HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Electronics, Communications and Automation

Antti Karjalainen

# WIRELESS SENSOR NETWORK PLATFORM FOR ASSISTIVE AUTOMATION APPLICATIONS

Thesis submitted for examination for the degree of Master of Science in Technology

Espoo 4.12.2009

Thesis supervisor:

Prof. Aarne Halme

Thesis instructor:

MSc Panu Harmo

TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Antti Karjalainen

Työn nimi: Langaton anturiverkkoalusta avustavan kotiautomaation sovelluksiin

Päivämäärä: 4.12.2009 Kieli: Englanti Sivumäärä: 7+85

Tiedekunta: Elektroniikan, tietoliikenteen ja automaation tiedekunta

Professuuri: Automaatiotekniikka Koodi: AS-84

Valvoja: Prof. Aarne Halme

Ohjaaja: DI Panu Harmo

Ikääntyvien ihmisten avustavan automaation järjestelmät voivat hyötyä langattomien anturiverkkojen käytöstä. Sovelluksista voi tulla halvempia, helpommin asennettavia ja käyttöympäristöönsä helpommin sulautuvia langattomien anturiverkkojen käytön avulla. Uudenlaisia sovelluksia voidaan toteuttaa, jotka eivät olisi ilman langattomia anturiverkkoja mahdollisia.

Tässä diplomityössä esitetään 8-bittisillä mikrokontrollereilla toteutetun edullisen ja vähän virtaa kuluttavan langattoman anturiverkkoalusta kehitysprosessi, alkaen alustavista vaatimusmäärittelyistä. Laitteiston ja ohjelmiston kehitystyön lisäksi käsitellään sovelluskehitystä. Lisäksi esitetään käytännöllinen avustavan automaation järjestelmä, Ovivalpas, joka hyödyntää tässä työssä kehitettyä langatonta anturiverkkoalustaa.

Tulokset osoittavat, että nykypäivänä saatavilla oleva langaton anturiverkko-, mikrokontrolleri- ja anturitekniikka ovat tarpeeksi kehittyneitä, jotta niitä voidaan hyödyntää todellisissa avustavan automaation sovelluksissa. Tämän diplomityön osana kehitetty langaton anturiverkkoalusta toimii luotettavasti ja se tarjoaa avustavan automaation prototyyppisovellusten kehittämisessä vaadittavat toiminnallisuudet, sekä laitteistoa voidaan käyttää useita vuosia ilman paristojen vaihtoa. Kehitettäessä avustavan automaation sovelluksia, joiden tulee olla hyvin joustavasti mukautettavissa loppukäyttäjien vaatimusten mukaan, on suositeltavaa, että sovellukset kehitetään käyttäen kotipalvelininfrastruktuuria langattomien anturiverkkojen lisäksi.

Avainsanat: langattomat anturiverkot, avustava automaatio, Zigbee, vähävirtainen radio, mikrokontrolleri, sulautetut järjestelmät, jokapaikan tietotekniikka

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author: Antti Karjalainen

Title: Wireless Sensor Network Platform for Assistive Automation Applications

Date: 4.12.2009 Language: English Number of pages: 7+85

Faculty: Faculty of Electronics, Communications and Automation

Professorship: Automation technology Code: AS-84

Supervisor: Prof. Aarne Halme

Instructor: MSc Panu Harmo

Assistive automation applications for aging people can benefit from the use of wireless sensor network technologies. Applications can become cheaper, easier to install, and more ambient with use of wireless sensor networks. New types of applications can be conceived that are not feasible without the use of wireless sensor networks.

This thesis presents the development process of a inexpensive and low-power wireless sensor network platform using 8-bit microcontrollers starting from initial requirement specifications. Hardware and software designs are presented along with application development discussion. A practical assistive automation application, door monitor, that is implemented by using the developed wireless sensor network platform is also presented.

Results indicate that wireless sensor network technologies, as well as microcontroller and sensor technologies that are available today, are mature enough for the use of real-life assistive automation applications. The wireless sensor network platform, that was developed as a part of this work, operates reliably and provides necessary features for assistive automation application prototyping, while at the same time is able to operate multiple years without having to have batteries replaces. However, when designing complicated applications that need to be highly customizable according to the end-user's needs, use of additional home server infrastructure is recommended in addition to the wireless sensor network.

Keywords: wireless sensor networks, assistive automation, Zigbee, low-power radio, microcontroller, embedded systems, ubiquitous computing

## Acknowledgments

This thesis was done in the Department of Automation and Systems Technology of Helsinki University of Technology, Finland as a part of the Devices and Interoperability Ecosystem (DIEM) project of the ICT SHOK program.

The subject of this thesis is one that I personally find greatly fascinating and I feel that I have had an excellent opportunity of developing my competence in the respective field of technology. I wish to thank my instructor Panu Harmo for giving me such a great subject and for supporting me in throughout the whole project. I also wish to thank Professor Aarne Halme for supervising this thesis.

This thesis required a great deal of hands-on work with electronics. I am very grateful for all my colleagues at the department who helped me with my hardware designs and gave me great insight on design considerations.

On a personal level I wish to thank my parents and my sister for their support throughout my whole studies. Of course, special thanks goes to my lovely fiancee for constantly being there for me and believing in me. I love you.

Otaniemi, 4.12.2009

Antti Karjalainen

# Contents

# Abbreviations

| | |
|---|---|
| API | Application programming interface. |
| APS | Application support sub-layer. |
| FFD | Full-function device. |
| GCC | GNU compiler collection. |
| GPIO | General purpose input/output. |
| HVAC | Heating, ventilating, and air conditioning. |
| IDE | Integrated development environment. |
| ISP | In-system programming. |
| LR-WPAN | Low-rate wireless personal area network. |
| LQI | Link quality indication. |
| MAC | Media access control. |
| MCU | Microcontroller unit. |
| NWK | Network layer. |
| OTA | Over-the-air. |
| PAN | Personal Area Network. |
| PCB | Printed circuit board. |
| PCINT | Pin change interrupt. |
| PHY | Physical layer of the OSI model. |
| PIR | Pyroelectric infrared. A PIR sensor element is often used in the construction of passive infrared motion detectors. |
| RFD | Reduced-function device. |
| RISC | Reduced instruction set computer. |
| RPC | Remote procedure call. |
| RSSI | Received signal strength indication. |
| SoC | System-on-a-chip. A single integrated chip containing multiple functionalities of an embedded system. |
| SPI | Serial peripheral interface bus. |
| USART | Universal asynchronous receiver/transmitter. |
| WPAN | Wireless personal area network. |
| WSN | Wireless sensor network. |
| ZC | Zigbee coordinator. |
| ZCL | Zigbee cluster library. |
| ZDO | Zigbee device object. |
| ZDP | Zigbee device profile. |
| ZED | Zigbee end device. |
| ZR | Zigbee router. |

# 1 Introduction

Assistive automation systems have potential to improve the quality of life of the aging population. These systems are intended to promote aging in place, independence, and safety of the aging population. Research shows that elderly people tend to approve of systems that enable them to live longer at home independently, as long as there is no high cost associated with the technology.

Assistive automation systems are usually retrofitted into private homes which means that they have to be able to adapt to varying environments and be as unobtrusive as possible. Also, since these systems reach for mass markets, cost efficiency and low maintenance are very important. Recent advancements in wireless sensor network, microcontroller, and sensor technologies have enabled the creation of systems that will fulfill these requirements. In addition, wireless sensor network technologies will enable information sharing between applications and interoperability of devices from multiple vendors, but also enable new types of applications that would not be possible to implement without wireless sensor networks.

This master's thesis describes the design process of a wireless sensor network platform for assistive automation applications. The process starts out with the definition of initial requirements, that are based on the needs of assistive automation applications. The selection of the wireless sensor network standard is based on these requirements. The selected standard then forms the basis for the hardware design. After the hardware is developed and built, a software framework for using the hardware is designed. This software framework is then used to develop a practical application that demonstrates the capabilities of the whole platform.

The structure of this thesis is as follows. This first section forms a brief introduction to this thesis. In Section 2, the concept of assistive automation is presented. In Section 3, the concept of wireless sensor networks is presented along with applications of wireless sensor networks, standards and technologies for implementing wireless sensor network systems, and introduction to two key standards in the field of wireless sensor networks. Section 4 presents recent related work, regarding wireless sensor networks in assistive automation.

Section 5 describes the design and implementation process of the developed wireless sensor network platform. This section also describes test arrangements and results for measuring the platform's power consumption and wireless range. In Section 6, application development related topics are discussed. This section also presents the demonstration application that was implemented as a part of this work.

Section 7 presents the final conclusions, evaluation of the results, and discusses future development need in applying wireless sensor networks in assistive automation applications. Appendix A contains layouts and schematics for every hardware component that was developed as a part of this work.

## 2 Assistive Automation

The term assistive automation describes best the type of applications that are discussed in this thesis. However, assistive automation is a concept that is not well defined in literature. In this section, the definition of assistive automation is discussed.

The following phrases describe assistive automation: Assistive automation:

- is aimed to enable aging in place
- promotes security
- systems operate autonomously or with little intervention
- is embedded, ambient, and ubiquitous
- uses sensors to measure physical environment
- uses actuators to manipulate physical environment
- systems generate statistical reports from physical events
- systems generate alarms based on physical conditions
- integrates various assistive devices, sub-systems, and services

An assistive automation application can be, e.g., a stove alert system that can automatically turn off the stove if certain conditions are met. A simple mechanical system that will turn off the stove according to a timer would not be characterized as an assistive automation system. The stove alert system could first sound an alert, if no one has been moving around the apartment or in the kitchen for a long time. After sounding the alarm, the stove alert system could turn off the stove if there is nothing cooking on the stove or the smoke detector is starting to detect something. Then this system could be considered as an assistive automation application.

Figure 1 illustrates concepts that are related to assistive automation. In this figure, concepts that overlap with each other are related. Assistive automation can be elaborated by defining concepts that are close to it and relations between these concepts.

Assistive technology is defined The Engineering Handbook of Smart Technology for Aging, Disability, and Independence [1] as follows: "Assistive technology encompasses all systems that are designed for people with special needs, and that attempt to compensate the handicapped." And "Assistive technology also include systems

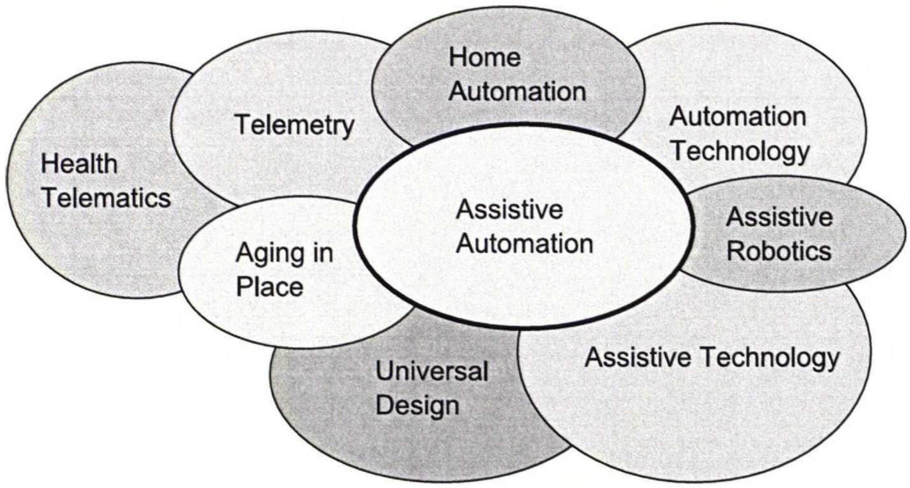


Figure 1: Assistive Automation

that restore personal functionality, such as external prostheses and ortheses." General automation technology, on other hand, encompasses topics like sensors, programmable logic, servers, actuators, user interfaces, and robotics. When combined, robotics and assistive technology can form the concept of assistive robotics. Assistive robotics can include, e.g., cleaning robots and robotic wheelchairs. Assistive robotics can be perceived as part of assistive automation in the sense that both technologies utilize sensory information from their surroundings to generate functionality that help people with special needs.

Home automation has common aspects with telemetry, assistive automation, and automation technology. Telemetry consists of remote measuring and reporting of information. These types of features are common in home automation applications, that let the house owners to see the status of their house remotely. In assistive automation applications, remote monitoring and reporting are also essential features. An assistive automation system could monitor the movements of an elderly person and use this information to generate alarms, e.g., if the person has not visited the kitchen for a whole day. Also, lighting automation systems can be perceived to be either a home automation system or an assistive automation system, depending on the purpose of its use.

According to universal design philosophy, buildings, products, and environments should be designed so that they are accessible and usable for everyone. Universal design guidelines should be taken into account when designing assistive automation applications. These include, e.g., tolerance for error, perceptible information, and simplicity of use. Aging in place is a concept that mens the ability of a person to live in one's own home for as long as reasonably possible. Assistive automation applications are intended to promote aging in place.

The concept of health telematics covers systems that enable the reporting and monitoring of health related information. This concept is not related to assistive automation in the sense that health telematics systems are generally intended to provide medical data for health care needs. A blood pressure monitor that can automatically send its data back to a hospital would be considered to be a health telematics system. In this case the patient would measure his/hers blood pressure regularly over a certain time period, after which his/hers doctor could see if there is need for any procedures. In health telematics, data is not analyzed automatically and the applications are not ambient in nature.

# 3 Wireless Sensor Networks

This section will introduce the concept of a wireless sensor network (WSN) and main application areas for WSNs. An overview of standards and technologies that are used in wireless sensor networks is presented. Two standards, IEEE 802.15.4 and Zigbee, that are considered to be the most important in practical wireless sensor networks are described in more detail.

In "Wireless Sensor Networks" by Sohraby et al. [2], a sensor network is defined as a group formed by a number of devices that have sensing, computing and communication capabilities. These devices enable instrumentation, observation and the ability to react to events and phenomena in a specified environment. With added wireless communication ability, this group forms a wireless sensor network. In "Wireless Sensor and Actuator Networks" [3] Verdone et al. adds the notion of actuators and defines the concept of wireless sensor and actuator network (WSAN).

A generic device in a wireless sensor network is called a node. A wireless sensor network is usually formed by a number of different types of nodes which can be either sensors, sinks, or gateways. A WSN comprises of a varying number of each of these devices. A typical sensor node is composed of a microprocessor for managing tasks, sensor elements for acquiring measurements, memory for storing temporary data, a radio transceiver with an antenna for communicating with the rest of the network. [3]

WSNs can be classified by their ability to relay messages as either single-hop or multi-hop networks, i.e., is it possible for the networks to route messages over longer distances than a single node's wireless transmission range [3]. Also features like the network's ability to self-organize and self-configure itself and the capability to dynamically route packages can be considered as a differentiating factor among WSNs [2].

Figure 2 illustrates a generic wireless sensor network with multi-hop routing. Sensor nodes acquire measurements and the measurement data is collected by sink nodes for further routing. The network's gateway node is connected to some other network, e.g. the Internet, and it acts as a data aggregation point for the whole network.

Nodes in a WSN can differ by computing capabilities and power source availability. Usually a node with large computational resources and plentiful power supply, i.e. mains power, act as a data aggregation point or a sink in the network. Battery operated computationally light weight sensor nodes feed sensing information into sink nodes. In terms of power supply, battery operated nodes form a bottle neck

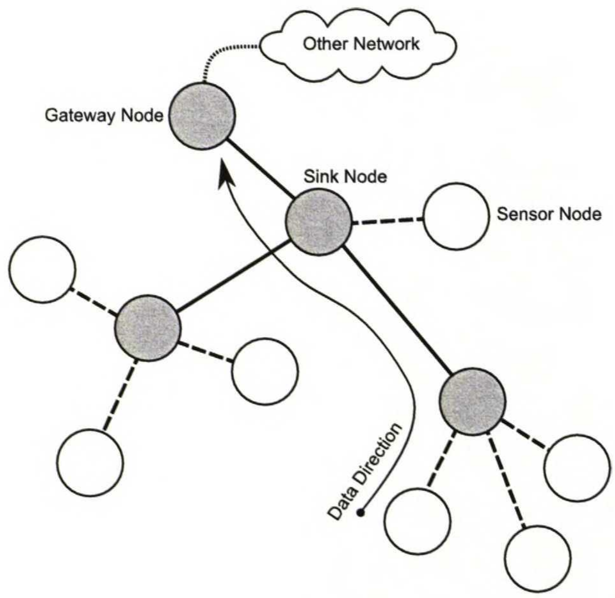


Figure 2: A Generic Wireless Sensor Network

in the network. This means that the time a WSN can operate without external intervention is defined by the energy efficiency of its battery operated devices and initial capacity of the batteries that are used to power battery operated devices. [3] [2]

Since most WSNs are designed to operate over a time period measured in years rather than months or weeks, the energy efficiency of battery operated sensor nodes is one of the most crucial aspects of WSN protocol and hardware development [2]. Power efficiency is being addressed by low-duty-cycle operation, low data volumes, and reduced energy budget of wireless transmissions, which is achieved with multi-hop routing [3] [2].

Currently many aspects of wireless sensor networks, such as security and localization, are under research, but commonly used standards are already starting to emerge. Many low data rate applications use physical (PHY) and media access control (MAC) layers that are defined by the IEEE 802.15.4 standard (described in Section 3.3) as the lower levels of their network stack. One of the main benefits of IEEE 802.15.4 is that it is low-cost and low-complexity but allows creation of multiple different topologies (star, mesh, tree, cluster-tree). IEEE 802.15.4 is also an open standard with good hardware availability, and it is flexible enough to fit well in different applications with varying requirements. [3]

Within the context of this thesis, unless otherwise stated, a WSN is seen as a low power wireless multi-hop mesh infrastructure that is capable of both sensing and acting on its environment. Also dynamic routing capabilities and the ability to self-organize and self-configure are assumed.

## 3.1 Applications

Application areas of wireless sensor networks are abundant. Both Sohraby et al. [2] and Verdone et al. [3] list a comprehensive array of foreseeable usages for WSNs. The definition of the term WSN given in Section 3 applies to this chapter, but it should be noted that some of the applications listed here could be easily implemented using less complicated wireless sensor network technologies. A brief description of the main application areas with a few examples from each category is presented below. The list is adapted, when not otherwise specified, from the works of Sohraby et al. [2] and Verdone et al. [3].

- Energy management
  - Automatic meter reading
- Military applications
  - Condition based monitoring
  - Military surveillance
  - Borders monitoring
- Medical and health care applications
  - Home monitoring of recovering patients
  - Body worn medical sensors
- Building automation
  - Lighting control
  - Centralized HVAC management
  - Localization of movable objects
- Home automation
  - Flexible interfaces to home controls
  - Retrofitted home controls (lighting, HVAC, etc.)
  - Utility usage data collection
- Environmental monitoring

  - Forest fire detection
  - Flood detection
  - Structural integrity monitoring
- Logistics
  - Assets management
  - Warehouse tracking
- Security
  - Intrusion detection
- Industrial applications
  - Equipment management and preventive maintenance
  - Assembly line and work flow inventory
  - Easily accessible field service link to sensors and equipment
  - Remote monitoring of assets, billing, and energy management
- Consumer electronics
  - Radio frequency remote controls for consumer electronics applications [4]
- Agriculture
  - Cattle tracking [5]

In addition to applications listed above, more specific listing of applications of WSNs in assistive automation is presented in Section 4.

## 3.2 Standards and Technologies

As it is with WSN applications, technologies for implementing wireless sensor networks are plentiful as well. Sohraby et al. [2] introduces Zigbee, Bluetooth, Wibree, and Ultrawide bandwidth (UWB) as prominent WSN technologies. A more recent source, "Zigbee Wireless Networking" by Gislason [6], adds to the list Wireless USB, WiFi, IEEE 802.15.4 MAC, Z-wave, SimpliciTI, Synkro, MiWi, TinyOS, and 6LoWPAN. These technologies are briefly described here, except for IEEE 802.15.4 and Zigbee which are described in more detail in Section 3.3 and Section 3.4.

From the above mentioned WSN technologies, the following fall under the same category: IEEE 802.15.4 MAC, SimpliciTI, Synkro, MiWi, TinyOS, 6LoWPAN and Zigbee. As seen in Table 1, the common denominator with these protocols is that they all use IEEE 802.15.4 standard as their basis and implement upper level protocols on top of that [6]. UWB, Wireless USB, WiFi, Bluetooth, and Wibree do not fit strictly under the definition of WSN, that is used within the context of this thesis, since they are not either multi-hop, low power, or mesh networks.

| IEEE 802.15.4 Based | Non-IEEE 802.15.4 Based |
|---|---|
| Zigbee | Bluetoot |
| IEEE 802.15.4 MAC | Wibree |
| SimpiciTI | UWB |
| Synkro | Wireless USB |
| MiWi | WiFi |
| TinyOS | Z-Wave |
| 6LoWPAN | |

Table 1: WSN Technologies

Although ultrawide bandwidth is mentioned along with WSN technologies, UWB itself is not a protocol for data transfer but rather a technology that can enable interesting WSN applications. The definition of UWB signal is a signal which exceeds 500MHz in bandwidth or occupies more than 20% of the center frequency. The main advantage of UWB in WSN usage is that an UWB signal will penetrate through obstacles, it allows for high data rates, and it is usable in high precision localization systems [2]. Wireless USB is an inexpensive technology, mainly targeted for PC peripheral, that utilizes UWB as its basis. Although the sensor and control space is not its main focus, Wireless USB could be adapted to that purpose as well [6].

WiFi can be used to form mesh networks. However, there are not a common standard available for mesh networking with WiFi. Despite this, WiFi is a common and widely used technology which could be adapted for WSN use. The main disadvantages of WiFi are relatively high price, excessive power consumption, and high complexity. [6]

Bluetooth is another well-understood and widely used wireless technology. It has good security functionalities but, like WiFi, it is not designed for WSN usage and it suffers from too high power consumption and inability to scale up to systems with thousands of nodes. Wibree is a new technology, similar to Bluetooth, that is designed to co-exist with Bluetooth in similar applications. The key difference with Wibree is that it is promised to have extremely low power consumption. Like Bluetooth, Wibree is designed for personal area networks (PAN) and it is not capable of forming mesh networks or route messages in multiple hops, which means that Wibree is not a very scalable technology either. [6]

Z-Wave is a sub-1GHz proprietary WSN technology by Zensys of Sigma Designs corporation. It is targeted specifically to home automation applications. Z-Wave is not a standard or an open technology, but it has some significant adopters in the home automation industry [6]. Z-Wave is quite similar to Zigbee (see Section 3.4), in the sense that both Zigbee and Z-Wave are designed for WSN usage and support mesh networking and multi-hop routing as well as low power sensor nodes [7].

IEEE 802.15.4 MAC can be used to form simple point-to-point or star networks. However, there are no application level interoperability mechanisms available in the MAC-only implementations. The host of different IEEE 802.15.4 based WSN technologies address this by implementing upper network layers on top of PHY and MAC layers that are defined in IEEE 802.15.4. [6]

SimpliciTI is Texas Instrument corporation's (TI) simple repeating protocol intended for use with TI's 802.15.4 radios. Similarly, MiWi by MicroChip corporation is a peer-to-peer mesh protocol that uses IEEE 802.15.4 as its lower layers. MiWi is intended for use with MicroChip's radios and solutions. [6]

TinyOS is an open-source embedded operating system, designed for wireless mesh networking. It uses a C-like programming language called nesC [8]. TinyOS has been adapted for various different WSN projects including IEEE 802.15.4 based Zigbee implementation by MeshNetics corporation [6]. Last, there is IETF standard 6LoWPAN, which stands for IPv6 over low power wireless area networks. 6LoWPAN defines a header compression mechanism which allows IPv6 packets to be transfered over IEEE 802.15.4 based networks [9].

There are many more protocols and technologies available for wireless sensor networks that are not covered here. This chapter gives only a superficial description of a few examples for the sake of general insight. It should be noted that a gross categorization can be made between IEEE 802.15.4 and non-IEEE 802.15.4 technologies. The following chapter will present more insight into IEEE 802.15.4.

## 3.3 IEEE 802.15.4

In this section, key elements of IEEE standard 802.15.4-2006 are reviewed. IEEE standard 802.15.4-2003 [11] was the first standard regarding the topic of wireless medium access control and physical layer specifications for low-rate wireless personal area networks. IEEE standard 802.15.4-2006 [10] is a revision of this standard. It is backward compatible, removes ambiguities, and extends the market applicability of IEEE 802.15.4-2003. Within the context of this work, IEEE standard 802.15.4-2006 is referred to simply as IEEE 802.15.4.

According to IEEE 802.15.4-2006 specification [10], IEEE 802.15.4 is a standard that defines a protocol and compatible interconnection for data communication devices that use low data rate, low power, low complexity, and short-range radio frequency transmissions in a wireless personal area network (WPAN). IEEE 802.15.4 defines both the physical layer (PHY) and medium access control (MAC) sublayer for low-rate wireless personal area networks (LR-WPAN). Thus, IEEE 802.15.4 defines a standard foundation for LR-WPANs [10].

In IEEE 802.15.4-2006 specification [10], LR-WPAN is defined more specifically as quoted: "A simple, low-cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements." A LR-WPAN is a network that is intended to be easy to install, that has reliable data transfer in short range operation, is extremely low cost, and has reasonable battery life while maintaining a simple and flexible protocol. Additional characteristics of a LR-WPAN are star or peer-to-peer operation, optional guaranteed time slots, carrier sense multiple access with collision avoidance (CSMA-CA) channel access, energy detection, link quality indication (LQI), and operation at 16 channels in 2450MHz band, 30 channels in 915MHz band, and 3 channels in 868MHz band [10]. In addition to afore-mentioned channels, an amendment to IEEE 802.15.4-2006, IEEE 802.15.4a-2007 [12], specifies alternative physical layers that include an UWB PHY at frequencies of 3-5GHz, 5-6GHz, and <1GHz [12].

IEEE 802.15.4 networks are formed by two different types of devices, full-function devices (FFD) and reduced-function devices (RFD). An FFD can operate as a personal area network (PAN) coordinator, a coordinator, or a regular device. RFDs are intended to be light weight, low-complexity, battery operated devices which can only communicate with FFDs, while an FFD can communicate with both RFDs and other FFDs. Every IEEE 802.15.4 WPAN has at least one FFD, operating as the PAN coordinator. [10]

LR-WPANs may operate in two topologies, star or peer-to-peer, depending on the application requirements. Topologies are illustrated in Figure 3. In star topology, communication in the network is focused around one central controller, the PAN coordinator, which can route communication and may be either the termination point or the initiation point for network communication. In peer-to-peer topology any device can communicate with any other device that is within the range of one another. A PAN coordinator is also present in a peer-to-peer network because it handles network functions such as address assignment and choosing the network's unique PAN identifier. [10]

All IEEE 802.15.4 devices have unique 64-bit addresses which can be used in direct communication. Devices may also be allocated with a short address by the PAN coordinator when the device is associating with a network. IEEE 802.15.4 does not

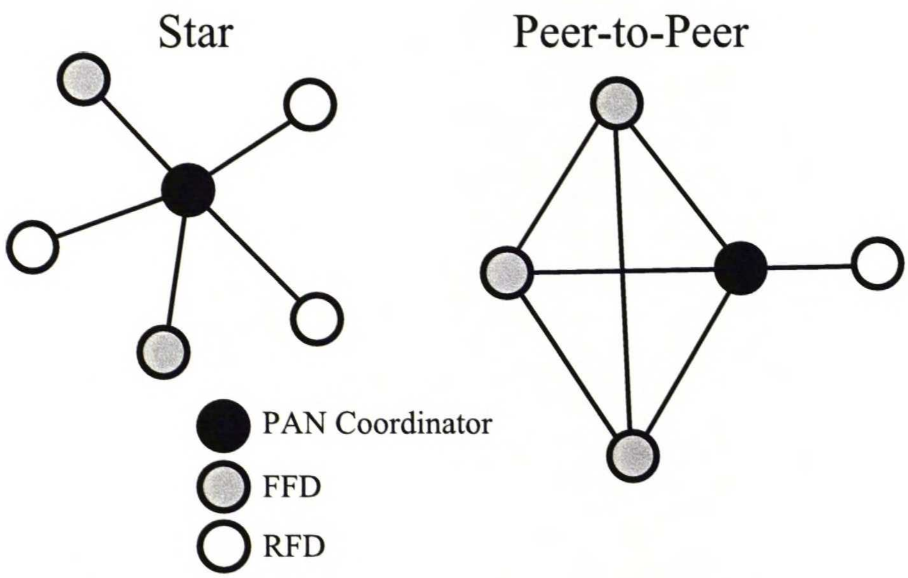


Figure 3: IEEE 802.15.4 star and peer-to-peer topologies [13]

define any means of multi-hop routing or mesh networking. Also functionalities like organized network formation are not within the scope of the standard as they are left to be handled by upper level network layers. [10]

IEEE 802.15.4 defines only the foundation for LR-WPANs with a versatile range of physical layers to choose from. It is widely used in other WSN protocols which add higher lever routing capabilities and networking functions on top of IEEE 802.15.4. IEEE 802.15.4 compatible radios and system-on-chip solutions are widely available from multiple silicon vendors and they are generally inexpensive. In the following section, one important IEEE 802.15.4 based protocol, called Zigbee, will be presented.

## 3.4 Zigbee

Zigbee is an IEEE 802.15.4 based low data rate protocol which aims to deliver reliable wireless communication for sensing and control applications. Zigbee is designed to be cost-effective, secure, open, reliable, and to have low power consumption. To achieve these attributes, Zigbee is constrained to have low data rate. The Zigbee protocol stack is compact enough to fit on a 8-bit microcontroller. Despite its compact size, Zigbee is a full wireless sensor network stack with advanced features such as multi-hop routing, mesh networking, ad-hoc formation of network, dynamic routing, and the capability to scale up to networks with thousands of nodes. [6]

The Zigbee Alliance, formed in 1997 by eight companies, is an organization that promotes and certifies Zigbee products and standards. At the moment the Zigbee Alliance has hundreds of participants in three classes of membership: promoters, participants, and adapters [6]. The Zigbee Alliance is currently focusing on the following application areas as their main markets: Consumer Electronics, Energy Management and Efficiency, Health Care, Home Automation, Building Automation and Industrial Automation [14]. Figure 4 illustrates a comparison of different wireless technologies. This comparison shows that Zigbee fills a gap in the spectrum of existing wireless technologies by providing medium range low data rate communication.

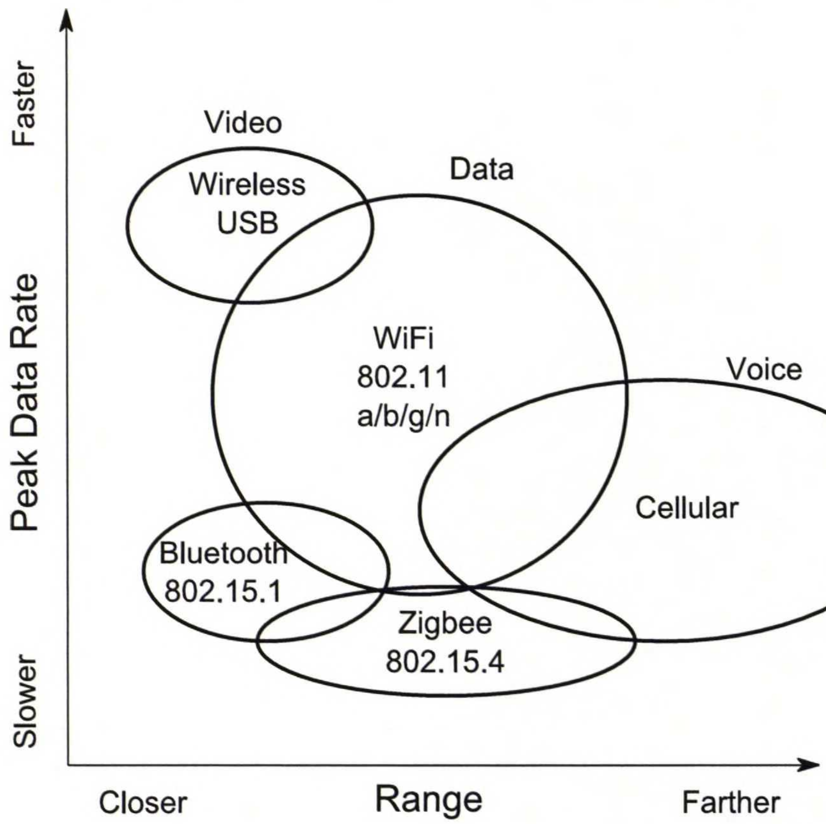


Figure 4: Comparison of Wireless Technologies [6]

In the following sections, aspects of the Zigbee protocol standard that are seen to be the most relevant within the context of this work are reviewed. Also, a brief description of the Zigbee cluster library (ZCL) is given. Zigbee specifications have been released in 2004, 2006, and 2007 from which the latest adds in a new feature set, Zigbee Pro [15]. Different feature set Zigbee devices are compatible with each other when operating as end-devices (see Section 3.4.2). Since Zigbee 2006 and 2007 specifications are fully compatible, in this section the Zigbee 2007 specification is assumed.

### 3.4.1 Architecture

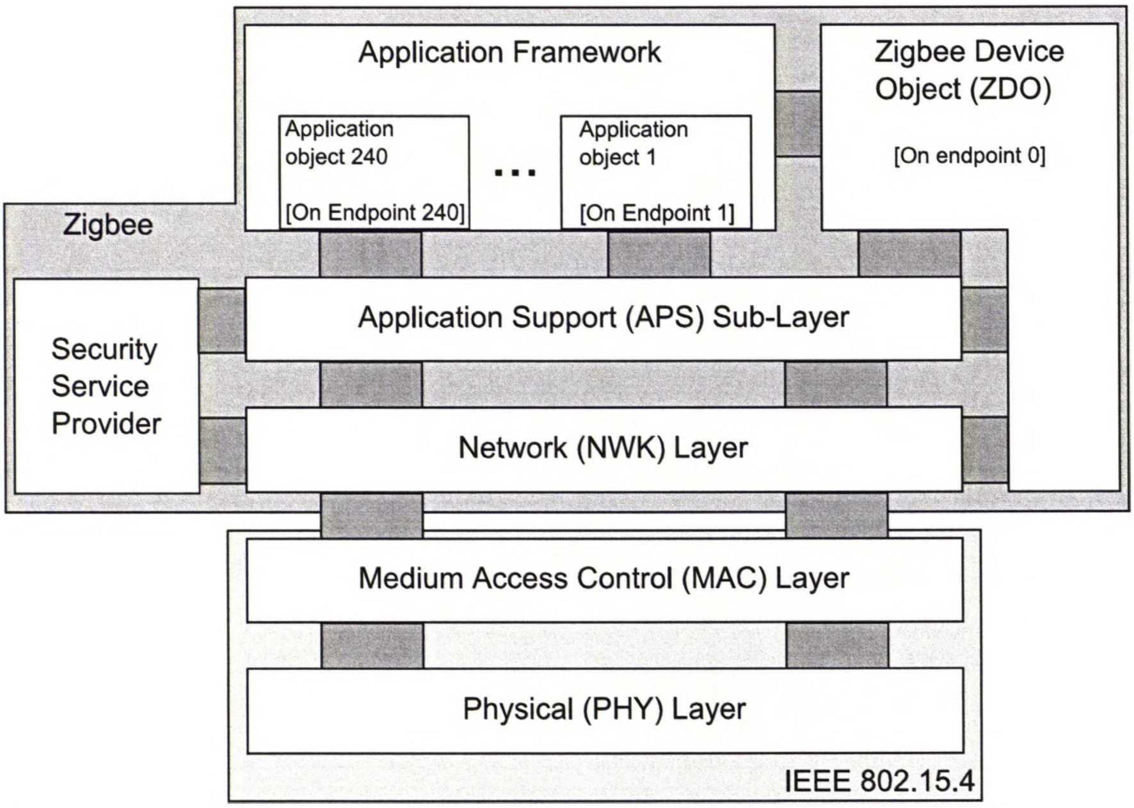


Figure 5: Zigbee stack architecture [15]

The Zigbee stack architecture is illustrated in Figure 5. Each layer of the stack builds on the foundation of its lower levels. The layers are connected to each other over service access points (SAP, indicated by dark gray rectangles in figure 5) which act as an abstraction and provide an application programming interface (API). The lowest two layers of the stack, the MAC and PHY, are defined by IEEE 802.15.4 (see Section 3.3). [6]

The network layer (NWK) handles mesh networking, routing, and broadcasting functionalities. It also enables security features by allowing secured joining, rejoining, and the encryption of the whole content of an NWK frame. The application support sub-layer (APS) filters network packets according to applications that run on the stack. This lightens the application's workload by not having to handle data that is not relevant to it. The APS also handles local binding tables, which indicates individual nodes or groups of nodes in the network that the node can communicate without referencing to the device's address directly, and it enables end-to-end acknowledgment (ACK) for data communication. [6]

The Zigbee device object layer (ZDO) has the responsibility of offering local and over-the-air (OTA) network management functionalities. The ZDO handles the node's state on the network. Last, the applications framework contains a framework within which Zigbee applications run. Endpoints (see Section 3.4.2) within the application framework differentiate separate applications from each other, i.e., one Zigbee stack can be host to multiple applications that are contained within different endpoints. [6]

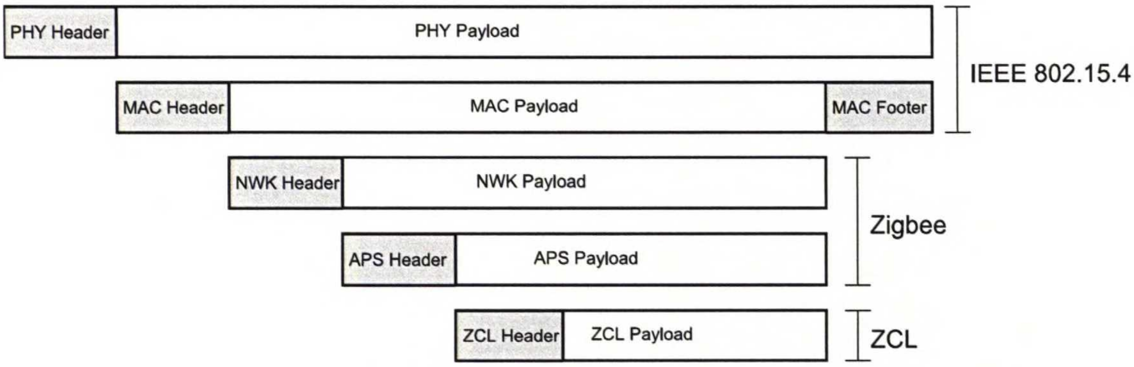


Figure 6: Frame Format of an Application Level Zigbee Message [10], [15], [16]

Figure 6 illustrates an OTA frame for an application level Zigbee message. This figure shows the relations between IEEE 802.15.4 and Zigbee standards. An important aspect of the frame format is that the APS frame's payload is the message's application level data, which in this case, is formed according to Zigbee cluster library (ZCL) definition. In the next section, ZCL and other central Zigbee concept will be introduced.

### 3.4.2 Concepts

In this section, key Zigbee concepts will be reviewed briefly. The objective of this section is not to fully explain how Zigbee works, but to show concepts that are vital in understanding how Zigbee systems are built and operated. These concepts include:

- Device types: end-device, router, and coordinator
- Differentiating networks and applications: channel, PAN ID, endpoint
- Network addressing: IEEE address/MAC address and NWK address/short address
- Interoperability: Zigbee cluster library, application profile, cluster, command, and attribute

- Network management: Zigbee device object (ZDO) and Zigbee device profile (ZDP)
- Messaging: unicast, broadcast, groups, and binding

Deriving from IEEE 802.15.4 standard, Zigbee networks have two different types of devices: full-function devices (FFD) and reduced-function devices (RFD). Zigbee takes this categorization a bit further by defining two types of FFDs which are the Zigbee coordinator (ZC) and Zigbee router (ZR). The RFDs in a Zigbee network are called end-devices (ZED). A coordinator is the only device that can form a Zigbee network and if security features are present, the coordinator has to be available to add new devices to the network, i.e., the ZC acts as a security trust center. A Zigbee network will contain only one ZC but it may contain multiple ZRs. Both FFD type devices, the ZC and ZRs, are otherwise very similar devices and both can be used to route packets and allow other nodes to join the network. [6]

Since ZC and ZR have the ability to route packets in the mesh network, they must be active at all times. This means that FFDs are usually stationary devices that are connected to an inexhaustible source of energy. The end-device, however, is a device that is capable to sleep (have radio turned off) during network inactivity and may be operated on battery power. A ZED will always have a parent node that is either a ZC or a ZR which is capable of temporarily storing data for its child nodes. When the ZED wakes from low-power sleep state, it will poll for its parent for data and it can transmit its own data for the parent to be further routed to the network. Since end-devices usually spend most of their operating time in sleep states and rarely communicate with their parent nodes, they can be mobile in within the network. If a ZED is not within the range of its parent node when polling for data, it will simply attempt to rejoin the network through another FFD. [6]

Zigbee uses the 2.4GHz unlicensed ISM (industrial, scientific, medical) radio band. As specified in IEEE 802.15.4, the channels in 2.4GHz band are numbered from 11 through 26 (16 channels all together). When a Zigbee network is being formed, the ZC can scan all of the channels or a predetermined subset of channels to determine which channel has the least amount of noise and form the network on that channel. Other devices can scan for channels as well when joining a network to see whether networks are available. Multiple Zigbee networks can operate in the same physical space using different radio channels. [6]

The channel is not the only way of separating different networks from each other, however. Multiple networks can co-exist on the same channel as well. The PAN ID is a 16-bit identifier that logically separates different networks from each other. PAN

ID may either be chosen randomly at network formation, in which case other devices joining the network will try to join any available network, or it may be assigned in the application profile definition which is usually the case when using a public profile. [6]

In a Zigbee network, defined by the channel and PAN ID, individual nodes are assigned with 16-bit network addresses, also called short addresses, which uniquely identify a device within its network. The address 0x0000 is always reserved for the ZC. The MAC address, also referred to as IEEE address, is a 64-bit number that uniquely identifies a Zigbee device globally. This means that when switching to another network the device's network address may change but the MAC address stays always the same. As the naming of these addresses already imply, the MAC address is used for per-hop message routing, where as the network address is used for end-to-end routing. [6]

In Zigbee, there is a notation of endpoints (EP) which are similar to Internet sockets, i.e., they provide means to address certain applications within a device. A Zigbee network might contain, e.g., three applications: an air-conditioner unit, a thermostat, and a temperature sensor. Each of these applications might be running on separate devices spread around a common space or they might be located in just one device that has all of the previously mentioned functionalities. In the latter case, the applications would reside in a single Zigbee node in separate endpoints and other devices in the network could address the applications just as they would in a decentralized environment. A single Zigbee node can have up to 240 endpoints. [6]

Messages that nodes send to each other in a Zigbee network are called clusters. Clusters, indicated by a 16-bit number, contain both command and data. They give a message its meaning and they are always directional. This means that, e.g., a switch could contain a cluster indicating on/off action as its output and a lamp could have an on/off input cluster. The switch can then send on/off messages to the lamp but the lamp can not send on/off messages to the switch. Clusters are also always meaningful only within a certain context, or an application profile. Cluster 0x0006 is always the on/off cluster in the public Zigbee cluster library, but in a proprietary cluster it might carry a totally different meaning. [6]

As already referred to, Zigbee contains something called application profiles indicated by 16-bit profile IDs. Every message in Zigbee is sent using a certain application profile, since it is the information that gives a cluster its context. A single Zigbee device can contain any number of different profiles on different endpoints. Profiles can be either public or manufacturer-specific. Public profiles are defined by the Zigbee Alliance and they are designed so that devices that operate on the same profile can interact even if they have different manufacturers. Here are some examples of publicly available profiles: Industrial plant monitoring, home automation, commercial building automation, telecommunication applications, personal home and hospital care, advanced metering initiative. [6]

Interoperability of Zigbee devices is not part of the Zigbee specification, but it is governed by a separate specification, the Zigbee cluster library (ZCL). The ZCL defines most of the clusters that are used in public profiles but the profile definitions may also define additional clusters as well. Clusters in the ZCL are grouped in functional domains such as HVAC and lighting. In private profiles the use of ZCL is optional, but for public profile the ZCL defines clusters which are mandatory for every device, mandatory for a particular device, or optional for a particular device. However, even in public profiles, it is allowed to extend functionality by adding manufacturer specific clusters as extension of the ZCL. In short, it can be said that each public profile is, in a sense, a sub-set of the ZCL with added definitions on radio channel usage, PAN ID, security features, device descriptions, preferred configurations for devices, and other common practices. [6]

Data that is sent over the air is described by a cluster ID and, as stated above, the used profile ID gives the cluster its meaning. When using a private profile without ZCL, there is no specification that restricts the payload of an OTA frame. It is up to the developer to decide what kind of data is to be transmitted. However, clusters operating according to the ZCL contain the notion of commands and attributes. Commands are used to cause action on a cluster and attributes store the state of a given cluster. The ZCL contains a set of cross cluster commands called the ZCL foundation. These commands include functionalities such as reading and writing attributes, discovering attributes, and reporting attributes. Other common clusters, called general clusters, that are found on nearly every device that supports ZCL are as follows: Basic, Power, Device temperature configuration, Identify, Groups, Scenes, On/off, On/off switch configuration, Level control, Alarms, Time, RSSI (received signal strength indication). For more information on these clusters, see the Zigbee cluster library specification [16]. [6]

What should be noted about the ZCL is that it is not a part of the Zigbee stack. The ZCL contains a vast amount of functional features that require some form of logic, such as reporting changes of an attribute that exceed a certain level. This means that the ZCL has a high overhead in both code space and computation, and it is within the domain of the application itself. ZCL guarantees network level interoperability independent of vendors, but it comes with a price tag.

Sending data in a Zigbee network can be done in two ways from which the most common is to use unicast messages, i.e., one node sends data to another node. Unicast messages can be optionally end-to-end acknowledged on the APS layer. Another way of sending messages is to use broadcasting. Broadcasting can be limited by the number of hops the message will be routed forward on the network and broadcast messages can address every node on the network or they can be limited to only nodes that are not sleeping or to FFD nodes only. The Zigbee specification also includes means of targeting predetermined groups of nodes. The so-called groupcasting uses the same underlying mechanisms as broadcasting but it is only an optional feature in the specification. [6]

Besides the above mentioned messaging schemes, Zigbee provides a way to address applications indirectly with bindings. Binding is always unidirectional and it can be used to connect one endpoint to one or more endpoints on other devices. Messages that are sent indirectly using bindings are sent as unicasts. Messages can be sent directly from one node to another when the message's source and destination endpoints, cluster ID, and destination network address are known. However, there can be a situation where the destination's network address changes during normal operation of the network. This can happen, for instance, when an end device loses connection with its parent and has to rejoin the network. Binding addresses the issue of changing network addresses by keeping a local table of the above-mentioned information. The binding table is smart enough to provide mechanisms to automatically update any destination network addresses that might have changed. [6]

Binding is a part of a larger topic of network commissioning and maintenance in Zigbee and WSNs in general. ZCL addresses commissioning by providing clusters that assist with organized network formation. The Zigbee stack itself also includes services for network commissioning and maintenance; the Zigbee device object (ZDO) together with the Zigbee device profile (ZDP). ZDO is an application like every other Zigbee application. It runs on endpoint 0 and uses application profile called ZDP, which is indicated by profile ID 0x0000. Unlike regular applications, however, the ZDO can interact directly with the networking layer of the Zigbee stack. The ZDP includes services such as device discovery, service discovery, binding, and network management. It is generally used for discovering, configuring, and maintaining Zigbee devices and services on the network. Every functionality of the ZDP can be accessed remotely by other devices just like normal ZCL profile clusters. [6]

As a summary the addressing concepts of Zigbee and the associated stack layers are presented in Figure 7. It should be noted that ZCL is not strictly a layer of the Zigbee stack but rather a protocol that is delivered as a payload of Zigbee frames.

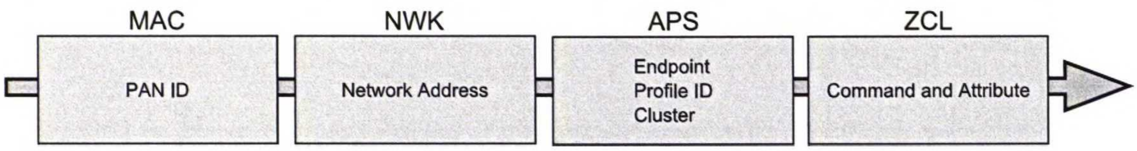


Figure 7: Addressing within Zigbee [6]

### 3.4.3 Implementation

It should be noted that the Zigbee specification is not concerned with the implementation of the Zigbee stack. It is only concerned with over-the-air operation of the stack. This means that Zigbee does not have a common API. Applications that are written for Texas Instruments Zigbee API are not directly portable to Ember Zigbee API. There are many vendors that offer both hardware and software solutions for Zigbee, but there are also vendors that concentrate on the stack implementation only. [6]

The Zigbee Alliance has a certification program for Zigbee platforms that ensures interoperability of different implementations. If a device is to interact with other public profile devices it has to be certified. With the term platform Zigbee Alliance covers the combination of IEEE 802.15.4 radio, microcontroller unit (MCU), and Zigbee stack software [6]. Different ways of implementing a Zigbee platform are listed below and illustrated in Figure 8, Figure 9, and Figure 10. The listing and comparison of the different implementation methods is based on Gislason's Zigbee Wireless Networking [6] and Texas Instrument's Low-power RF Guide [17]

- System on chip, illustrated in Figure 8.
  - Everything is included on a single chip.
  - Advantages: Small footprint on PCB, low cost.
  - Disadvantages: Application development bound to a predetermined processor and toolchain, memory constraints might be an issue due to high stack overhead.

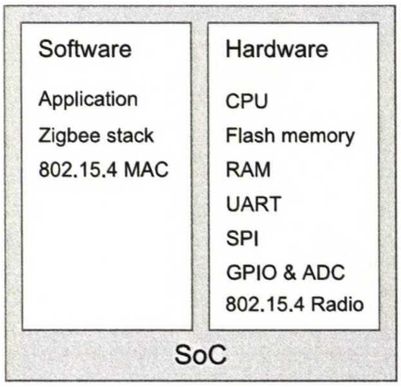


Figure 8: System on chip solution with integrated 802.15.4 radio, MAC and Zigbee stack software on the same processor with application

- MCU with stand-alone radio, illustrated in Figure 9.
  - The Zigbee stack resides on the MCU which controls an IEEE 802.15.4 compliant radio. The host application is on the same processor as the stack.
  - Advantages: Less overhead than in a SoC solution, radio can be freely chosen from multiple vendors.
  - Disadvantages: Zigbee stack software can be difficult to obtain on a specific processor.

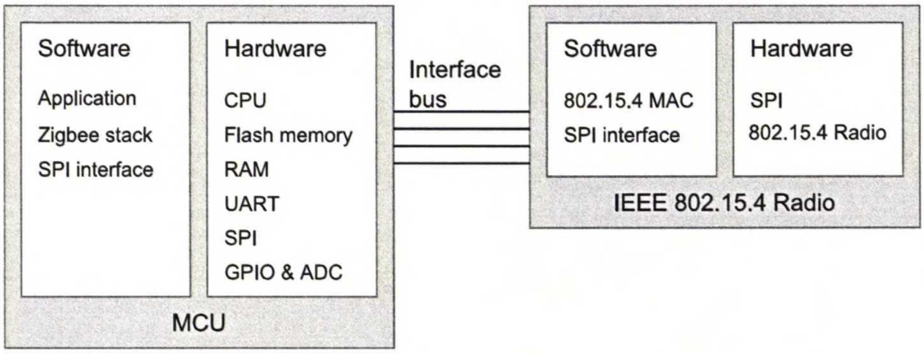


Figure 9: Zigbee stack on an MCU with external 802.15.4 interface

- Communication module, illustrated in Figure 10.
  - Zigbee stack runs on a separate processor. MCU handles host application and the communication module handles Zigbee networking.
  - Advantages: Flexible configuration, can be used with any processor, clear division of tasks; host processor handles the application while network processor handles the Zigbee stack.
  - Disadvantages: Large footprint on PCB, expensive.

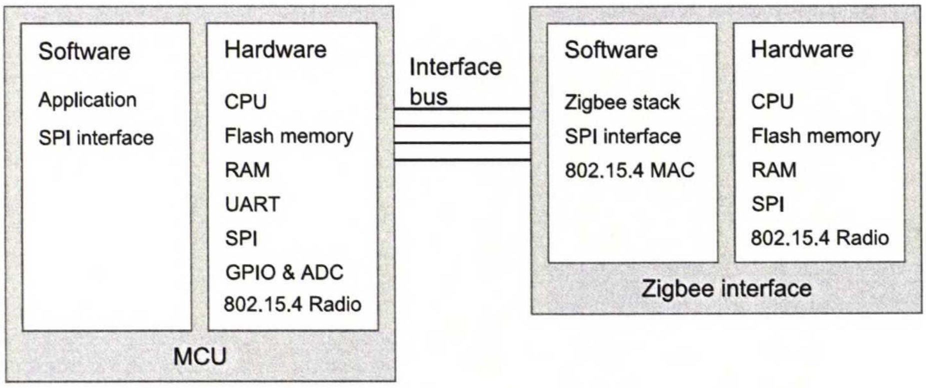


Figure 10: Zigbee as a communication module

The easiest way of implementing a Zigbee platform might be to obtain a readily available Zigbee module. These modules are usually either communication modules, that will have an interface which is used to control a processor running a Zigbee stack, or SoC solutions that can be programmed to use multiple different IEEE 802.15.4 based protocol stacks, including Zigbee. The biggest advantage with modules is that they are often already Zigbee certified platforms. The most significant drawback of Zigbee modules is, of course, higher price. [6]

# 4 Applications of Wireless Sensor Networks in Assistive Automation

This section discusses the utilization of wireless sensor networks in assistive automation applications and presents the state of the art. Although wireless sensor network technologies have a huge potential in improving the quality and functionality of assistive automation applications, WSNs in assistive automation systems are not widely used today. This is partly due to the relative novelty of WSN technologies, but it can be assumed that the complexity of WSN technologies can be a factor that delays its adaptation.

Today there are some commercial assistive automation applications available that use simple point to point radio communication. E.g. elderly people can wear special bracelets that have buttons that will alert caretakers when pushed. Sensor elements that are placed underneath a mattress in a bed can be used to detect if a person leaves his/hers bed. This information can be used to trigger bathroom lighting at night time or an alarm can be raised if the person does not return to bed after certain timeout period.

Tunstall corporation has recently launched an activities of daily living (ADL) monitoring solution, called ADLife. This system uses a range of electronic appliance monitoring devices, magnetic switches, PIR sensors, and occupancy sensors that communicate with a central unit by using a 869MHz radio [18]. Everon corporation has also developed and commercialized similar wireless systems that are based around a central GPRS hub. Everon offers a wide range of devices that can be connected to the central hub. The GPRS hub can be configured to send SMS alarms according to information that is provided by connected devices and the GPRS hub can also be used to control these devices. Applications of this system can be e.g. nurse call, passage control, bed alarm, fire safety, burglar alarm, stove guard, activity monitoring, device control, or environmental monitoring [19].

Both Tunstall's and Everon's application could benefit from the use of WSN technologies in number of ways. E.g., Zigbee adds reliability in the operation of devices by adding redundancy in the routing of network packages. Also, applications that are implemented with Zigbee technology could be smaller in size and need less battery capacity. Most importantly, when using Zigbee, interoperability of sensors and applications from different vendors could be achieved.

Everon has also developed wearable safety systems that can be categorized as assistive automation applications. The Everon Vega GPS bracelet is a device that is designed for people with Alzheimer's disease and for people who suffer from episodes of confusion that can lead to wandering. The system establishes a virtual security zone around the person's house. The person is allowed to move within familiar surrounding, but if he/she leaves farther away from home, an alarm with GPS coordinates is sent over cellular telephone network. The alarm is sent to a response center, which will then react to it according to a predetermined procedure protocol. [19]

Everon's Pers mobile GPS emergency system share the same hardware with the Vega GPS bracelet. Pers mobile is an emergency call system that will allow its wearer to contact a response center by simply pushing one button. The system will initiate voice conversation with the emergency center and send the user's GPS coordinates to the response center operator. At home, the system's location is known by using a home base station device [19]. The Vega GPS bracelet and Pers mobile are system's that would not benefit greatly from the use of WSNs, since they need very long range communication. This goes to show that WSNs are not the only solution for assistive automation applications, but it should be applied along with other technologies as well.

Although wireless sensor networks are not being utilized in large scale commercial applications, review of recent studies shows that wireless sensor networks are being applied to assistive automation application prototypes. Two common research subjects are monitoring of activities of daily living and fall detection. A paper by Bhatia et al. [20] presents an energy efficient Zigbee system for measuring activities of daily living. The study shows that a Zigbee accelerometer unit with proper data analysis capabilities can be used to track ADL events for weeks with a small power source, such as a coin battery. Casas et al. [21] apply a large scale Zigbee network for fall detection in rural areas. In their paper, they present a fall detection badge that will be carried by elderly people. A Zigbee network that covers an area of a whole village is deployed to route messages of these badges.

Localization of individual nodes in wireless sensor networks is also a frequently occurring research subject. If efficient localization methods are developed, they can be put to use in assistive automation applications, as well as in many other commercial application. Hussain et al. [22] propose integration of RFID and WSN technologies for localization and target identification. In their paper, Hussain et al. envision an elderly care system that identifies caregivers as they enter a room and control lighting in the house automatically according the preferences of individual users.

An important factor in successfully deploying wireless sensor networks in assistive automation applications is to consider end-users opinions about such systems. A qualitative study by Steele et al. [23] examines the perceptions, attitudes, and concerns of elderly persons toward wireless sensors network technology. The study shows that independence is highly valued by elderly people and any system that could prolong that independence is highly regarded. Participants of the study were mostly concerned about the cost of such systems. Systems that require the user to wear something, such as a wrist pendant, were perceived as negative, since there is a risk for the user to forget to wear the device. Also, there might be shame issues related to such devices, since they show other people that the user is in need of outside assistance.

Steele et al. also report that elderly people are generally not concerned about confidentiality of any health information that a WSN could carry, as long as a camera is not used to monitor them. Some concerns were also raised about the ability of elderly people to interact with WSN system. This indicates that assistive automation systems should be designed so that they need as little input as possible from the end-user. Although, participants of the study express that the end-user has to have some level of control regarding the system.

# 5 Ilmari Wireless Sensor Network Platform

Motivation for an affordable and adaptable wireless sensor network platform in assistive automation is evident. Not only do the applications vary widely in nature but also the main application area, the common household, is a vastly heterogeneous and unpredictable environment. Therefore there is a need for a wireless sensor network platform that enables the prototyping of assistive automation applications with ease of implementation and installation while maintaining the ability to be fully customized and flexibly configured.

A platform in this context is seen as a technology foundation, including both the hardware and the software, that enables the implementation of wireless sensor network applications. This chapter describes the wireless sensor network platform that was designed and implemented as a part of this work. The platform is designed especially keeping in mind the different, often conflicting, requirements of various assistive automation applications. The platform is named "Ilmari" which is the middle name of the author but also adapts the Finnish word for "air".

The structure of this chapter is as follows. First, the premise of the platform development is defined. Next, key aspects and criteria that were considered in the development of the platform are discussed. System level design of the platform is reviewed in Section 5.3 and the hardware and software design of the platform is described in Sections 5.4 and 5.5 respectively. Finally, Section 5.6 presents test procedures and results for the platform's key attributes; power consumption and wireless range.

## 5.1 Premise

When first setting out to devise a wireless sensor network platform that would fit the needs of assistive automation applications a list of different requirements for the platform was composed out of probable use cases for different applications. These applications are discussed with more detail in Section 6. However, it should be noted that both end-user applications and applications sold as a service were considered. Also stand-alone applications as well as applications operating alongside with home server infrastructure were taken into account.

The requirements for the platform are as listed:

- Individual applications that are implemented with the platform should be able to function without existing infrastructure.

- Wireless devices should be able to operate for considerable times without battery change.
- Minimum effort should be needed to associate new devices or the system should recognize new devices automatically.
- The devices should be small in physical size in order to enable easy placement and installation.
- The system should enable the monitoring of individual device's operating states.
- The system should have a gateway to a home server in order have an interface for the end-users, for remote monitoring over the Internet, and configuring and maintaining the system.
- The system's parameters should be configurable after the initial installation.

In addition to these requirements a set of optional features was formed. This list includes features that were seen as possibly useful but not as something the system could not operate without. Optional features for the system are listed as follows:

- The system should enable means for over-the-air firmware updates of individual devices.
- Individual devices in the system should be able to operate even while being moved around within the range of the wireless sensor network.
- Different devices and applications in the network should be able to share information without excessive effort. For example the same sensor information could be utilized in multiple applications within the same network without special measures.

With the above-defined premise set, special design considerations that are associated with the Ilmari platform can be elaborated on in the following section.

## 5.2 Design Considerations

First design criteria for the platform is that it should be relatively easy to develop on. Wireless sensor network protocols usually tend to have a lot of overhead which means that the developer has to familiarize oneself with vast amounts of detailed protocol procedures and learn a huge code base by heart before being able to write practical applications. Some means of abstraction for the WSN protocol stack besides the standard API, if possible, is clearly needed.

Of course when talking about wireless system one key attribute is the battery life or more accurately the power consumption of the system. Battery life can be prolonged to some extent by increasing battery capacity but in the long run the ability to restrict the overall system's power consumption is vital. Lifetime of the system should be measured in years rather than weeks. This means that the communication protocol of the system should be purposely designed for WSN use.

Next comes the cost issues. A practical system can not be too expensive. An individual application can require installation of tens of individual devices. This means that the unit cost of one device can not be too high in order for the whole application to be cost-wise feasible. Also, the cost of development tools has to be considered. It is not sensible to develop a prototyping platform that will require thousands of euros worth of software licenses.

Last, the system should be as multi-functional as possible. This calls usually for either a highly modular or a feature rich platform. As previously stated, the nature of the assistive automation applications that can be conceived to be implemented with the Ilmari platform can vary greatly. With this perspective it is justified to say that a modular system fits the purpose better than a feature rich system that has to accommodate for every foreseeable function. Also the cost aspect supports a modular system.

## 5.3 System design

The field of wireless sensor networks is in a stage of rapid evolution. Since the advent of IEEE 802.15.4 (see 3.3) there has been a significant growth in the amount of WSN protocols and vendors [6]. When choosing the protocol that is to be used with Ilmari platform emphasis was put on the number of companies promoting the technology, openness and maturity of the protocol, and the perceived activity around the protocol.

Zigbee was chosen as the core technology to be used in Ilmari platform because it fits well with the above mentioned qualifications and it is also well aligned with the requirements set in Section 5.2. The field of wireless sensor networks is still very fragmented since WSNs are a relatively new concept and WSNs are, at the moment, relatively rarely used in commercial applications. This means that there are a lot of different vendors offering solutions for WSNs. In addition to Zigbee, Z-wave and IEEE 802.15.4 based technologies were considered for Ilmari platform. Z-wave has a firm stand in the home automation market but it is a proprietary and closed technology. Many smaller vendors offer IEEE 802.15.4 radio based nodes that can be used with various different protocol stacks. These, however, rarely offer an upper level multi-hop routing protocol.

The next step after deciding on Zigbee is to decide on how to the implement it. Section 3.4.3 discusses different ways of implementing Zigbee platforms and, as mentioned in that section, there are many ways to do this. Here the use of a SoC solution versus a separate network processor will be discussed.

Texas Instruments is one of the promoter members of Zigbee Alliance and TI's Z-stack is a certified implementation of the Zigbee stack. A versatile SoC, like the Texas Instruments CC2430, described in CC2430 datasheet [24], could seem like the ideal hardware base for a Zigbee platform. The CC2430 has multiple protocol stacks available: Z-stack (Zigbee), TIMAC (IEEE 802.15.4 MAC), Nanostack (6LoWPAN), RF4CE (Zigbee), SimpliciTI (proprietary) [17]. The CC2430 is a compact and inexpensive solution that would be well-suited for a commercial product, but it can not be easily integrated into existing applications. Also the CC2430 would need commercial software to compile and program its Zigbee stack.

The communication module implementation, described in Section 3.4.3, was seen as a feasible approach for Ilmari platform for numerous reasons. A communication module does not dictate what kind of processor should be used in the application or what kind of tool-chain has to be used to compile and program the application. Also the communication module could be, if necessary, integrated into existing applications that run on different types of processors. Texas Instruments CC2480, described in CC2480 datasheet [25], was chosen for Ilmari platform. The CC2480 is a Zigbee processor that offers an SPI interface to Zigbee applications. Ember offers a network processor EM260, described in EM260 datasheet [26], that is quite similar to the CC2480. The EM260 was seen to have a slightly lower level interface than the CC2480 and therefore CC2480 was chosen over EM260.

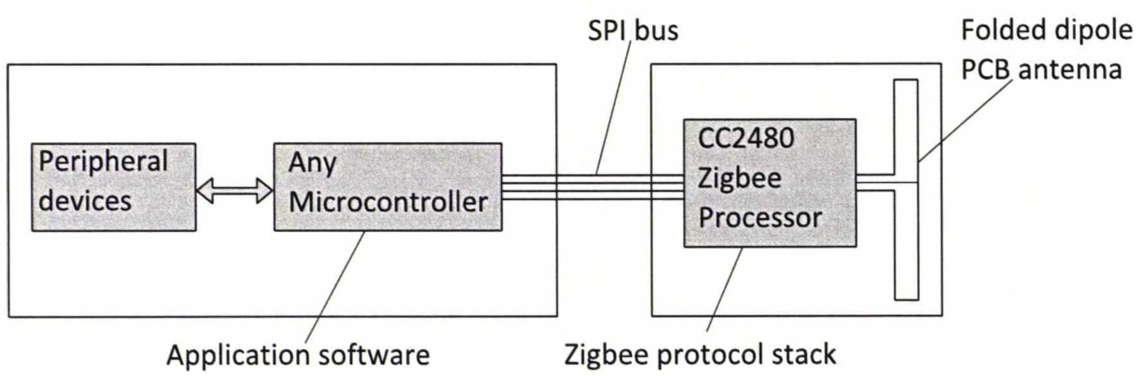


Figure 11: Ilmari Platform Architecture

Architecture of the platform is illustrated in Figure 11. With the CC2480 as a network processor, Ilmari platform can use any microcontroller that is capable of SPI communication as its application processor. A low-power 8-bit Atmel AVR processor was chosen as the application processor. Hardware design of the Ilmari platform is discussed in detail in the following section.

## 5.4 Hardware Design

Ilmari platform is a modular system which has a separate microcontroller unit (MCU), Zigbee processor, and peripheral devices such as power and sensing modules. Every node is assembled from these elements as seen best to fit for its function. In this section the hardware that was developed as a part of this work is presented. The layouts and schematics for the hardware are presented in Appendix A.

### 5.4.1 Radio Module

At the time when Ilmari platform was being designed there was no readily available CC2480 modules on the markets. Because the CC2480 was seen as the best candidate for the Zigbee network module, an original module design was needed. Texas instruments offers an application note on designing a folded dipole antenna for the CC2480 [28]. Together with the antenna design application note and a reference design for CC2480 [29] it is possible to design and implement a Zigbee radio module with reasonable amount of effort.

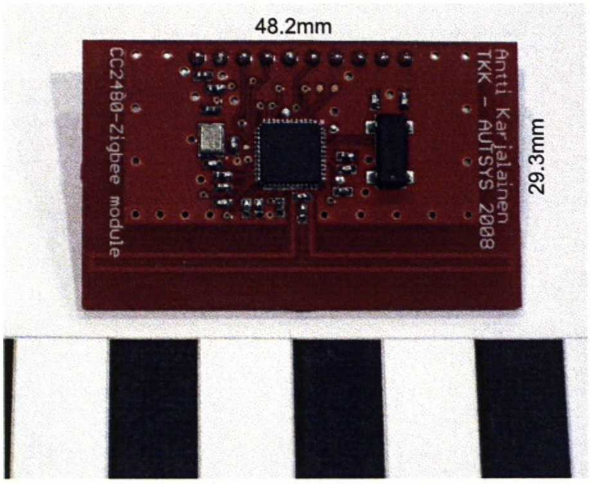


Figure 12: Zigbee Network Module; centimeter scale

The Zigbee module for Ilmari platform is presented in Figure 12. The module has a

10-pin header row which is used to connect the module to its host MCU. This header has the following pins: two configuration pins (CFG0 and CFG1)), SPI clock, SPI MISO and MOSI lines, slave select signal (MRDY), slave interrupt signal (SRDY), reset signal, voltage and ground. The pinout of the connector is illustrated in Figure A2 and Figure A4.

Later Radiocrafts corporation released a CC2480 module, the RC2300-ZNM, described in RC2300-ZNM datasheet [27], which was adapted into use in Ilmari platform. The RC2300-ZNM module with integrated antenna and custom motherboard is presented in Figure 13. The motherboard for the module is designed according to Radiocrafts' layout recommendations, found in the RC2300-ZNM datasheet [27]. The motherboard's pinout, illustrated in Figure A4, is identical to the above-described interface.

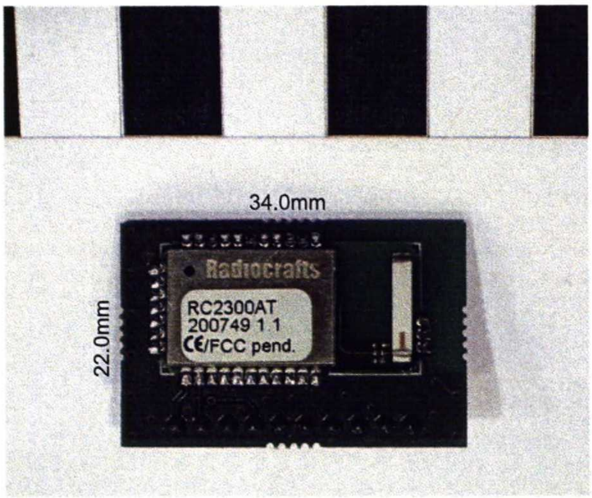


Figure 13: Radiocrafts RC2300-ZNM Zigbee Network Module; centimeter scale

### 5.4.2 Microcontroller Unit

When choosing the microcontroller for Ilmari platform, the main design criteria were power consumption, price, connectivity, and tool-chain. The low-power and low-price requirements lead the initial decision for the favor of 8-bit RISC microcontrollers. Multiple vendors offer microcontrollers that are capable of attaining sub-1$\mu$A sleep currents. Atmel has a product line of PicoPower AVR microcontrollers that offer competitive power consumption specifications. The main advantage with Atmel's AVRs is that they can be used with open source software development tools supported by the WinAVR project.

The ATmega644PV, described in Atmel's datasheet for ATmega 164P/324P/644P

[30], was chosen for the application processor of Ilmari platform. According to the datasheet [30] the main features of ATmega644PV are listed as follows:

- 64K bytes of in-system self-programmable flash memory
- 2K bytes EEPROM
- 4K bytes internal SRAM
- JTAG interface
- Peripheral features:
  - Two 8-bit timer/counters
  - One 16-bit timer/counter
  - Real time counter with separate oscillator
  - Six PWM channels
  - 8-channel, 10-bit ADC
  - Two-wire serial interface
  - Two programmable serial USART
  - Master/slave SPI serial interface
  - Programmable Watchdog timer with separate on-chip oscillator
  - On-chip analog comparator
  - Interrupt and wake-up on pin change
- 6 sleep modes
- Operating voltage range: 1.8 - 5.5V
- Speed grades: 0 - 4MHz @ 1.8 - 5.5V, 0 - 10MHz @ 2.7 - 5.5V
- Power consumption at 1 MHz, 1.8V:
  - Active:0.4mA
  - Power-down mode: 0.1$\mu$A
  - Power-save mode, with 32kHz RTC: 0.6$\mu$A

The MCU is presented in Figure 14. The MCU is designed so that as many of the microcontroller's features as possible would be available for prototyping purposes. The PCB contains two oscillators; one 4 MHz crystal oscillator as the microcontroller's main clock source and one 32.768 kHz crystal oscillator that can be used as a real-time counter and to provide external interrupts during low-power sleep states. The oscillators as well as the microcontroller's voltage inputs and analog supply are connected according to Atmel's hardware design considerations application note [31].

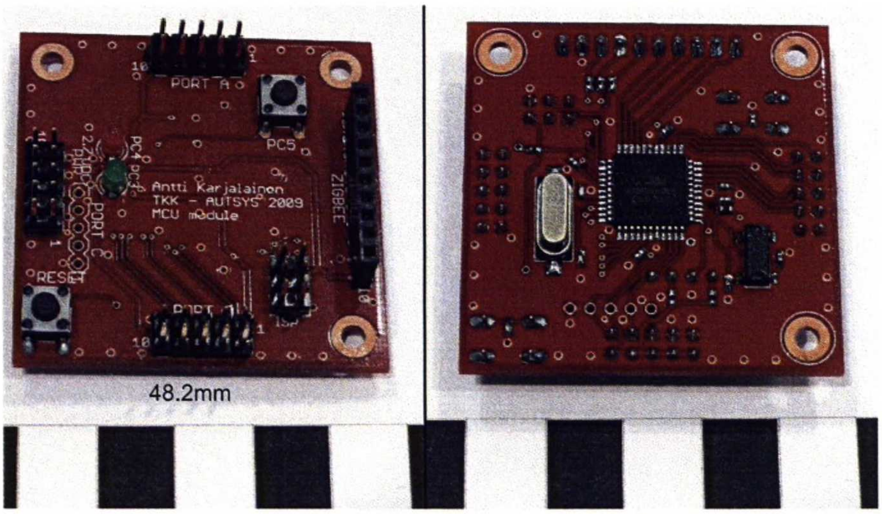


Figure 14: Ilmari Platform Microcontroller Unit; centimeter scale

The MCU contains two buttons from which one is the reset button and the other can be used as an external interrupt source. There are also two LEDs, red and green, which can be freely controlled by the microcontroller. The LEDs are connected to series resistors so that the microcontroller will sink current when the LEDs are illuminated. The reset line is pulled up with an external resistor and the reset button will draw the line low when pushed. The other button will also conduct to ground when pushed. Both buttons are equipped with decoupling capacitors.

Connectors on the MCU are arranged according to the microcontrollers ports with added ground and voltage connectors (pin 10: voltage, pin 9: ground). ATmega644PV has four ports; A, B, C, and D port from which each has eight pins. Each pin can operate as a general purpose input and output (GPIO) pin and each pin can also be used as an external pin change interrupt (PCINT) source. In addition to GPIO functionality, each pin has additional features that are listed in the ATmega644PV datasheet [30]. The MCU's port layout and functionalities are presented in Figure 15.

The Zigbee module for Ilmari platform is designed to be connected to the port B, labeled "ZIGBEE" on the PCB. The microcontroller on the MCU supports in-system programming (ISP) and it is programmed using a serial programming interface with dedicated pin header. The power connector on the MCU is a 10-pin header for added mechanical stability. However, not all of the pins on the header are connected to either ground or the voltage line. There are three pins on the header which can be used as GPIO pins or two of them can be used in a 2-wire serial bus. This feature is added to support any kind of smart power solution that could be used as the

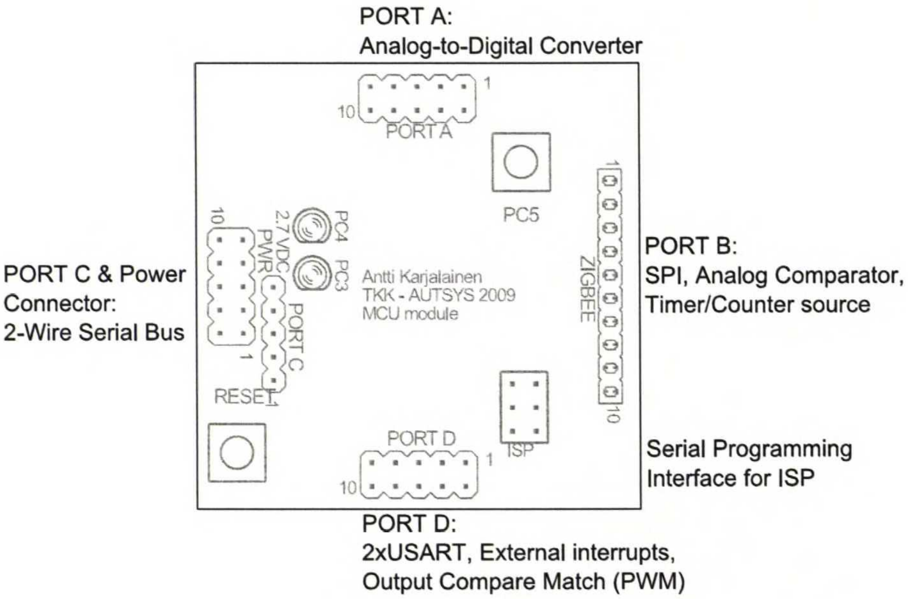


Figure 15: Microcontroller Unit Port Layout and Additional Pin Functionalities

power source of the MCU. For more detailed information on the MCU's layout and schematics, see Figure A5 and Figure A6.

### 5.4.3 Peripheral Devices

The modular design of the Ilmari platform allows peripheral devices to be stacked on top of the MCU board. This way different types of devices can be created by using the same MCU and radio hardware. In Figure 16, two power modules for the platform are presented.

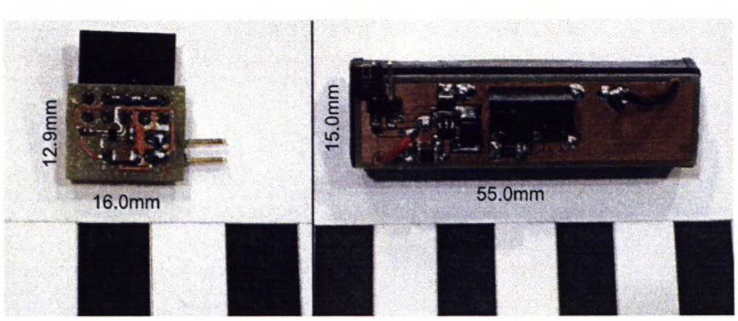


Figure 16: Regulator and Battery Modules; centimeter scale

Both power units provide 2.7V output voltage. This voltage level was chosen because it fits well with the operating voltage ranges of the MCU and the Zigbee module, while being low enough to accommodate for sensor units that are rated for maximum

3.0V. The Zigbee module is rated for maximum 3.6V and the microcontroller can be used with voltages as high as 5.5V. This means that, instead of 2.7V, more common 3.3V power supplies could be used with the platform if other peripheral devices are rated for it. The MCU's operating frequency, 4.0MHz, is selected so that it supports operating voltages starting from 1.8V. The Zigbee module, however, requires minimum of 2.0V, thus setting the operating voltage range of the Ilmari platform in the range of 2.0 - 3.6V.

The regulator module uses a tiny 2.7V 150mA ultra-low-noise low-dropout regulator from Maxim, described in MAX8840 datasheet [32]. The regulator module has two pins that will connect to an external power supply of maximum 6V. The regulator module is designed to power a Zigbee router unit or a Zigbee coordinator unit.

The battery module uses a low-quiescent-current step-up DC-DC converter from Maxim, described in MAX1724 datasheet [33]. The converter is designed to be extremely efficient in light-load conditions with only 1.5$\mu$A quiescent supply current. The PCB is designed to fit seamlessly to the base of a regular 1.5V AA battery holder. The battery module will provide 2.7V operating voltage for a Zigbee end device. There is a two-pin header on the PCB that is intended to be connected to a power switch. If external power switch is not used, a jumper can be connected to the header two turn on the converter. Figure 17 illustrates how a platform can be assembled using the RC2300-ZNM radio module and a battery module.

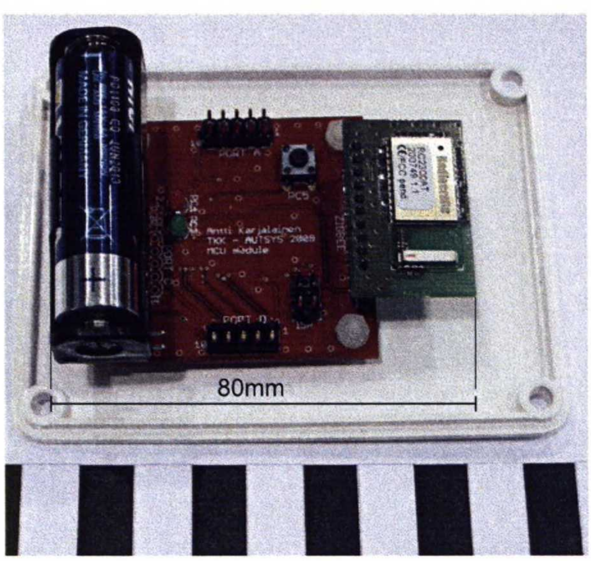


Figure 17: MCU with RC2300-ZNM Radio on the right and Battery Module on the left edge; centimeter scale

The step-up converter will consume power and increase the overall power consumption of the system. However, with a step-up converter it is possible to keep the input voltage of the system steady at all times. Output voltage of a battery will decrease during its discharge. If a device would be powered with an unregulated battery power, over time the output voltage of the battery could decrease enough to go below the cut-off voltage of the device. In such case, the device could not operate even though the battery would still have some capacity left. With a step-up converter it is possible to drain a single AA battery to as low as 0.83V (according to MAX1724 datasheet [33]) while maintaining full operation.

Of course, a key element of a wireless sensor network platform is the sensors itself. In Figure 18 a combined PIR sensor and switch module is presented. The sensor module is designed to be attached to the MCU with a 10-pin IDC cable. The cable interface will provide the sensor with operating voltage and two interrupt lines. The PIR sensor element, Panasonic Napion AMN41122, described in Panasonic's MP Motion Sensor technical documentation [34], is a low-voltage low-current digital PIR sensor that is connected to an on-board transistor. The transistor will drive an interrupt line high when the PIR element is sensing.

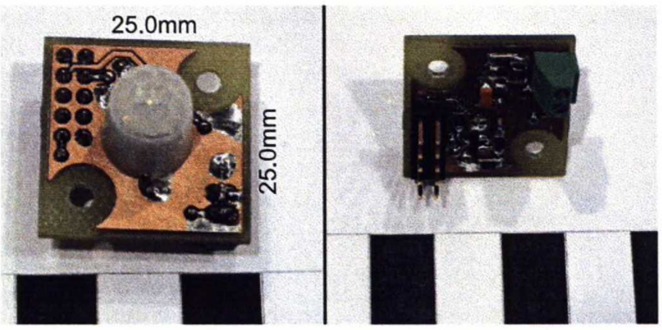


Figure 18: PIR Sensor and Switch Module; centimeter scale

The switch section of the sensor module can be used, e.g., with a magnetic switch component. It is a formed by a series resistor, a capacitor, and a terminal block with two connectors, as illustrated in Figure A8. The switch will hold an interrupt line high while the terminal block's connectors are not connected with each other. Figure 19 shows how the sensor module can be attached to an enclosure to form a Zigbee motion detector node.

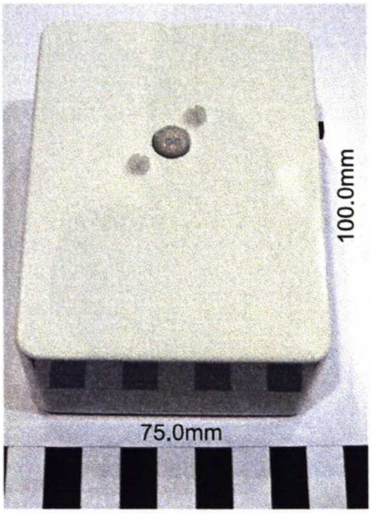


Figure 19: PIR Sensor Node; centimeter scale

## 5.5 Software Design

In this section, the software foundation for Ilmari platform is presented. The software is designed so that the Zigbee radio interface can be ported to any other platform. The MCU's software framework provides basic utilities and access to on-board functionalities and external hardware. The MCU's software is also designed in a way that enables applications that are written for the Ilmari platform to be ported on other platforms, such as a Linux PC.

### 5.5.1 Zigbee Interface Protocol

The Zigbee module is a network processor that offers an serial peripheral interface bus (SPI) interface to Zigbee functionalities with remote procedure calls (RPC). Every command supported by the interface have been implemented in the Ilmari platform's software foundation. Here, this interface is explained in brief. More detailed explanation can be found in the CC2480 interface specification [35].

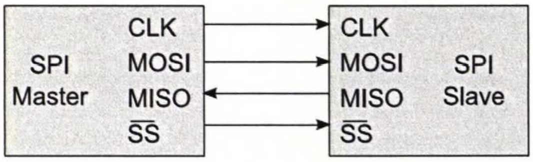


Figure 20: Signals for a Generic SPI Bus

SPI is a full-duplex single master, multiple slave bus. SPI has four lines that connect each slave to the bus master. These lines are illustrated in Figure 20. The clock signal (CLK) is generated by the master and the slave will use the same clock signal to transmit data to the master. The slave's data will be transfered on the master input, slave output (MISO) line and the master's data will be transfered on the master output, slave input (MOSI) line. Slave select line (SS) is used by the master to initiate data transfer. [36]

The CC2480 operates as an SPI slave and the host processor is the SPI master. The SPI interface of the CC2480 is augmented with a dedicated slave ready (SRDY) interrupt line which is used in SPI transaction handling and power management. This line enables the CC2480 to initiate communication with the host processor. In the physical interface, there is also a master ready (MRDY) line that is, in this implementation, hard-wired to SPI's slave select line.

Message types on the CC2480 SPI interface are as follows: POLL messages, synchronous requests (SREQ), synchronous responses (SRSP), and asynchronous requests (AREQ). A POLL message is used by the master to retrieve queued data. A synchronous request is a message, generated by the master, that requires immediate response in the form of a synchronous response (SRSP), which will be generated by the slave. Asynchronous requests do not generate immediate responses and they can be generated by either the host or the slave. [35]

The above-mentioned message transaction scenarios are illustrated in Figure 21, Figure 22, and Figure 23. The figures and descriptions are adapted from CC2480 interface specification [35]. These scenarios form the basis for the Zigbee module's software framework.

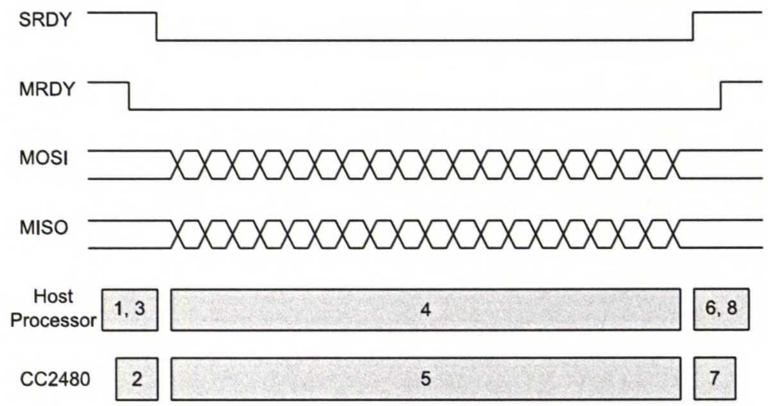


Figure 21: CC2480 AREQ Command [35]

Procedure of sending an AREQ data frame from host processor to the CC2480 (see Figure 21):

1. Host processor sets MRDY low to indicate that it has an AREQ frame to send.

2. CC2480 sets SRDY low to indicate that it is ready to receive data.
3. Host processor starts data transmission.
4. Host processor transmits until frame is complete.
5. CC2480 receives full frame.
6. Host processor waits for SRDY to go high.
7. CC2480 sets SRDY high to indicate that frame has been received.
8. Host processor sets MRDY high.

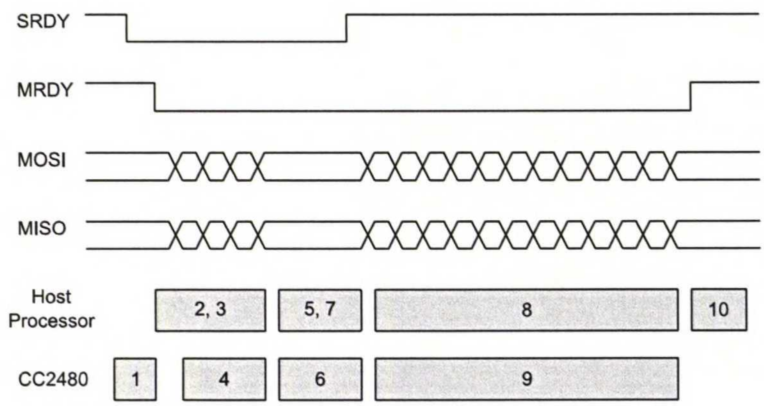


Figure 22: CC2480 POLL Command [35]

Procedure of sending a POLL command from host processor to the CC2480 (see Figure 22):

1. CC2480 indicates that it has an AREQ to send by setting SRDY low.
2. Host processor sets MRDY low and prepares POLL command.
3. Host processor transmits POLL data frame.
4. CC2480 receives POLL data frame.
5. Host processor waits for SRDY to go high.
6. CC2480 prepares AREQ data frame and sets SRDY high when ready to transmit.
7. Host processor starts data reception.
8. Host processor receives data until frame is complete.
9. CC2480 transmits until frame is complete.
10. Host processor receives full frame and sets MRDY high.

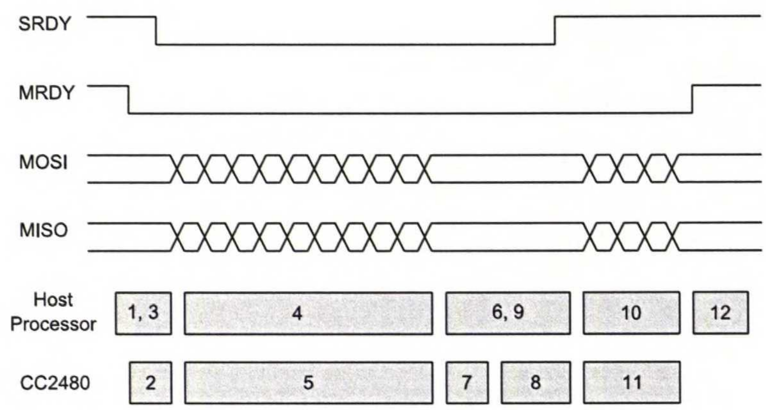


Figure 23: CC2480 SREQ Command [35]

Procedure of sending a SREQ command from host processor to the CC2480 and receiving a SRSP from CC2480 to the host processor (see Figure 23):

1. Host processor indicates that is has an SREQ frame to send by setting MRDY low.
2. CC2480 set SRDY low when ready to receive.
3. Host processor starts data transmission.
4. Host processor transmits until frame is complete.
5. CC2480 receives until frame is complete.
6. Host processor waits for SRDY to go high.
7. CC2480 processes SREQ command and executes function.
8. CC2480 prepares SRSP frame and indicates by setting SRDY high when ready.
9. Host processor starts data reception when SRDY is high.
10. Host processor receives until data frame is complete.
11. CC2480 transmits until data frame is complete.
12. Host processor receives full data frame and sets MRDY high.

The CC2480 interface frame and command formats are illustrated in Figure 24. Each frame starts with length byte, indicating the length of the data field (0-253). Command field indicates the frame type. The data field contains variable amount of frame data. The command formatting consists of two bytes, from which the first indicates the command type and subsystem and the second indicates the command ID which identifies the command within its subsystem. Possible command types are listed in Table 2 [35].

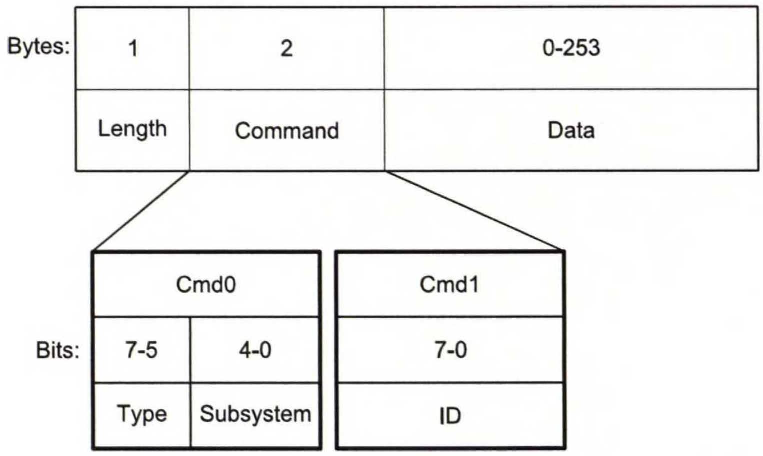


Figure 24: CC2480 Frame and Command Formats [35]

| Value | Name | Description |
|---|---|---|
| 0 | POLL | Poll Command |
| 1 | SREQ | Synchronous Request |
| 2 | AREQ | Asynchronous Request |
| 3 | SRSP | Synchronous Response |

Table 2: CC2480 Interface Command Types [35]

Different command subsystems are listed in Table 3. The SYS interface provides a low-level access to CC2480's hardware and software. It provides, among others, the following functionalities: reset request, random number generator, software timers, SPI interface loop-back test, and radio test. The Simple API (SAPI) interface provides a limited set of Zigbee functionalities that can be used to form quick but simple Zigbee applications. SAPI contains also the configuration interface, which is used to read and write device and network specific settings on the CC2480 [35].

The AF and ZDO interfaces together provide a full Zigbee interface. The abbreviations refer to Zigbee stack layers Application Framework and Zigbee Device Object. The AF interface allows the host processor to send and receive data. The ZDO provides a range of management features for network formation, service discovery, and network management. [35]

| Value | Subsystem Name | Description |
|---|---|---|
| 1 | SYS interface | Low level interface to CC2480 hardware and software. |
| 4 | AF interface | Interface for registering applications and sending and receiving data. |
| 5 | ZDO interface | Interface for Zigbee management functionalities. |
| 6 | Simple API & Configuration interface | Interface for device and network parameter configuration and interface for simplified applications. |

Table 3: CC2480 Interface Command Subsystems [35]

### 5.5.2 Zigbee Software Interface

The Zigbee software interface on Ilmari platform is based on the transaction scenarios described in the CC2480 interface specification [35] and illustrated in Figures 21, 22, and 23. The interface is categorized in functional domains as illustrated in Figure 25. Host function calls provide the application with access to every request that is defined in the CC2480 interface specification. The SPI handler identifies different message types and handles SPI transactions accordingly.

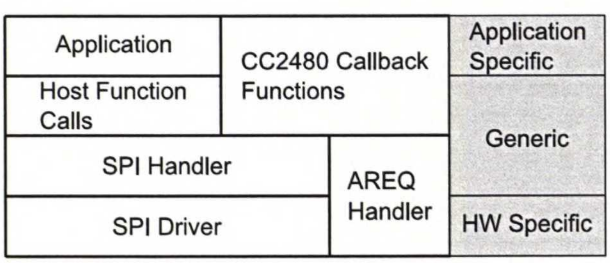


Figure 25: Zigbee Software Interface Architecture

The AREQ handler is partly a hardware specific software component that is capable of reacting on AREQ requests and receiving AREQ calls. CC2480 callback functions are a collection of procedures that are to be executed as a reaction to AREQ calls from the CC2480. Callback function behavior is defined by the application, thus the CC2480 callback function set is partly an application specific component. Software procedures for different messaging scenarios are presented in Figure 26 and Figure 27.

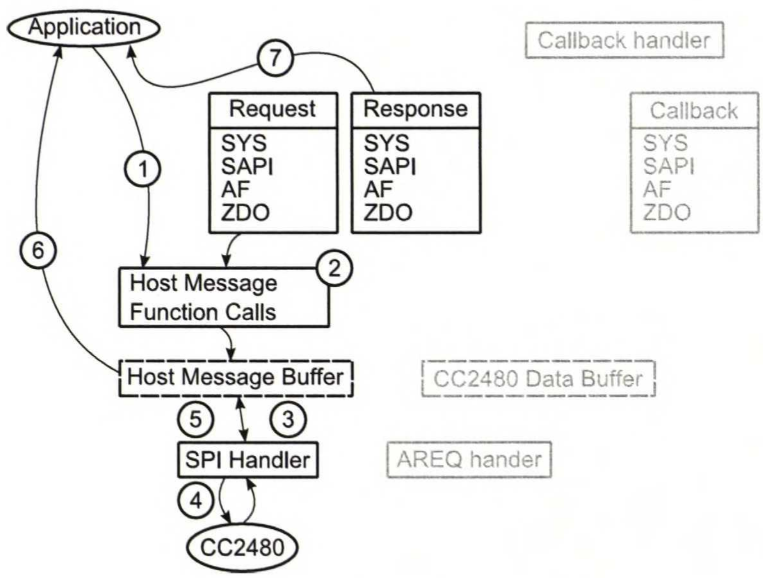


Figure 26: Sending SREQ, Receiving SRSP

Figure 26 shows how the application sends a SREQ message to the CC2480 and receives a SRSP message:

1. Application calls for a host message function.
2. The host message function call is formed according to subsystem and ID definition and packed into host message buffer.
3. Host message buffer is passed to SPI handler.
4. SPI handler concludes the message type and initiates SREQ procedure with the CC2480.
5. SPI hander receives SRSP from the CC2480 and packs it into host message buffer.
6. SRSP is passed to the application in host message buffer.
7. Application handles SRSP according to subsystem and ID definitions.

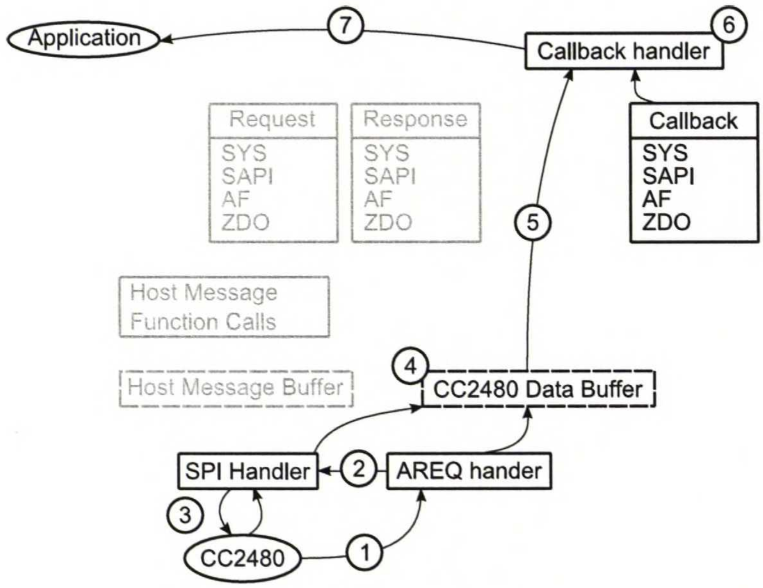


Figure 27: Receiving AREQ

Figure 27 shows how the CC2480 initiates an AREQ transaction with the host processor:

1. CC2480 indicates that it has a queued AREQ for the host processor by setting SRDY line low.
2. AREQ handler reacts to the hardware condition and initiates a POLL routine with SPI hander.
3. SPI handler receives AREQ data from the CC2480.
4. AREQ data is put into CC2480 data buffer.
5. AREQ handler passes CC2480 data buffer to callback handler.
6. Callback handler concludes the message type according to subsystem and ID definitions.
7. Appropriate callback function is executed and the results are passed to the application.

### 5.5.3 Microcontroller Unit Software Framework

The MCU's software framework, which provides basic utilities and support for peripheral devices, is built for open source WinAVR toolchain [37]. The framework is arranged so that applications written for the Ilmari platform can be ported to other hardware and software platforms. Figure 28 illustrates the platform's architecture. It should be noted that Ilmari platform does not use any microcontroller operating system.

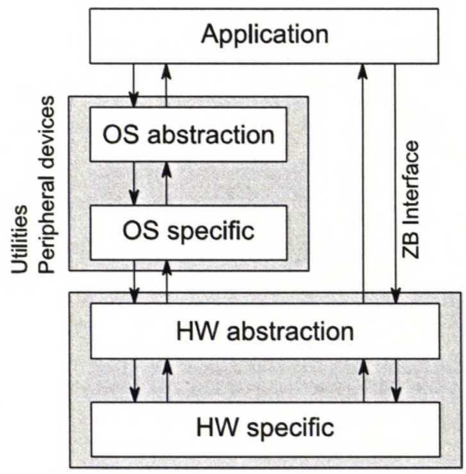


Figure 28: General Software Architecture of Ilmari Platform

Hardware (HW) specific components of the framework are designed to be used with Atmel's ATmega 164P/324P/644P microcontrollers. Basic hardware access is provided by WinAVR project's avr-lib-c. The hardware abstraction layer wraps basic functionalities, like low-power sleep states, behind function calls and makes it possible to port applications to other microcontrollers. The operating system (OS) specific components on the platform are peripheral device access, such as LEDs and buttons, and utilities, like debug printing, EEPROM access, and interrupt handling. Operating system abstraction layer wraps these functionalities behind function calls so that operating system specific parts can be re-programmed when changing to another platform, like a Linux PC.

The MCU's software framework is fairly light weight and does not offer any services or advanced functionalities, such as scheduling. These features can be obtained by introducing a microcontroller operating system. The framework is intended for prototyping and testing the platform's capabilities. A full software development kit would be outside the scope of this work.

## 5.6 Test Results

Ilmari platform was tested for two aspects seen as key characterizing factors in a wireless sensor network platform: power consumption and radio range. These tests were carried out to indicate the platform's capabilities in realistic use scenarios. In the following sections, test arrangements and results are presented.

### 5.6.1 Power Consumption Measurements

Power consumption measurements were carried out with an accurate digital multimeter that was set to measure the platform's input current as illustrated in Figure 29. The CC2480 module used in the tests is the folded dipole model shown in Figure 12 and the power source is the battery module with step-up converter shown in Figure 16. The components were set up as seen in Figure 17.

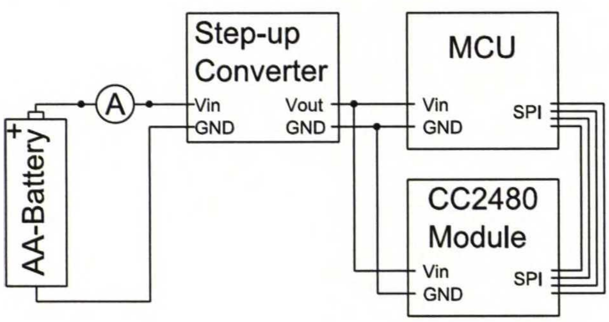


Figure 29: Current Consumption Measurement Arrangement

Two tests were carried out: Zigbee device in idle state and Zigbee device sending a payload of two bytes once every 10 seconds. The device that was used in the tests was set as a Zigbee end device. The MCU was set so that it would go into power-save mode when idle. Only task for the MCU, besides handling the Zigbee module, was to wake-up and update its real time counter once every second. The Zigbee module was set to poll messages from its parent every two second and poll for response messages from its parent after sending a data frame. Otherwise the Zigbee module would be in a power-save mode. Test conditions are summarized in Table 4

Test results were logged with a PC by using an infrared data link from the multimeter. Tests were let to run for two minutes and the results were reported as averages over two seconds. Results from the tests are shown in Table 5.

| | MCU |
|---|---|
| Operating voltage<br>Frequency<br>Wake-up interrupt source<br>Sleep mode | 2.7V<br>4.0MHz<br>Asynchronous timer @ 1Hz<br>Power save |
| | CC2480 Module |
| Operating voltage<br>Device type<br>Poll rate<br>Data sending rate<br>Packet payload | 2.7V<br>End device<br>1/2s<br>1/10s<br>2 bytes |
| | Battery |
| Voltage | 1.545V |
| | Environment |
| Temperature<br>Distance to receiver | 23.8°C<br>1m |

Table 4: Summary of Current Consumption Measurement Conditions

| Scenario | Current, Average (mA DC) | Current, High (mA DC) |
|---|---|---|
| Idle | 0.203 | 8.389 |
| Sending | 0.318 | 16.392 |

Table 5: Current Consumption Measurement Results

The battery that was used in the tests was a standard alkaline AA battery, which was measured to have voltage of 1.545V. By using this voltage, power consumption for the platform in test scenarios can be calculated by multiplying the measured current consumption with the battery's voltage. Power consumption figures are shown in Table 6.

| Scenario | Power, Average (mW) |
|---|---|
| Idle | 0.314 |
| Sending | 0.491 |

Table 6: Power Consumption of Ilmari Platform in Test Scenarios

Theoretical idle current consumption for the platform can be calculated by using information provided by component manufacturers. According to CC2480 datasheet [25], the chip will automatically enter low power mode in idle periods when configured as end device. In low power mode, the CC2480 will draw less than 0.5$\mu$A of current when operated with 3.0V, 25 °C. ATmega644PV datasheet [30] states that the microcontroller's power consumption, including a 32.768kHz oscillator, in power down mode is approximately 0.6$\mu$A when operated with 2.7V at 25 °C. According to MAX1724 datasheet [33], the step-up converter, that is used to convert the voltage of an AA battery to 2.7V, draws a quiescent current of 1.5$\mu$A at 25°C.

Using the above-mentioned current consumption characteristics, it is possible to calculate the theoretical minimum current consumption for the platform. Power consumption of MCU and CC2480 will stay constant, while MAX1724 will draw 1.5$\mu$A quiescent current at all times. The battery's voltage will range from 0.85 to 1.6V during its operating period [38]. Since power equals voltage times current, in which the voltage is the voltage of the battery, current consumptions for the system's individual components and the whole system are as illustrated in Table 7

| MCU | |
|---|---|
| @ 1.60V | 1.01$\mu$A |
| @ 0.85V | 1.91$\mu$A |
| CC2480 | |
| @ 1.60V | 0.94$\mu$A |
| @ 0.85V | 1.77$\mu$A |
| MAX1724 | |
| @ 1.60V | 1.5$\mu$A |
| @ 0.85V | 1.5$\mu$A |
| Total | |
| @ 1.60V | 3.5$\mu$A |
| @ 0.85V | 5.2$\mu$A |

Table 7: Theoretical Minimum Current Consumptions

Assuming linearity of battery's voltage drop, the average minimum current drain from the battery, during the whole use period, can be concluded to be around 4$\mu$A. With current consumption this low, the platform could be in a sleep state, waiting for an external interrupt for periods many times longer than an average battery's shelf life. However, the measurements show current consumption in the practical idle scenario to be 203$\mu$A on average.

The current consumption figures are much higher in a practical scenario, like in the test setting that was used, because even in the idle state, the MCU must wake-up from power save mode every second to perform a small task and the CC2480 must also wake-up every two seconds to poll for messages from its parent. When the CC2480 polls for messages, it will have both RX and TX sections of the radio operational for short periods of time. The current consumption gets even higher when the CC2480 is sending messages to its host.

According to Duracel's technical bulletin for alkaline-manganese dioxide batteries [38], an alkaline AA battery can have capacity in the range of 2800mAh. Using this figure as the given battery capacity, operating times for the test scenarios can be calculated by dividing the capacity with the measured average currents. The whole capacity of the battery can be assumed to usable because a low drop-out step-up converter is used power the system. Operating times for a system with two batteries connected in parallel can be obtained by assuming double capacity. When using two batteries, it is recommendable to still use the step-up converter, since two AA batteries in series would provide a voltage that would decrease over time from 3.2V to 1.7V and thus series connected batteries would not provide optimal operating conditions throughout the batteries life-time. The results of these calculations are shown in Table 8.

| 1 Battery | Days | Years |
|---|---|---|
| Idle | 574.7 | 1.57 |
| Sending | 366.9 | 1.01 |

| 2 Batteries in parallel | Days | Years |
|---|---|---|
| Idle | 1149.4 | 3.1 |
| Sending | 733.8 | 2.0 |

Table 8: Operating Times for Test Scenarios, Using 2800mAh AA Batteries

Results from Ilmari platform's current consumption measurements showed that the actual current consumption of the platform is low. The platform performs well as a low-power Zigbee end device. These results, however, do not show current consumption measurements for any peripheral device, such as a sensor element. To give a broad sense of a sensor's effect on the platform's power consumption performance, Panasonic's Napion low-power digital PIR sensor will consume typically 46$\mu$A at 2.2 - 3.0V amounting to 0.101 - 0.138mW [34]. This will in turn add up to operating time of roughly one to two years with a system that uses two AA-batteries and operates according to test scenarios. In reality, a practical system that uses such sensor elements could be in sleep states most of the time and only wake up if the sensor is sensing something that needs to be reported. Thus the actual battery life would be more than one year.

### 5.6.2 Radio Range Measurements

A test arrangement was devised to give general knowledge about the radio performance of the two Zigbee modules used in Ilmari platform; the Radiocrafts RC2300-ZNM (ZNM) module and the folded dipole (FD) module. Tests were carried out in the hallway of a large commercial building with two Zigbee nodes: a mobile end device and a stationary coordinator. The coordinator was connected to a PC with UART.

The measured variable in the tests is link quality indicator (LQI), which is the only indicator available on the CC2480 for estimating radio reception quality. According to CC2480 interface specification [35], LQI readings are obtained with every data frame that is sent using AF_DATA_REQUEST command and received with AF_INCOMING_MSG callback. IEEE 802.15.4-2006 standard [10] defines LQI as follows: The LQI measurement is a characterization of the strength and/or quality of a received packet, which can be implemented using receiver energy detection, signal-to-noise ratio, or a combination of these methods. The LQI measurement shall be reported as an integer ranging from 0x00 to 0xFF. The minimum and maximum LQI values should be associated with the lowest and highest quality compliant signals detectable by the receiver, and LQI values in between should be uniformly distributed between these two limits, using at least eight unique values.

The test plan is as follows: The coordinator will be stationary at all times. The two devices shall form a network in which the end device can send data to the coordinator. The end device will be moved away from the coordinator in intervals of one meter. The two devices shall be directed so that the antennas point directly at each other in a way that is expected to be the best case for radio reception, even though both antennas are expected to be fairly omnidirectional. At each interval, the end device sends three frames, containing two bytes of payload data, to the coordinator. Link quality indicators, provided by the Zigbee stack, are recorded from the coordinator for later analysis. The test is carried out until no data is being received for three consecutive distances or a distance of 30 meters is reached. The test will be run four times with equipment setups illustrated in Table 9.

| Test no. | Coordinator | End device |
|---|---|---|
| 1 | Folded dipole | Folded dipole |
| 2 | Folded dipole | RC2300-ZNM |
| 3 | RC2300-ZNM | Folded dipole |
| 4 | RC2300-ZNM | RC2300-ZNM |

Table 9: Equipment Setups For Radio Range Measurements

Test hypothesis was that both radio modules in every combination will perform acceptably over the range of 10 meters. The test arrangement for radio range measurements is illustrated in Figure 30.

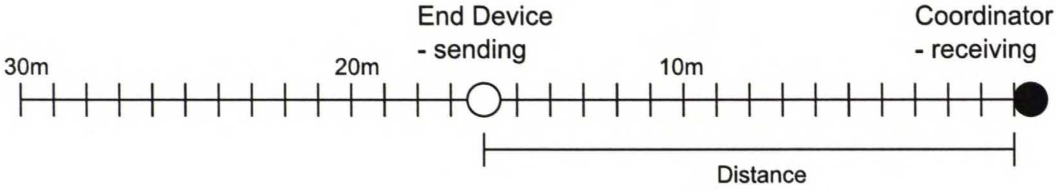


Figure 30: Range Measurement Arrangement

The tests were carried out in an open hallway illustrated in Figure 31. The hallway was seen as an environment, that would have interference comparable to that of a real-life use cases; The hallway has got multiple WiFi routers operating at the same 2.4GHz frequency band as Zigbee and there are no uniform contours that would provide a reflective surface for radio waves. Both devices had clear line-of-sight during the tests.

Figure 31: Range Measurement Environment

Figure 32 illustrates the orientation of the devices during the tests. Each device, in every test setup, was directed in the same manner. The radios were elevated 18cm above the floor surface using empty cardboard boxes that would provide no radio interference.


Figure 32: Range Measurement Radio Placement and Orientation

A total of 360 individual data frames was sent; 1 meter intervals from 1 to 30 meters, three frames per interval, with four different radio module configurations. The LQI data, that was gathered from the tests, is illustrated in Figure 33. Each successfully received data point is plotted with a green marker. The black line indicates averages of the LQI readings. Red markers indicate that the message was not received. Blue circles indicate messages that were received successfully but from which the acknowledgment frame was not received by the end device. The red line in the figure indicates a level that is considered to be an acceptable LQI reading for normal operation.

The test results show that both radio modules are usable with ranges of 15 meters. The radio performance of the modules exceeded expectations that were set in the hypothesis. In Figure 33, it can be seen how the LQI is affected by the environment. Changing contours along the direct flight path of the radio signal will in some cases cause distractions in the reception and in some cases allow good reception, possibly due to favorable reflections.

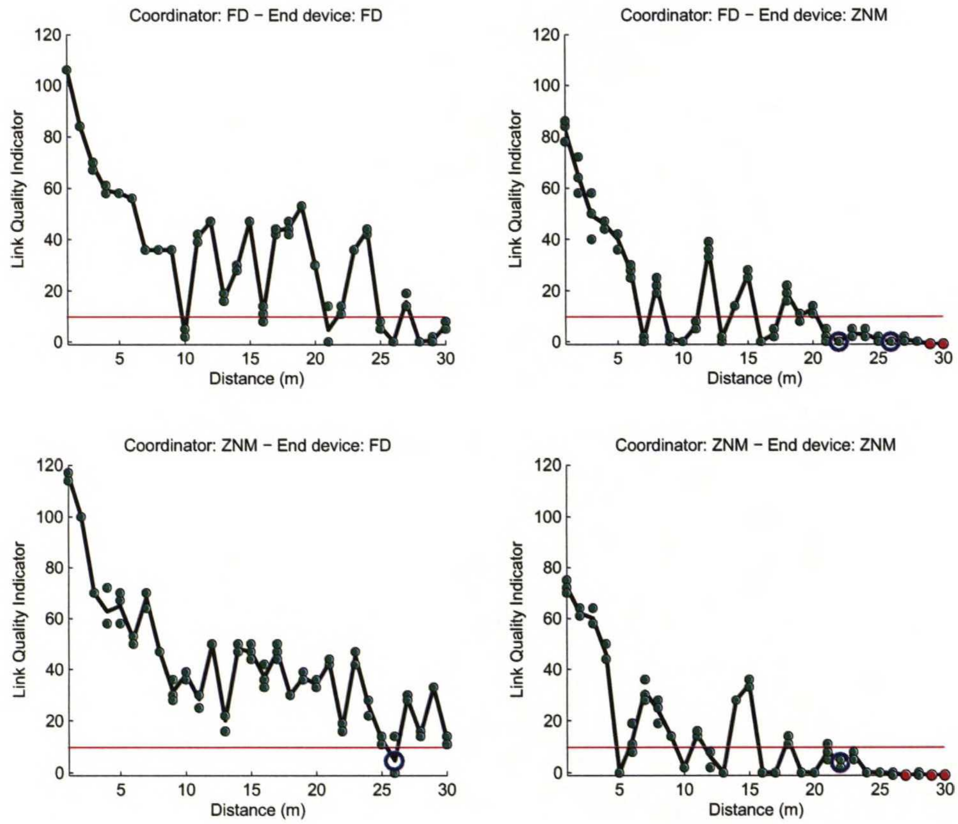


Figure 33: Range Measurement Test Results

It can be also noted from Figure 33, that Radiocrafts' RC2300-ZNM module had some performance issues while operated as the end device. It is assumed that the alignment of the ZNM module was not optimal for transmitting. Texas instruments application note on antenna design [28] provides radiation patters for the folded dipole antenna design that was used as the basis for the FD module. This data was used to determine the orientation of the devices during the tests. Because similar data was not available from Radiocrafts, the ZNM module's orientation was the same as for the FD module.

# 6 Applications

In this section, application development on the Ilmari platform and foreseeable applications that could be implemented using the platform are discussed. Also, a practical assistive automation application implemented as a part of this work is presented.

## 6.1 Application Implementation

It is safe to say that the software framework of Ilmari platform is the least finalized part of this work. This means that, at the moment, there are no strictly defined way of developing applications for the platform, i.e., no SDK or even an operating system is available. There is, however, an example project that provides an initial application framework that can be used as the basis of future development.

It is clear that a full Zigbee-AVR SDK would be outside the scope of this work. Also, imposing too much code overhead in the form of an operating system would not benefit the platform as an prototyping environment. It would simply force any future developer to adapt, in addition to Zigbee, new software concepts and in extreme cases, even to adapt a whole new programming paradigm.

This subsection will cover practical application implementation related topics. First, tools used for application development are introduced. Second, the sample application is presented, followed by discussion on some key application development aspects that are not covered in the sample application. Last, application interoperability and integration with a home server environment are elaborated on.

### 6.1.1 Development Environment

Atmel's AVR microcontrollers are mostly programmed using C programming language. GCC has good support for the AVR line, through WinAVR project. WinAVR also provides other necessary tools for application development, such as avrdude (programmer) and avr-gdb (debugger).

The standard application development environment (IDE) for Atmel's AVR processors is the AVR Studio. It was found out, however, that AVR Studio does not provide necessary tools for maintaining larger projects. Instead of AVR Studio, Eclipse C/C++ Development Tooling (Eclipse CDT) with AVR plug-in is recommended. Eclipse will allow the use of integrated version control systems, like Subversion (SVN), and what is best, Eclipse allows for multiple build configurations.

This feature is essential, when developing and testing software for multiple WSN devices at the same time. It is not uncommon that the developer will want to compile and program software for an end device, a coordinator, and a router at the same time.

The AVR plug-in for Eclipse provides the following features: support for different target AVR processors, creating hex files that can be used to program an AVR processor, and support for the avrdude programmer that is used to transfer a project to a MCU. In addition to right software environment, an ISP tool is needed. Tools like the AVR ISP or STK500 can be used for programming. In addition to ISP, the STK500 provides an UART to RS232 converter for communicating with a PC.

### 6.1.2 Sample Application

The Ilmari platform sample project is a bare-minimum application that covers every crucial part of Zigbee application development, excluding binding. The sample application uses every Zigbee device type: end device, coordinator, and router. The coordinator forms a Zigbee network, selects the best available channel, and sets a random PAN ID. The application uses a proprietary profile with profile ID 0XC035. This profile will have only one cluster: LED cluster with ID 0x0001. The LED cluster has a message format that is simple and straight forward, as shown in Table 10.

| | Frame | |
|---|---|---|
| Byte: | LED_ID | Action |

| Values | |
|---|---|
| LED_ID: | 0x00 - Red<br>0x01 - Green<br>0x02 - Both |
| Action: | 0x00 - Off<br>0x01 - On<br>0x02 - Toggle |

Table 10: LED Cluster Message Format

Only network addressing is used to send messages in the application. Each device will act on a button push by sending a LED cluster command. An end device will send red LED toggle command to the coordinator with network address 0x0000. The coordinator will send red LED toggle command to every router with broadcast address 0xFFFC (broadcast to routers and coordinators). A router will send green LED toggle command to the coordinator with network address 0x0000. No binding is used in the application; in a Zigbee network, the coordinator has always network address 0x0000.

The sample application's execution routine is illustrated in Figure 34. After the system is reset, various hardware initializations are executed. The system's state is kept in non-volatile memory, which enables the application to carry on from its previous operating state after power-off. This means that devices can be restarted without having to go through lengthy initialization, such as network re-formation, and the network does not suffer if devices go offline at any given moment. Devices simply carry on operating as they were before power-off.

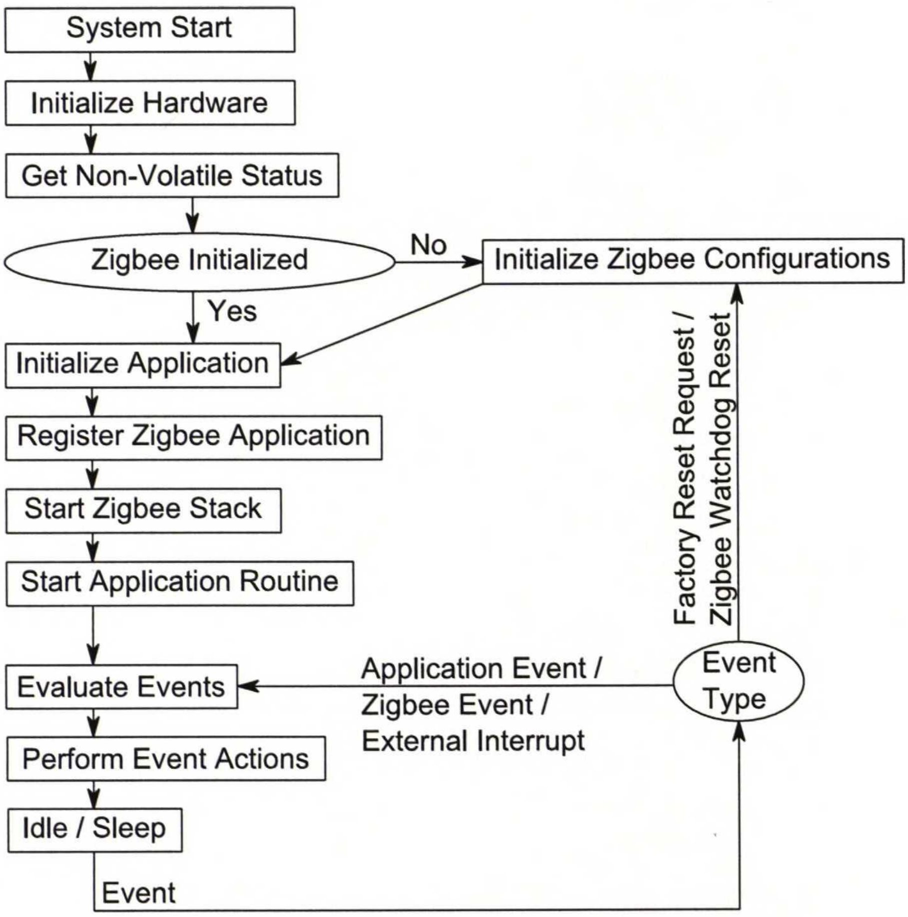


Figure 34: Sample Application Execution Routine

Each time a device is restarted, however, the application itself has to register to the CC2480's Zigbee stack. This procedure will initialize the Zigbee stack - providing information on supported profile, clusters, etc. - before it is started with a specific start request command. After the stack is started, the application can allow other devices to join the network through the device, when operating as router of coordinator. Joining can be allowed either indefinitely or for a certain time period. If a device is associated with a network that has joining turned off, and the device is suddenly powered off, the device can still re-join its network by using existing configurations that are stored on non-volatile memory. This type of situation can occur, for example, when batteries are being replaced from an end device.

When configurations and initializations are done, the application can start to execute its routine. In the sample application, the application flow is controlled by events. If no events are available, the device will go into a sleep or idle state, depending on the device type. If a new event occurs, i.e., an interrupt is triggered, the host processor will wake up to evaluate the event. The CC2480 operates in the idle state without intervention from the host processor.

For every device in the sample application, there is a factory reset mechanism, that will wipe out any existing network and device information, leaving the device in an uninitialized state. The factory reset is performed by long-pressing the MCU's button. After pressing for two seconds, both LEDs will turn on. When the button is released, the device will start initializations as illustrated in Figure 34. This functionality will allow the developer to force re-formation of the Zigbee network.

### 6.1.3 Advanced Features

Binding is an essential feature of Zigbee. They provide means for the application to communicate with other Zigbee devices without knowing their network addresses. The sample application does not use binding, because bindings require either complicated service discovery mechanisms or user action to form.

Zigbee device object (ZDO) is an application that resides on the CC2480, or more precisely in the Zigbee stack. ZDO provides many useful functionalities for service discovery and network maintenance through the ZDO interface, described in the CC2480 interface specification [35]. Interesting thing about ZDO is that it acts almost like any other Zigbee application. It can be accessed over the network from other devices to request for information or action. The ZDO provides a mechanism to bind Zigbee clusters on any device to any other device with matching clusters through any other device. Binding with ZDO is illustrated in Figure 35.

In Figure 35, ZDO on router A is used to form a binding for cluster 1 between devices B and C. First, A sends a bind request to B, telling it to bind with C. The bind request carries information on the target device and the cluster that is being bound. After receiving a valid request, ZDO on device B handles the request by forming a binding and responds to the bind request with a bind response. The bind response

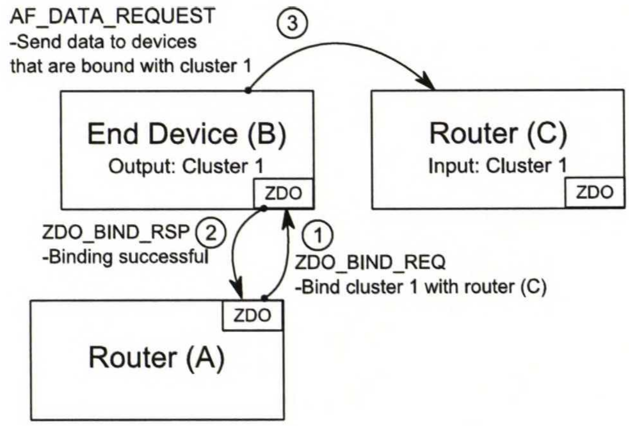


Figure 35: Zigbee Device Object (ZDO) Bind

message indicates the result of the bind attempt. If the binding is successful, B can now send cluster 1 messages to C without knowing C's network address. Sending messages to a bound device is done by using a special network address, the indirect network address 0xFFFE.

In the above described scenario, ZDO binding allows router A to act as a network commissioning tool. This type of binding can enable a more complex node in the network to first discover services from each device through ZDO and then bind appropriate devices to each other. It should be noted that bindings are unidirectional, cluster specific, and require that both devices have registered appropriate input and output clusters. Powerful aspect of bindings is that they can be formed from one source to multiple targets, i.e., if a device has a output cluster that is bound to multiple targets, sending a message with that cluster to the indirect network address 0xFFFE, the message would be delivered to each of the target devices.

Figure 36 illustrates another binding scheme, end device binding. This type of binding can be used in less complex networks. The idea is that end device binding is initiated on both devices by some sort of user action, like a button push. In end device bind, both devices send a listing of their input and output clusters to the coordinator, which will determine if matching clusters are available. If matches are found, the coordinator will use zdo binding to form bindings from output clusters to matching input clusters. If end device binding is successful, devices can send messages to each other by using the indirect network address.

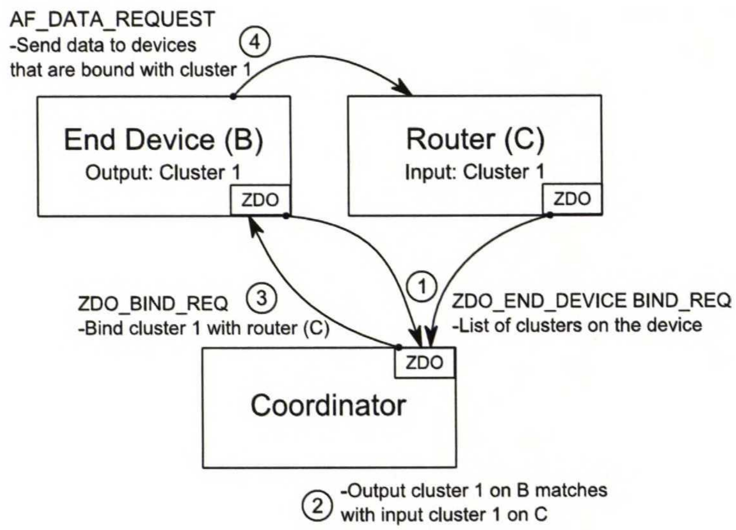


Figure 36: Zigbee Device Object (ZDO) End Device Bind

### 6.1.4 Interoperability

Zigbee cluster library (ZCL) provides interoperability for certain types of Zigbee applications. However, ZCL is part of the Zigbee application and in Ilmari platform, the application resides on the MCU's host processor. This means that in order to achieve interoperability features, described in the ZCL [16], a ZCL application would have to be implemented on the host processor. When considering the vastness and functional requirements of the ZCL, implementing it is not a trivial task. This means that using proprietary application profiles, instead of ZCL defined public profiles, is far more attractive in prototype systems.

With Ilmari platform, interacting with a commercial public profile device could still be possible without the full implementation of ZCL. The platform would just have to be configured to operate with the proper profile ID, channel, and PAN ID as defined in the public profile. After initialization, proper cluster IDs would be registered on the CC2480. Since ZCL messages are passed as the payload of regular Zigbee frames, the application can easily decompose ZCL messages and interpret them. This allows the use of a simple public profile device, such as an on/off switch, without the implementation of the whole ZCL. This would, of course, not be acceptable in commercial systems, but only in prototypes.

### 6.1.5 Integration with Home Server Environment

Another approach to interoperability is the use of a home server of similar infrastructure to connect Zigbee networks that operate with different profiles. A home server could operate on both profiles and aggregate data from both networks. The server could then use the gathered data, no matter from which network it was originated, and use it to control devices on both networks. In this type of setup, the server would be the only device that understands every message. The server could also operate as a gateway to other networks, like the Internet.

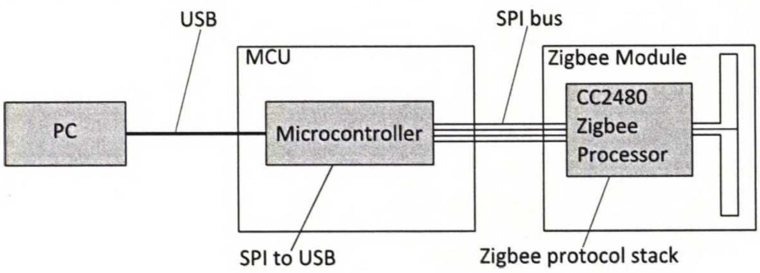


Figure 37: Connecting CC2480 to a PC

Figure 37 shows how a CC2480 module could be connected with a PC. The Zigbee stack would operate on the CC2480 and the MCU would provide the PC with an USB interface for the CC2480. Ilmari platform's software has been designed for this type of integration. Applications that are written for the AVR MCU are easily portable to a Linux environment.

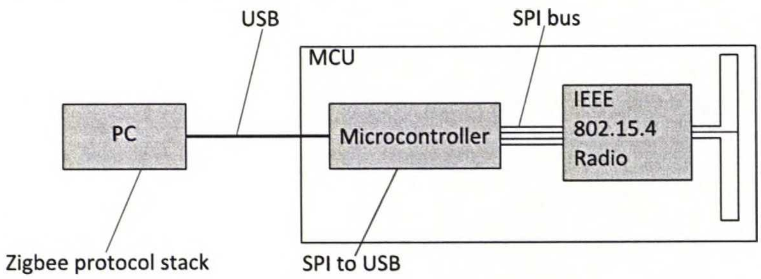


Figure 38: PC with IEEE 802.15.4 Radio

Figure 38 shows a setup in which the PC controls an IEEE 802.15.4 radio directly. In this case, the PC would have to operate the Zigbee stack. This type of solution would bring Zigbee into the PC and offer greater flexibility by providing full access to the Zigbee stack. Given the relative novelty of Zigbee technology, this type of solutions are not yet widely available. In the longer term, however, it is expected that this type of solution will become more common, as it is in the case of WiFi and Bluetooth.

## 6.2 Foreseeable Applications

Figure 39 illustrates applications within the scope of assistive automation that are envisioned to be implemented with Ilmari platform. The apartment in the figure has multiple assistive automation applications operating in the same Zigbee network.

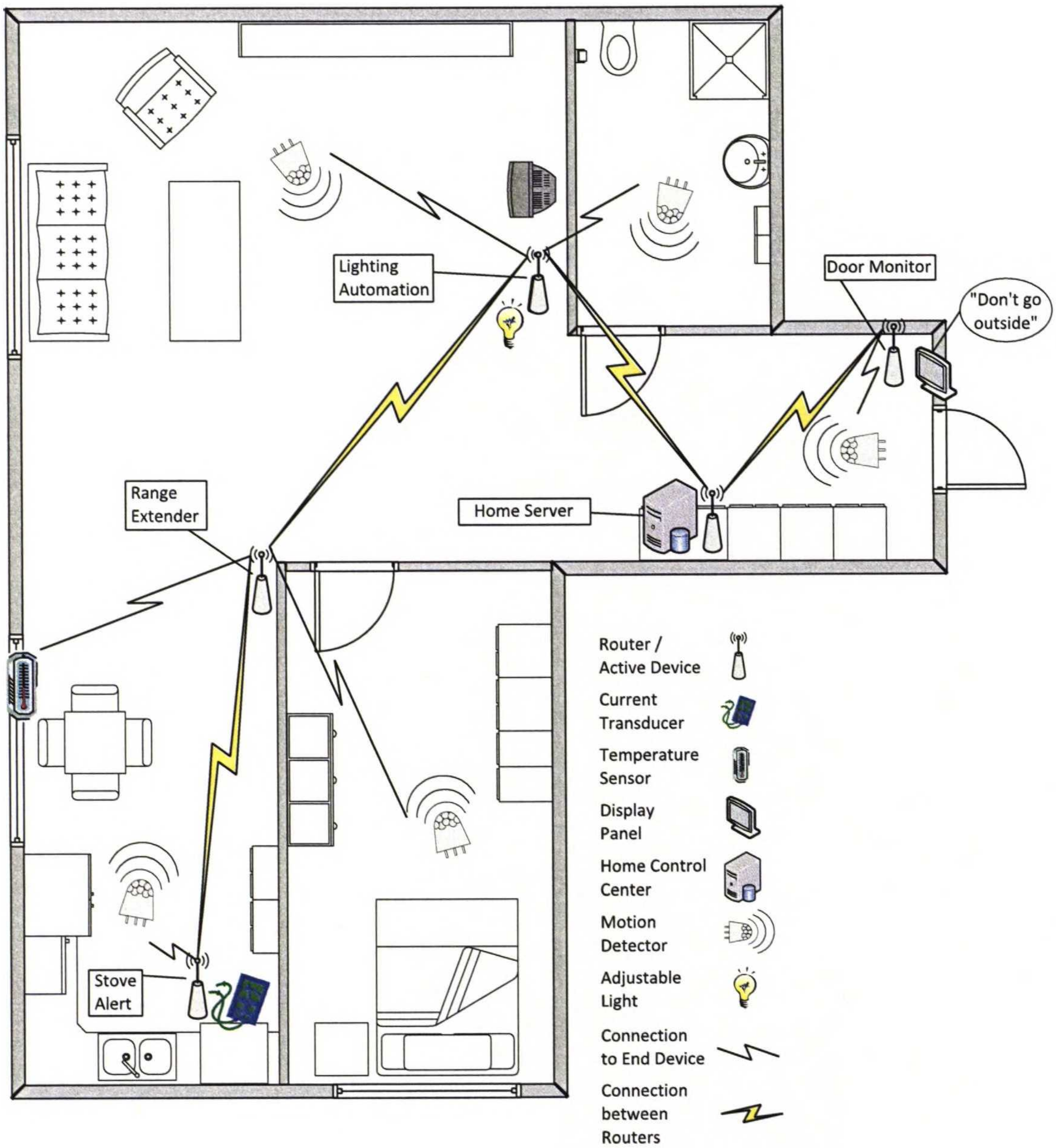


Figure 39: Installation Proposal For an Assistive Automation WSN System

A Zigbee router is used in every device that needs to be stationary and mains powered, such as a lighting automation application's mains dimmer unit. These devices have plentiful power supply, which means that they will form the infrastructure of the Zigbee network. The Zigbee coordinator will be attached to a home server that will act as a gateway to the Internet, making all sensor information available through a web interface. The server can also be used to configure and commission the network and to periodically observe the network's status.

Applications that are depicted in Figure 39 are as follows: lighting automation, door monitor, and stove alert. In addition, a general activity monitoring application could be implemented by using the home server. The activity monitoring application would collect data from every node within the apartment into a database. This information could then be used to evaluate long-term trends in the use of the apartment, such as visits to the bathroom or kitchen per day and time spent in the bedroom per night. Also, when combining real-time information from every sensor in the apartment, short-term alerts could be implemented. E.g. the system could notice an incident where a person falls down in the bathroom by drawing simple conclusions from motion detector data. If the last observation by a motion detector was one hour ago from the bathroom after which no other observations were obtained, it could be concluded that the person might be in need of some form of assistance.

Other example of the system's potential is an application that would warn the resident if he/she is leaving the apartment while the stove is on. The display unit next to the door could be used to show a warning text or to give an alarming sound when the door is being opened and the stove is turned on. This goes to show that by enabling data interchange within the WSN, new types of applications and features could be implemented with relatively small effort.

## 6.3 Door Monitor

The door monitor is an application concept that was first devised in the Automation Technology laboratory of Helsinki University of Technology. The application is intended to be installed into homes of elderly people who suffer from a moderate memory disorder. The main objective of the door monitor system is to prevent situations where an elderly person would become disoriented and wander outside at night. However, if such an event would occur, the system will alert caretakers in real time.

### 6.3.1 Concept

A door monitor application demo was implemented as a part of this work to demonstrate the potential of Ilmari platform. Figure 40 illustrates the application's basic concepts.

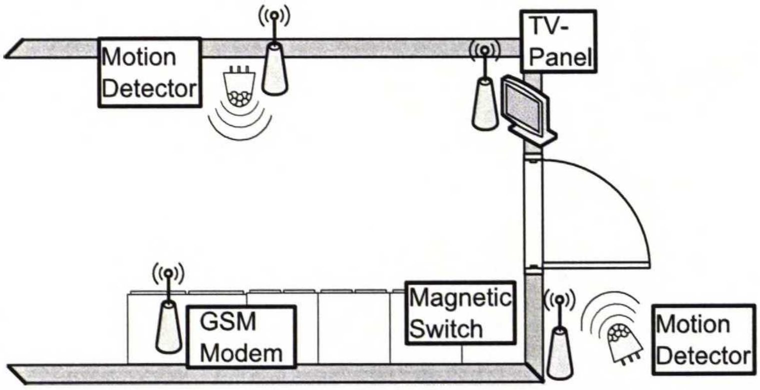


Figure 40: Door Monitor Concept

The system uses three sensors: one magnetic switch and two motion detectors. The magnetic switch is placed so that it will detect if the front door is opened. Motion detectors should be placed so that one detector can detect if a person is moving toward the front door and the other detector can detect that a person is moving outside the front door.

In addition to the sensors, there are two active components: a TV-panel and a GSM modem. The TV-panel is used to give out warnings if a person is approaching the front door at night time. The same TV-panel can also be used to sound audio alarms, thus adding a redundancy into the warnings. The GSM modem is used to send alarm messages to caretakers. Also, the GSM modem can be used to acquire time information into the system.

Selection of Zigbee device types and grouping different functionalities into devices can be done in multiple different ways. In the demo application, the GSM modem is controlled by the coordinator, the TV-panel is controller by a router, indoor motion detector and magnetic switch are handled by an end device, and the outside motion detector is handled by an end device as well. This type of grouping will allow the application to demonstrate every Zigbee device type.

### 6.3.2 Hardware

Figure 41 illustrates door monitor's hardware components. The figure also shows Zigbee clusters that are used in the system. In order to make battery operated devices even more energy efficient, optional time-of-day clusters could be implemented. These clusters would tell end devices when they need to report motion detector data and when they can sleep for longer periods, without reporting any data. Also, end devices could implement a ping cluster for added reliability. End devices would send ping commands back to a host device every thirty minutes or so. This way the system could alert caretakers if one of the sensors would go offline.

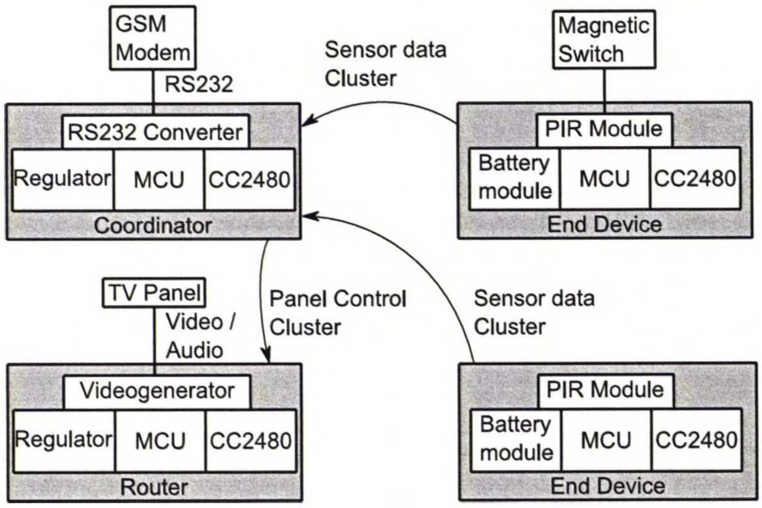


Figure 41: Door Monitor Hardware Components and Clusters

Figure 42 shows a video generator unit for door monitor. This unit is designed to generate black and white PAL video signal with 240 (width) x 115 (height) pixels. The video generator has an ATmega644PV processor which runs at 20MHz. The processor has two USART units from which one is used to communicate with a host processor and the other is used in SPI mode to generate the video signal. By using USART to generate the video signal, the microcontroller can utilize the output buffer to generate the video signal in sets of eight pixels.

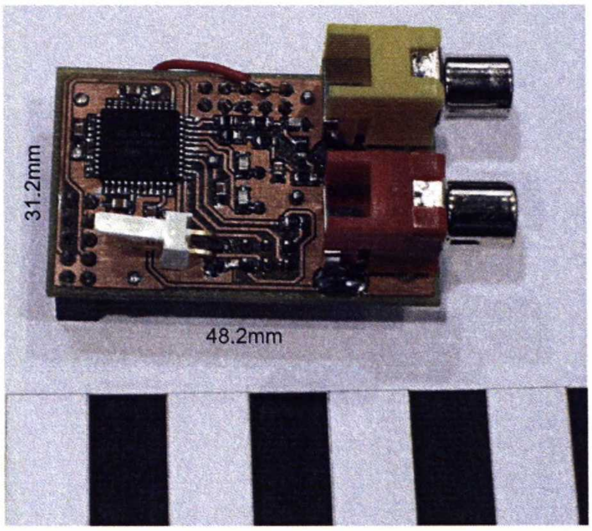


Figure 42: Video Generator for Door Monitor; centimeter scale

Figure 43 illustrates how the video generator microcontroller is used to generate three necessary signal levels that needed for black and white PAL video signal. The advantages of using this type of setup to generate video in the door monitor application are as follows:

- Analog video signal can be easily scaled to different sized screens.
- Almost every inexpensive display panel has a composite video input.
- The video generator can output both simple graphics and bitmap fonts.

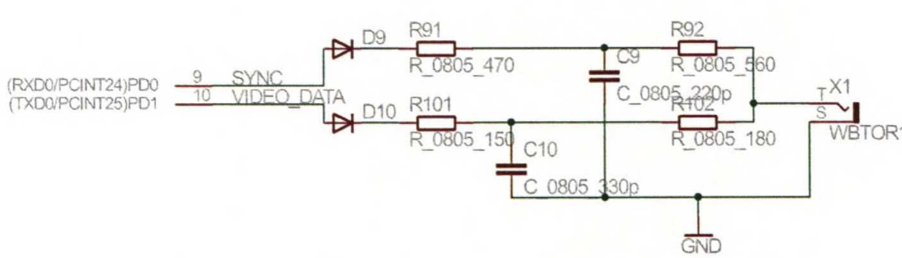


Figure 43: Video Generator Signal Schematic

Figure 44 shows the small display panel that is used in the demo application. The panel is an inexpensive 8.4" TFT LCD TV with integrated digital TV support. In the figure, the panel is showing a bitmap image of the author. This goes to show that the video generator can be even used to produce crude but recognizable images. The same display panel can also be used to produce 8-bit sound effects by using the microcontroller's PWM output. This feature is used in the door monitor application to generate a warning sound.

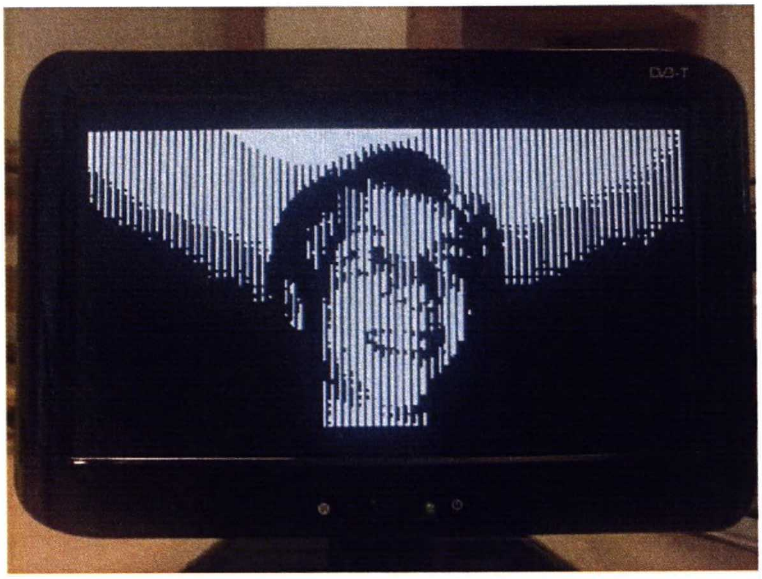

Figure 44: Display Panel for Door Monitor

Coordinator of the door monitor application is used to control a GSM modem. In order to control the modem with AT commands, the coordinator MCU has to be fitted with a RS232 converter module. This module is illustrated in Figure 45.

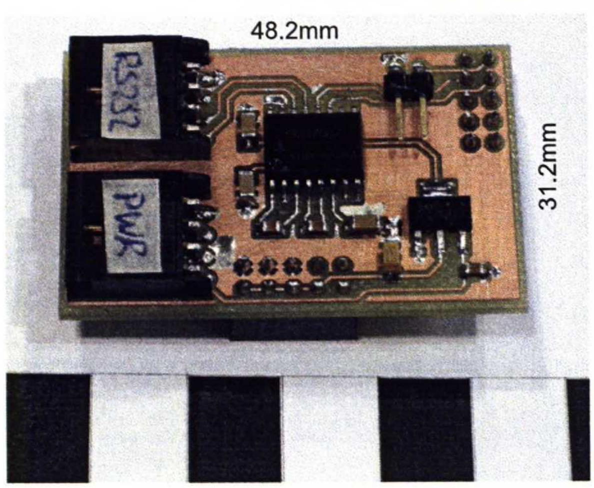


Figure 45: RS232 Converter for Door Monitor; centimeter scale

Other hardware that is used in the door monitor demo are as follows: battery step-up converter modules, PIR sensor modules, which both are described in Section 5.4, and a standard 2-pin magnetic switch. The system assembly with enclosures is illustrated in Figure 46.

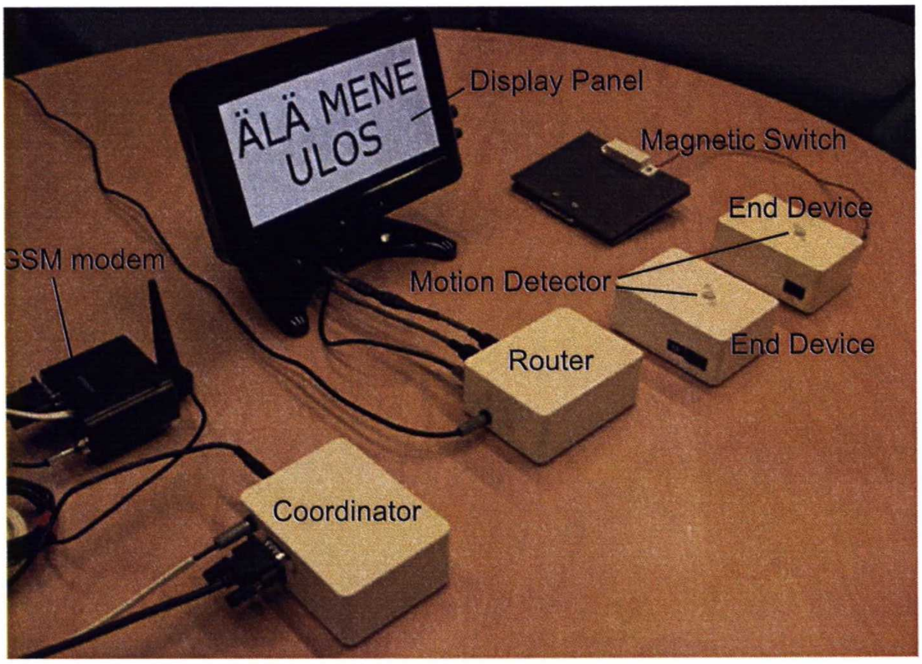


Figure 46: Finished door monitor demo assembly

### 6.3.3 Software

The Zigbee coordinator handles the main logic of the door monitor application. Motion detectors are configured to send data of detected motion at maximum of once every two seconds when new motion is detected. The magnetic switch is set to be reported instantly, when changes in its state occur. The display panel unit is only used to extend the range of end devices by acting as a router and to control the display panel according to incoming display data messages.

The application's main logic is illustrated in Figure 47. Functionalities are formed by a state machine running on the coordinator MCU. A state machine representation was seen as a good way to form the door monitor's functionalities. As seen in the figure, every state in the state machine has functions that are executed when entering or exiting a certain state. Texts along with state transition arrows indicate rules that must be met in order for the state transition to take place.

Neutral state is the default state of the system. In this state, only an empty black screen is shown on the display panel. When motion is detected inside close to the front door at night time, transition to visual alarm state is executed. In visual alarm state, display panel shows a text that is requesting the resident to stay inside. Upon entering visual alarm state, a timer is started. If the timer expires before front door is opened or new motion is detected inside, the application will transition back to

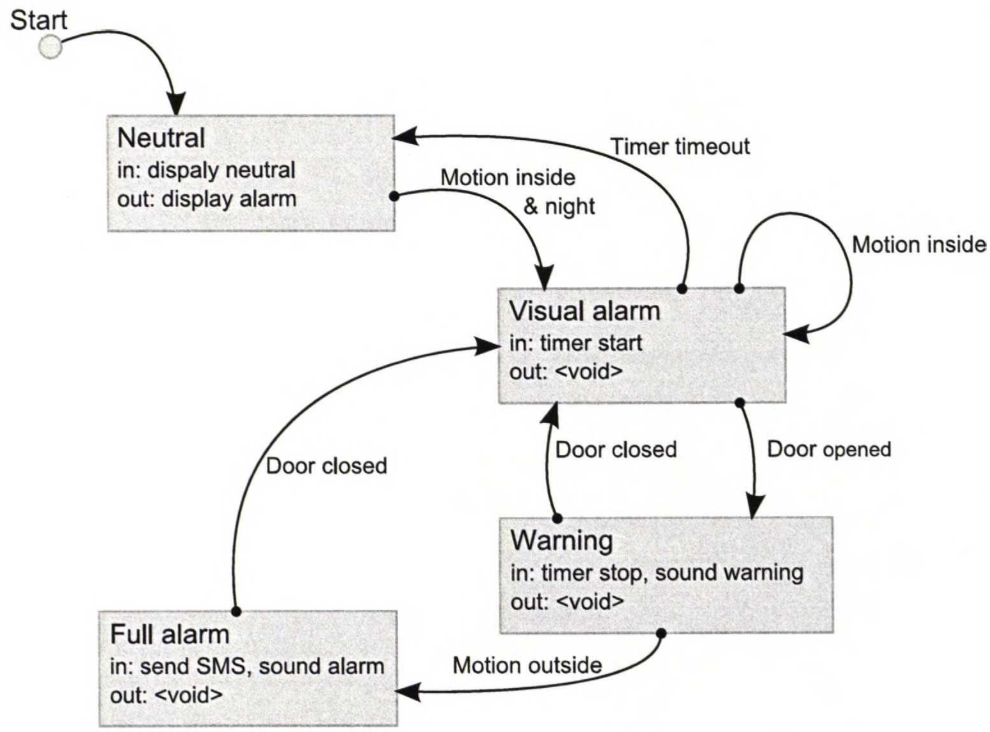


Figure 47: Door monitor State Machine

neutral state.

If front door is opened while the application is in visual alarm state, a transition to warning state is executed. When entering to warning state, an audio warning is sounded with the display panel's speakers. The system stays in warning state until the front door is closed or motion takes place outside the apartment. If the front door is closed, new timeout period in visual alarm state is started.

Transition to full alarm state is executed if the person has moved inside the house at night, opened the front door, and motion is detected outside the house. In this state, the system will send an SMS message to a predetermined phone number and give out a loud alarm sound. Full alarm state is exited when the front door is closed. After exiting full alarm state the system will first go into visual alarm state and after a timeout period, back to neutral state. The SMS message could be dynamically formed to e.g. include time and date, but only a static message is used in the demo application. Figure 48 illustrates an SMS message sent by the application.


Figure 48: Door Monitor Alarm Message

### 6.3.4 Results and Future Development Needs

The demo applications works reliably and fills the purpose of exhibiting Ilmari platform's capabilities. It proves that such an application can be implemented with the platform but also that the door monitor application genuinely benefits from the use of a wireless sensor network. The demo setup, although being just the first draft of the system, is elegant enough to be installed with relatively small amount of effort. Also, the hardware that is used in the system is compact enough so that it would not stand out too much in private home surroundings.

The door monitor application that was developed is only a demo in the sense that it only operates in night mode and without advanced power saving and reliability features. At its current form the application does not handle changes in time of day. If a door monitor application would be installed into an actual use environment for pilot testing, further development would still be needed. The system would need a human interface for inputting time information or the GSM modem could be used to acquire time information. This can be done in the simplest form by sending a SMS to the device itself. The incoming message will contain time information from the GSM network.

Also, in a practical pilot system, there is no need for as distributed system as the demo application is. Functionalities could be re-arranged so that only one device is needed to handle both the display panel and the GSM modem. Also, a server PC could be used to provide a remote user interface for the system for debugging, commissioning, and statistical usage data collection.

# 7 Conclusions

In this thesis, a wireless sensor network platform for assistive automation applications is presented. The result of this work is the Ilmari platform; a low-power Zigbee based modular WSN prototyping platform. Test results show that the platform is capable of operating multiple years with single AA-battery and that the platform has wireless range capabilities of minimum 15 meters in practical operating conditions. This thesis also presents a demo application for Ilmari platform, the door monitor.

Zigbee, that was chosen for Ilmari platform's standard, is proved in this work to be usable in practical applications and excellent choice for assistive automation applications. In this last sections, results from this thesis are summarized. Also, evaluation of the results is presented and future development needs regarding wireless sensor networks in assistive automation are discussed.

## 7.1 Summary of Results

Ilmari platform was developed according to initial requirement, given in Section 5.1. These requirements were met almost completely. Ilmari platform can be used to develop stand alone applications as well as applications that require the availability of home server infrastructure. Applications are easy to install, since in most stand-alone applications there is no need for device association or special network commissioning. If device associations are needed in a more complex application, a button push procedure can be implemented to form binding between devices. Also, automated service discovery and binding is possible, but because of its complexity, automated network commissioning is best to be left done by a device with more plentiful computational capabilities than a 8-bit microcontroller.

Ilmari platform devices are small in size. Even smaller devices can be built with the same components, but miniaturization was not seen to be a central topic in prototype applications. Also, Zigbee networks enables the use of mobile end device, which was one of the optional requirements for the platform. One optional requirement that is not met by the current Ilmari platform is the over-the-air (OTA) firmware updates. It would be possible to create a bootloader that would enable OTA firmware updates, but it would require considerable development effort and consume additional hardware resources. Therefore, the OTA bootloader was left unimplemented.

Design considerations that are given in Section 5.2 are as follows: The platform has to be easy to develop on, have long battery life, have inexpensive hardware, have inexpensive development tools, and to be highly modular. It can be said that every one of these requirements is met with the Ilmari platform. The platform can be considered easier to develop on than a system-on-chip solution, since there is no direct interaction with the Zigbee protocol stack. The hardware that was developed is extremely modular, since even the power supply is not directly included on the MCU. Development tools that are used to create applications for the platform are free. The battery life of the platform is proved to be extremely good in real-life use case scenarios. The only thing in the given requirement list that can considered to be slightly arguable is the cost of the hardware; The hardware is inexpensive component-wise, but since it is not being mass-produced, unit costs are relatively high.

## 7.2 Evaluation

Ilmari platform performs according to initial requirements. Individual sensors can be re-used in multiple applications across the whole WSN by implementing button-push binding mechanisms. Individual applications can generate higher level information that could be used in other assistive automation applications to implement augmented features. However, the WSN infrastructure alone would not be enough to provide reasonable support for developing such applications. The microcontroller application software would grow in complexity as every possible application co-existence scenario would have to be implemented into the firmware. Also, maintainability of such applications would suffer from added complexity.

It is recommended that a dedicated interoperability server is used to implement more complex features into assistive automation WSNs. An interoperability server would be used to implement the "intelligence" into assistive automation applications, while embedded sensors and actuators in the WSN are left to deal with relatively simple task with high reliability. Any customization can be made centrally and there is no need to program meticulous details into every active device in the WSN.

General automation design philosophy suggests that the most critical safety mechanisms should be implemented in the lowest levels of a system. This means that, although most of the system's logic would be implemented on a server, the WSN itself could implement some safety mechanisms through a dedicated broadcast mechanism. This design is an analogy of human nervous system; reflex reactions are not processed in the brain, but in the spinal cord. If, e.g., a smoke detector detects smoke, it broadcasts an alarm message to the whole WSN. If a stove alert system receives an alarm message from a smoke detector, it will immediately turn off the stove. The server will also receive the alarm message, but it will only check if required actions were executed and send an alarm message to appropriate authorities.

## 7.3 Future Development Needs

The field of wireless sensor networks is developing rapidly. Already at the time of Ilmari platform's development, next generation Zigbee solutions are beginning to emerge. Although, WSN is a relatively new concept, Zigbee and IEEE 802.15.4 are starting to gain a firm stand among WSN standards. Also, the fragmented WSN market is starting to converge as smaller companies are being bought by bigger ones. The time seems to be ready for large scale commercial WSN applications.

Future development efforts in applying WSN technologies to assistive automation applications should be directed in developing application development models that enable the customization and co-operation of multiple applications. Implementing smart applications by using only WSN nodes is possible but maintenance and customization of the system will suffer. Experience with Ilmari platform shows that implementing simple WSN nodes that are controlled by a smart central unit is considered to be a better approach.

## References


[1] Helal A, Mokhtari M, Abdulrazak B editors. *The Engineering Handbook of Smart Technology for Aging, Disability and Independence.* John Wiley & Sons, Inc.. 2008.

[2] Sohraby K, Minoli D, Znati T. *Wireless Sensor Networks: Technology, Protocols, and Applications.* John Wiley & Sons, Inc.. 2007.

[3] Verdone R, Dardari D, Mazzini G, Conti A. *Wireless Sensor and Actuator Networks - Technologies, Analysis and Design.* Academic press: Elsevier. 2008.

[4] Zigbee Alliance. *ZigBee RF4CE.* 2009. Available at: "http://www.zigbee.org". Accessed 10/6, 2009.

[5] Nadimi ES, Søgaard HT, Bak T, Oudshoorn FW. *ZigBee-based wireless sensor networks for monitoring animal presence and pasture time in a strip of new grass.* Comput.Electron.Agric. 2008 5;61(2):79-87.

[6] Gislason D. *Zigbee Wireless Networking.* 2008. Newnes.

[7] Zensys. *Sensor Networks - Z-Wave.* 2007. Available at: "http://www.zen-sys.com". Accessed 10/7, 2009.

[8] Wikipedia. *TinyOS.* 2009. Available at: "http://en.wikipedia.org/wiki/TinyOS". Accessed 10/7, 2009.

[9] Wikipedia. *6LoWPAN.* 2009. Available at: "http://en.wikipedia.org/wiki/6LoWPAN". Accessed 10/7, 2009.

[10] Institute of Electrical and Electronics Engineers (IEEE) *IEEE 802.15.4-2006 - Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks.* 2006. Available at: "http://ieee.org".

[11] Institute of Electrical and Electronics Engineers (IEEE) *IEEE 802.15.4-2003 - Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).* 2003. Available at: "http://ieee.org".

[12] Institute of Electrical and Electronics Engineers (IEEE) *IEEE 802.15.4a-2007 - Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).* 2007. Available at: "http://ieee.org".

[13] Wikipedia. *IEEE 802.15.4-2006*. 2009. Available at: "http://en.wikipedia.org/wiki/802.15.4". Accessed 10/9, 2009.

[14] Zigbee Alliance. *Zigbee Alliance Web Page*. 2009. Available at: "http://www.zigbee.org". Accessed 14/9, 2009.

[15] Zigbee Standards Organization. *Zigbee Specification (r17)*. 2008. Available at: "http://www.zigbee.org". Accessed 1/7, 2009.

[16] Zigbee Standards Organization. *Zigbee Cluster Library Specification*. 2007. Available at: "http://www.zigbee.org". Accessed 1/7, 2009.

[17] Texas Instruments. *Low-power RF Guide*. 2009. Available at: "http://www.ti.com/lprf". Accessed 10/20, 2009.

[18] Tunstall. *ADLife - An Activities of Daily Living (ADL) monitoring solution, Solutions Sheet* 2009. Available at: http://www.tunstall.co.uk/. Accessed 10/26, 2009.

[19] Everon. *Everon website* 2009. Available at: http://www.everon.fi/. Accessed 10/26, 2009.

[20] Bhatia D, Estevez L, Rao S. *Energy Efficient Contextual Sensing for Elderly Care*. Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE 2007:4052-4055.

[21] Casas R, Marco A, Plaza I, Garrido Y, Falco J. *ZigBee-based alarm system for pervasive healthcare in rural areas*. Communications, IET 2008;2(2):208-214.

[22] Hussain S, Schaffner S, Moseychuck D. *Applications of Wireless Sensor Networks and RFID in a Smart Home Environment*. Communication Networks and Services Research Conference, 2009. CNSR '09. Seventh Annual 2009:153-157.

[23] Steele R, Lo A, Secombe C, Wong YK. *Elderly persons' perception and acceptance of using wireless sensor networks to assist healthcare*. Int.J.Med.Inf. 2009 12;78(12):788-801.

[24] Texas Instruments. *CC2430 Data Sheet - SWRS036F*. Revision 2.1. 2007. Available at: "http://focus.ti.com/lit/ds/symlink/cc2430.pdf". Accessed 10/6, 2009.

[25] Texas Instruments. *CC2480 Data Sheet - SWRS074A*. Revision 1.0. 2008. Available at: "http://focus.ti.com/lit/ds/symlink/cc2480a1.pdf". Accessed 10/6, 2009.

[26] Ember. *EM260 Data Sheet - 120-0260-000F*. Revision F. 2006. Available at: "http://www.ember.com/pdf/120-0260-000_EM260_Datasheet.pdf". Accessed 10/6, 2009.

[27] Radiocrafts. *RC2300-ZNM Data sheet*. Revision 1.01. 2009. Available at: "http://www.radiocrafts.com". Accessed 10/22, 2009.

[28] Jonsrud GE. *Texas Instruments Application Note AN040 - Folded dipole antenna for CC2400, CC2420, CC2430, CC2431, and CC2480*. Revision D. 2008. Available at: "http://focus.ti.com/lit/an/swra093d/swra093d.pdf". Accessed 7/14, 2008.

[29] Texas Instruments. *CC2480 Target Board Reference Design 1.4*. Revision A. 2008. Available at: "http://www.ti.com/litv/zip/swru156a". Accessed 9/24, 2008.

[30] Atmel. *ATmega 644P/V datasheet*. Revision 8011M-08/09. 2009. Available at: "http://www.atmel.com". Accessed 10/28, 2009.

[31] Atmel. *Application note AVR042: AVR Hardware Design Considerations*. Revision 2521F-AVR-04/08. 2008. Available at: "http://www.atmel.com". Accessed 10/29, 2009.

[32] Maxim. *MAX8840-MAX8842 Datasheet*. Revision 1. 2007. Available at: "http://datasheets.maxim-ic.com/en/ds/MAX8840-MAX8842.pdf". Accessed 11/2, 2009.

[33] Maxim. *MAX1722-MAX1724 Datasheet*. Revision 0. 2007. Available at: "http://datasheets.maxim-ic.com/en/ds/MAX1722-MAX1724.pdf". Accessed 11/2, 2009.

[34] Panasonic. *MP Motion Sensor "Napion"*. 2008. Available at: "http://panasonic-denko.co.jp/ac/e_download/control/sensor/human/catalog/bltn_eng_mp.pdf". Accessed 11/2, 2009.

[35] Texas Instruments. *CC2480 Interface Specification*. Revision A. 2008. Available at: "http://focus.ti.com/lit/er/swra175a/swra175a.pdf". Accessed 11/3, 2009.

[36] Wikipedia. *Serial Peripheral Interface Bus*. 2009. Available at: "http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus". Accessed 11/4, 2009.

[37] WinAVR. *WinAVR project website*. 2009. Available at: "http://winavr.sourceforge.net/". Accessed 11/5, 2009.

[38] Duracell. *Alkaline-Manganese Dioxide Battery Technical Bulletin*. 2005. Available at: "http://www.duracell.com/oem/Pdf/others/ATB-full.pdf". Accessed 11/6, 2009.

# Appendix A

This appendix illustrates hardware that was designed as a part of this thesis. PCB layouts and schematics are presented for each component of the developed hardware platform, as well for components that are used in the door monitor application demo. Layout images are best viewed in the electronic version of this document.

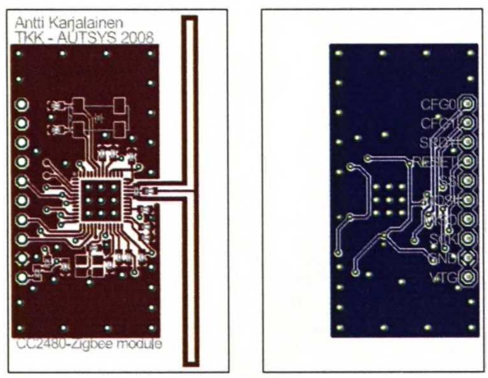


Figure A1: Zigbee Module Layout; 1:1 scale

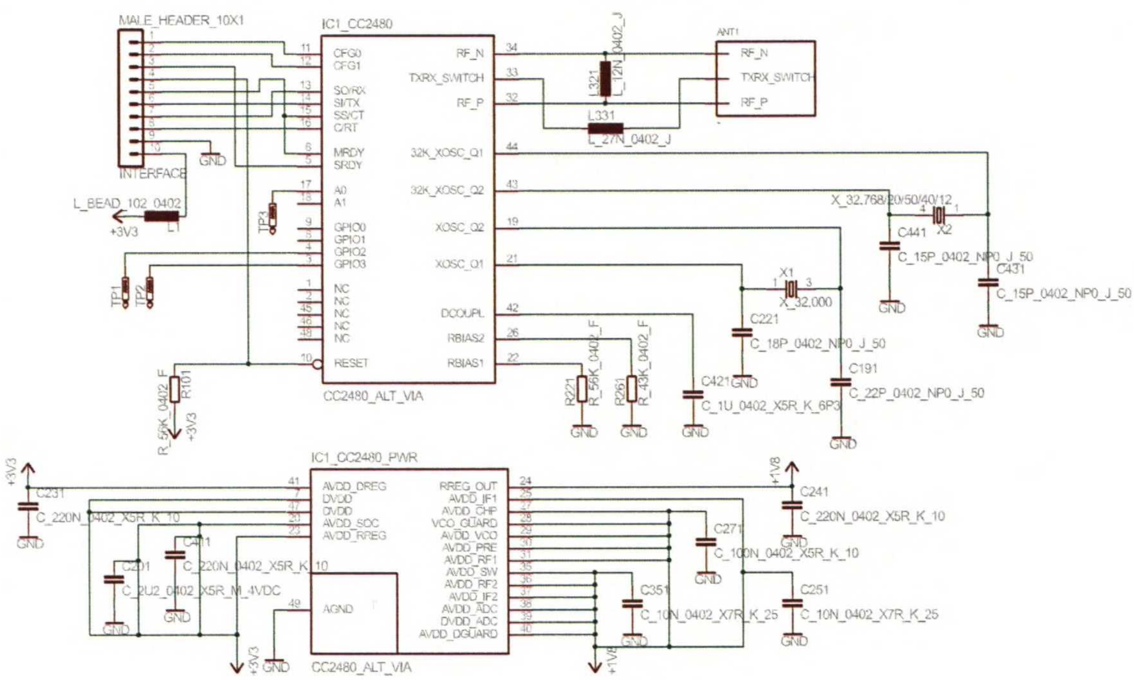


Figure A2: Zigbee Module Schematic

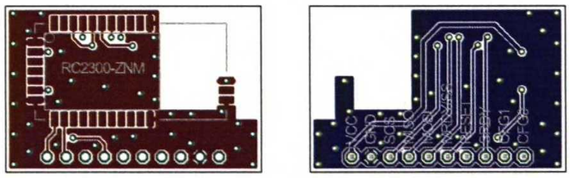

Figure A3: RC2300-ZNM Module Motherboard Layout; 1:1 scale

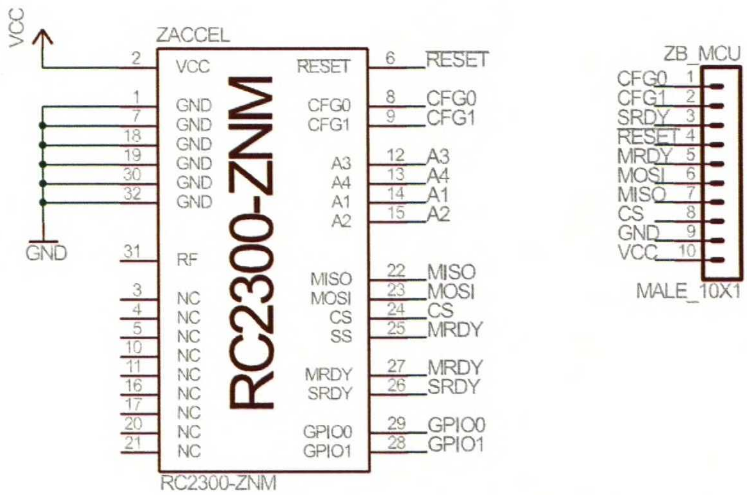


Figure A4: RC2300-ZNM Module Motherboard Schematic

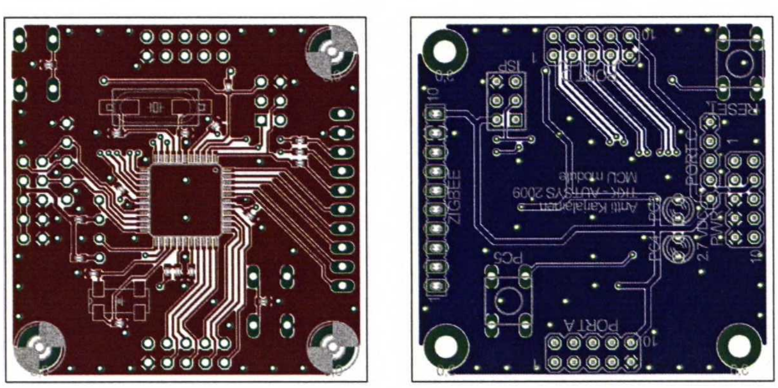

Figure A5: MCU Layout; 1:1 scale

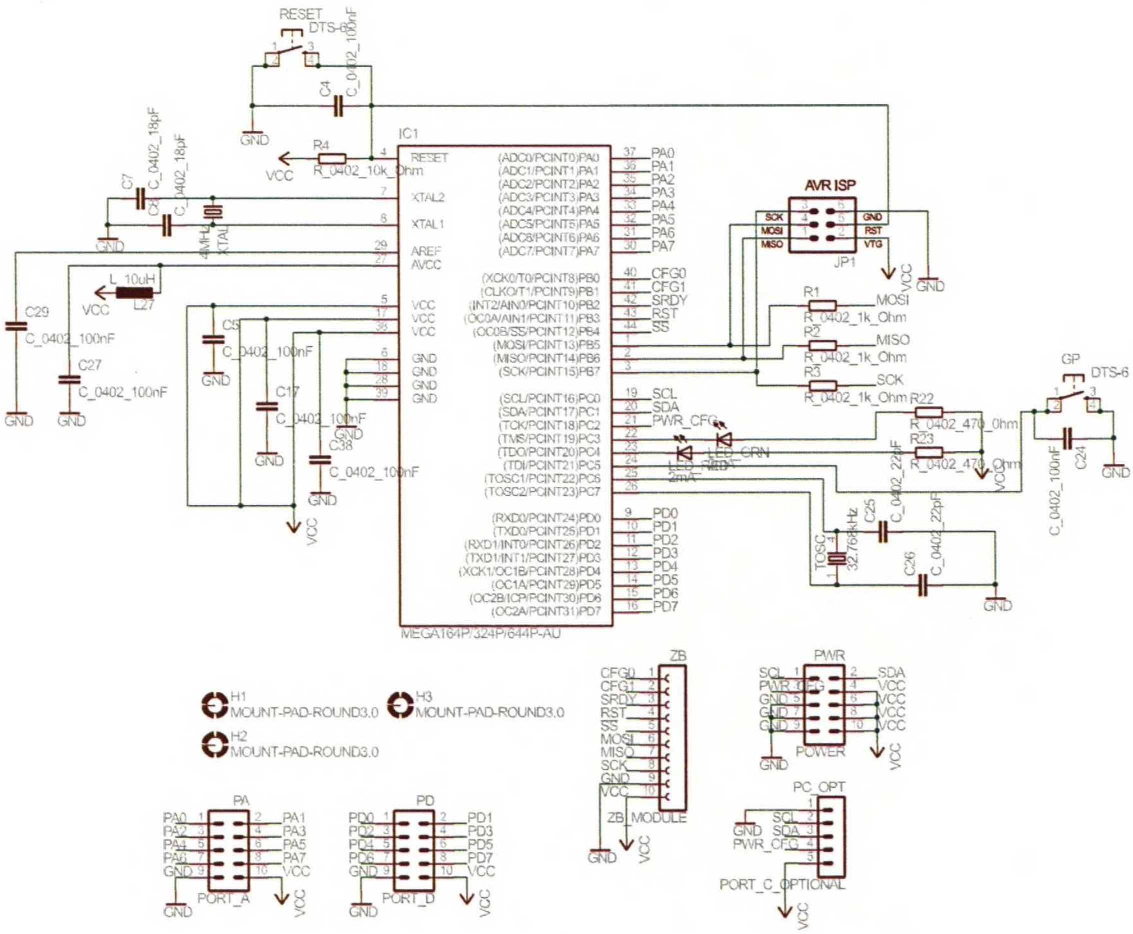


Figure A6: MCU Schematic

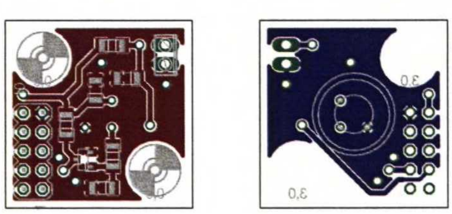

Figure A7: PIR Module Layout; 1:1 scale

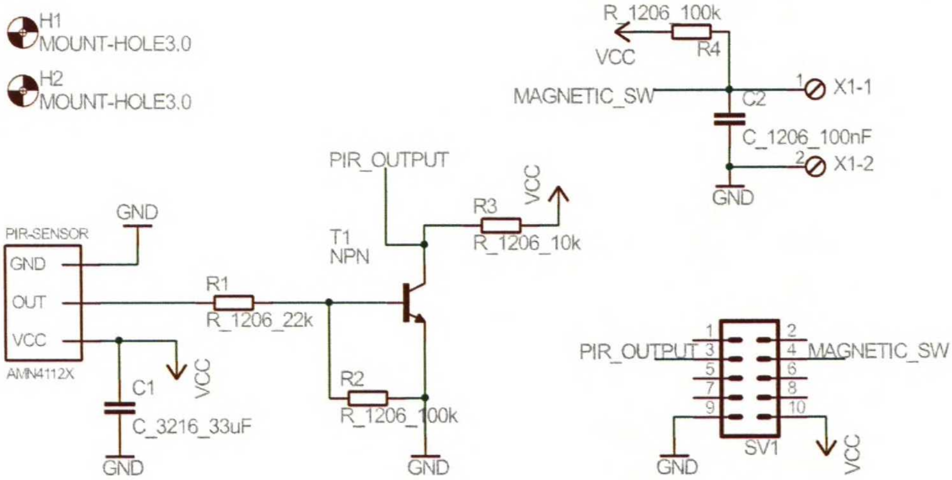


Figure A8: PIR Module Schematic

Figure A9: Step-up Converter Module Layout; 1:1 scale

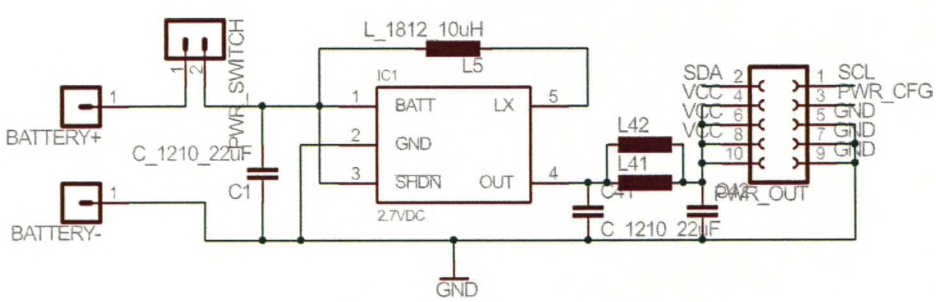


Figure A10: Step-up Converter Module Schematic

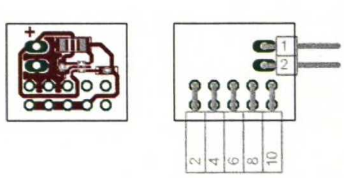

Figure A11: Regulator Module Layout; 1:1 scale

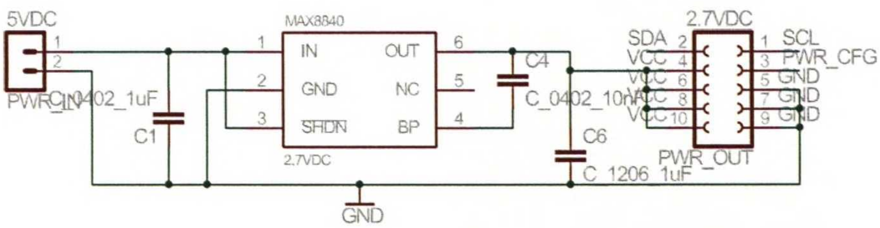


Figure A12: Regulator Module Schematic

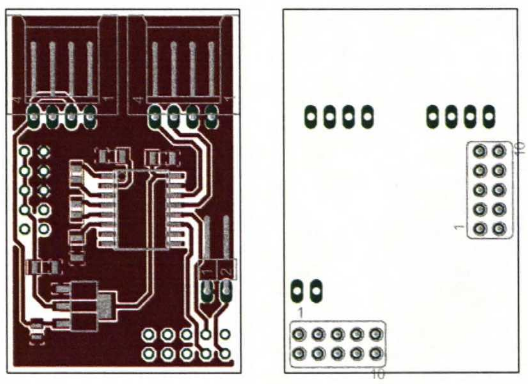

Figure A13: RS232 Converter Module for Door Monitor Layout; 1:1 scale

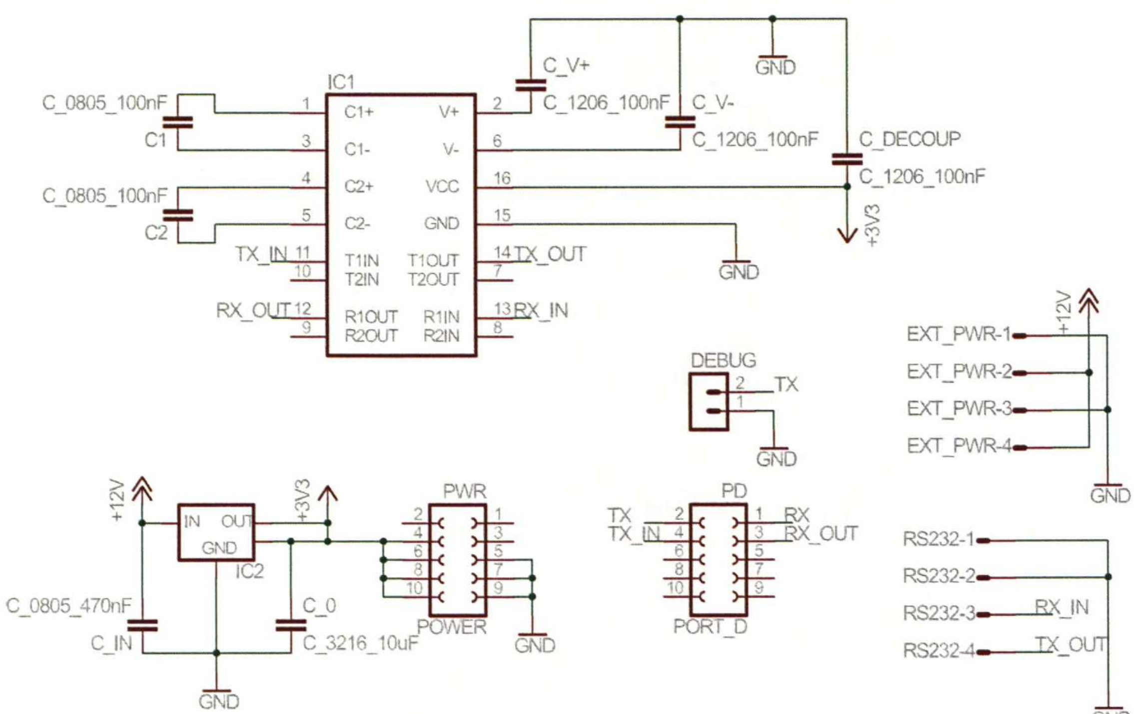


Figure A14: RS232 Converter Module for Door Monitor Schematic

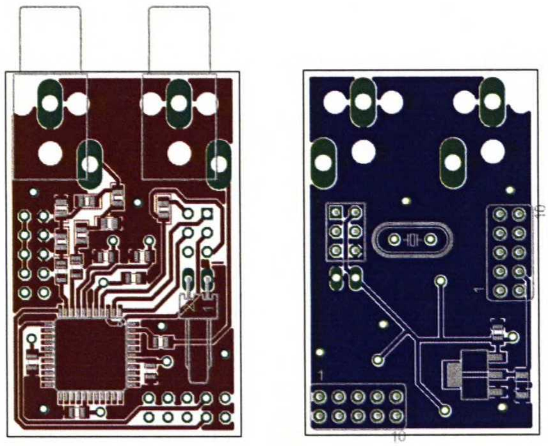

Figure A15: Video Generator Module for Door Monitor Layout; 1:1 scale

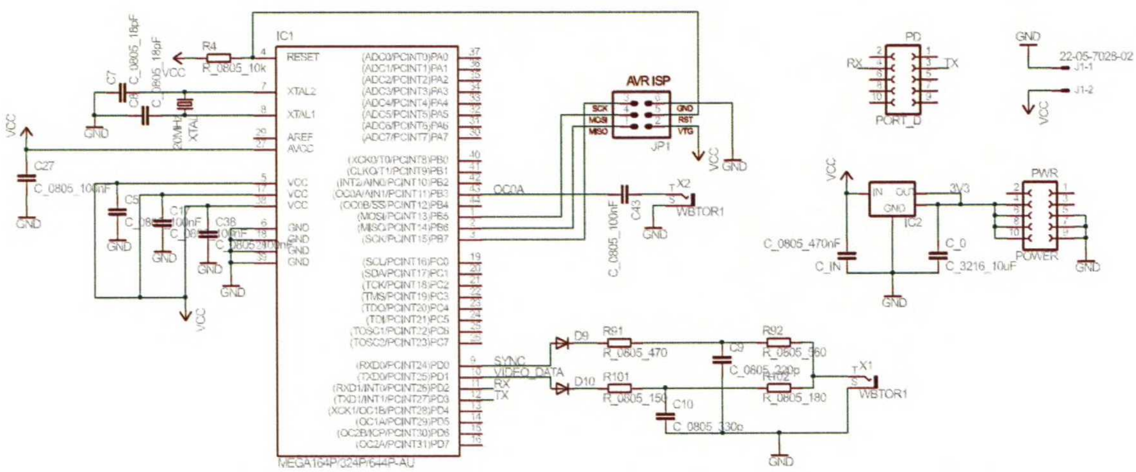

Figure A16: Video Generator Module for Door Monitor Schematic