.
- USB programming adapter.
ATAVRRZ200 Demonstration Kit (cont.)
Microcontroller communicates with transceiver over SPI interface.

Additionally, an interrupt and a timer clock is supplied to the microcontroller.
Atmel supplies a programming library for its transceiver modules:

- The library is called libl2_rdk230_rel.a and is available for the avr-gcc and iar C compiler.
- Only IEEE 802.15.4-2003 support.
- ZigBee layer has to be implemented within the application (no endpoint-to-endpoint support).
- Library uses timer 1 with ICP.
- Library calls (for request and response) are prefixed with wpan_.
- The application may define certain callback functions that are invoked by the library (for confirm and indication events). This callback functions are prefixed with usr_.

Atmel MAC Library for AT86RF230 transmitters
Outline

1. Introduction

2. Protocol Overview
   - Node Types
   - Network Topology
   - Protocol Stack Architecture
   - Communication between Nodes

3. Atmel ZigBee Nodes
   - Hardware
   - Library

4. Example Application
The following application code shows how to use the Atmel MAC library for implementing a device application. The application starts the MAC stack and associates with the coordinator node. (Code snippets taken from the demos supplied with the Atmel MAC library, see disclaimer)
Including Header Files

/* Include interrupt.h for the sei() macro */
#include <avr/interrupt.h>

/* defines for demo applications */
#include "wpan_defines.h"

/* ieee_const.h holds IEEE 802.15.4 constants and attribute definitions */
#include "ieee_const.h"

/* wpan_mac.h includes function definitions for all library functions */
#include "wpan_mac.h"
Define a variable to hold our status

/* define status variable */
typedef struct
{
    uint16_t device_short_address;
    uint8_t coord_address_mode;
    uint64_t coord_address;
    uint16_t pan_id;
    uint8_t logical_channel;
    uint8_t msdu_handle;
    device_state_t state;
} device_status_t;

static device_status_t d_status;
Function to initialize the application

```c
/* define status variable */
static void application_init(void) {
  /* reset global application status variable */
  memset(&d_status, 0, sizeof(d_status));
  /* init IO ports .... */
  /* init mac layer */
  wpan_init();

  SET_STATE(INIT_DONE);
  /* enable interrupts */
  sei();
  return;
}
```
Define main user task

```c
static void switch_task(void)
{
/* do something ... */

if (send_data)
{
    /* send data */
    wpan_mcpsdata_addr_t ai;
    ai.SrcAddrMode = WPAN_ADDRMODE_SHORT;
    ai.SrcPANId = d_status.pan_id;
    ai.SrcAddr = d_status.device_short_address;
    ai.DstAddrMode =
        d_status.coord_address_mode;
}
```
Define main user task (cont.)

```c
ai.DstPANId = d_status.pan_id;
ai.DstAddr = d_status.coord_address;
wpan_mcps_data_request(&ai,
    d_status.msdu_handle++,
    WPAN_TXOPT_ACK,
    (void *) &d_status.led,
    sizeof(uint8_t));
}
return ;
}```
Define a callback function for received data

```c
void usr_mcps_data_ind(
    wpan_mcpsdata_addr_t *pAddrInfo,
    uint8_t mpduLinkQuality,
    uint8_t SecurityUse,
    uint8_t ACLEntry, uint8_t msduLength,
    uint8_t *msdu)
{
    if ((d_status.state == RUN) &&
        (pAddrInfo->DstPANId == d_status.pan_id))
    {
        /* do something with received data */
    }
}
```
Main function

```c
int main(void)
{
    application_init();
    /* send reset, scan request, ... */
    mac_do_reset();
    while(1)
    {
        while(wpan_task())
        {
            /* short running tasks */
        }
        switch_task(); /* main user task */
    }
}
```
Some References

**IEEE.**
IEEE std 802.15.4-2003: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs).
Available at http://www.ieee802.org.

**ZigBee Alliance.**
ZigBee specification.
Specification, ZigBee Alliance, December 2006.
Available at http://www.zigbee.org.

**Atmel.**
*IEEE 802.15.4 MAC User Guide.*
2325 Orchard Parkway, San Jose, California, September 2006.

**Atmel.**
*ATAVRRZ200 Demonstartion Kit AT86RF230 (2450 MHz band) Radio Transceiver User Guide.*
2325 Orchard Parkway, San Jose, California, July 2006.
Thank you for your attention!
Questions?
Copyright (C) 2006, Atmel Corporation. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of ATMEL may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.