A course on Wireless Sensor Networks (WSNs)

Luis Sanabria, Jaume Barcelo

December 13, 2012

Contents

Chapter 1

About the course

1.1 Course Data

Code: 21754

Course name: "Xarxes de Sensors Sense Fils"

Teacher: Luis Sanabria and Jaume Barcelo

Credits: 4

Year: 3rd or 4th year (optional)

Trimester: Spring

1.2 Introduction

The reduction in price and size of computing and wireless communication platforms over the last years opens a new possibility for gathering and processing information: Wireless Sensor Networks. A wireless sensor node is an electronic device of small dimensions that gathers measures from the environment and transmit the data wirelessly. In wireless sensor nodes, communication is often established with other wireless sensor nodes to exchange or pass information. It is common to have this data directed to an special device that gathers all the data and is called the network sink. As wireless sensor nodes are often battery-powered, energy saving is a relevant issue in these networks.

What follows is an extract of the first pages of [11].

> Wireless Sensor Networks (WSNs) are a result of significant breakthroughs on wireless transceiver technology, the need of event sensing and monitoring. One might think of a WSN as the skin of our bodies; apart from its importance on many other subjects, our skin senses events nearby it, like touch, temperature changes, pressure and so forth. These events are generated by an external entity, the nerves or sensors of our skin are capable to react to such events and transmit this information to the brain.
>
> There are enormous differences among characteristics of WSN and the skin, but the example given above will work as head start to understanding the technology. For instance, our skin sends the sensed event information towards the brain through the nerves, we could safely relate this medium to a wired network infrastructure. While in WSN, as its name suggests sends the sensed data towards a central node (Sink) via a wireless medium. Because of the limited radio range of each node, the route to the Sink is generally composed of jumps through different nodes (which is called a multi-hop route).
>
> The majority of wireless nodes in a WSN are very constrained devices due to the restrictions in

costs and sometimes harsh environments where these networks are developed. These constrains go from cost, processing power, memory, storage, radio range, spectrum and, more importantly, battery life. One of the most popular low-end nodes model, the TelosB, is equipped with 16 MHz CPU, very small flash memory (48 KB avg.), about 10 KB of RAM and works on the very crowded 2.4 GHz spectrum at rates around 250 Kbps. These limitations force WSN engineers to design applications capable of working with low processor-intensive tasks and powered with limited battery (usually two AA batteries).

Many WSN applications process the sensed event before sending the data, this processing tries to reduce the information to send. As mentioned in [1], it is less energy consuming to process one bit of information than sending it. WSN protocols and applications are tailored to power conservation rather than throughput, mainly due to cost, dimension, processing and power constraints.

WSNs may contain different kind of sensors that help monitor metrics related to: temperature, humidity, pressure, speed, direction, movement, light, soil makeup, noise levels, presence or absence of certain kinds of objects, mechanical stress and vibration. Also further information like node location can be derived from a Global Positioning System (GPS) device embedded at each node.

Because of the variety of measures than can be monitored with these small and (generally) cheap devices, a wide range of applications have been developed; the authors of [1] divide them in: military, environmental, health, home and industry applications.

- *Military Applications:* one of the first applications of WSNs. The main advantages in this area are the fact that the deployment of low cost sensors (that are subject to destruction in a battlefield) proposes a cheaper approach to sensing different types of metrics, which in turn brings new challenges to WSN applications (increased power and processing constraints). Some of the applications are related to: monitoring the movement of troops, equipment and ammunition, battlefield surveillance, terrain reconnaissance, damage assessments, snipper detection [8], [9] and threat detection, as in the case of biological, radiological or chemical attacks.

- *Environmental Applications:* most of these applications are related to animal tracking, weather conditions and threat contention [10], [12].

- *Health Applications:* a great deal of these applications are dedicated to monitor patients inside hospitals and provide them with better care. This is achieved by tracking the patients vitals or other information of interest and making it available to doctors at any time from anywhere securely through the Internet.

- *Home Applications:* technology is making its way inside our homes from various fronts, and WSN are no exception. Sensor nodes inside domestic devices will result in an increased interaction among them and allow access via the Internet. Theses applications are of great importance in fields like domotics towards a smart home/work environment. Home surveillance and multimedia WSNs for home environments are also a growing field of research.

- *Industrial Applications:* historically the monitoring of material fatigue was made by experts introducing the observed situation inside PDA devices to be collected on a central site for processing. Further sensing techniques were developed on the form of wired sensors; nevertheless its implementation was slow and expensive due the necessary wiring. WSNs bring the best of both methods by sensing the events without the need of expert personnel and the cost of wiring.

- Other implementations as mentioned in [1] are: inventory management, product quality monitoring, smart offices/houses; guidance in automatic manufacturing environments, interactive museums, factory process control and automation, machine diagnosis, transportation, vehicle tracking and detection, spectrum sensing for cognitive radio networks, underground and underwater monitoring.

1.3 Syllabus

- Lectures
  1. Introduction to WSNs
  2. Arduino Platform
  3. Zigbee and 802.15.4
  4. Xbee and Xbee explorer
  5. Processing
  6. tbd
  7. tbd
  8. Project presentation
  9. Invited talk
- Seminars
  1. Dimming LED + button
  2. tbd
  3. tbd
  4. tbd
- Lab Assignments
  1. Project proposal and planning
  2. Project implementation
  3. Project prototype (alpha)
  4. Project prototype (beta)
  5. Project demonstration

1.4 Bibliography

The course closely follows the book:

Robert Faludi "Building Wireless Sensor Networks" ([6]).

1.5 Evaluation Criteria

The grading is distributed as follows:

- Lectures continuous assessment, 20%
- Lab assignments, 20%
- Project proposal, 20%
- Project presentation, 20%
- Project demonstration, 20%

It is necessary to obtain a decent mark in all the different evaluation aspects. To pass the course, 50 out of the total 100 points need to be obtained.

1.6 Survival guide

1.6.1 Questions and doubts

WE like to receive questions and comments. Normally, the best moment to express a doubt is during the class, as it is likely that many people in the class share the same doubt. If you feel that you have a question that needs to be discussed privately, we can discuss it right after the class.

1.6.2 Continuous feedback

At the end of the lecture, we will ask you to anonymously provide some feedback on the course. In particular, I always want to know:

- What is the most interesting thing we have seen in class.
- What is the most confusing thing in the class.
- Any other comment you may want to add.

  In labs, I will ask each group to hand in a short (few paragraphs) description of the work carried out in class, and the members of the group that have attended the class. Note that this is different from the deliverables, which are the ones that are actually graded.

1.6.3 How to make you teachers happy

Avoid speaking while we are talking.

Chapter 2

Introduction to Arduino

2.1 Open Hardware

"There's a fine line between open source and stupidity", says Massimo Banzi to a reporter from Wired Magazine while having dinner at a restaurant in Milan.

Banzi is the man behind Arduino, an open hardware platform. The open about it relates to the fact that the device's manufacturing schematics, programming language and software development environment are free and open source. This basically means that everyone interested on building hardware-coupled solutions may take an Arduino board's schematics, modify it at will, send the new design to a China manufacturer and get the final product back home for around €10 [5].

Open hardware is supported by a variety of available licenses (like open software with LGPL, GPL, Copyleft, and others) that ensure that the protected platform can be copied, enhanced and even sold, but always recognizing the original authors. It also ensures that the resulting products are open as the original.

2.2 The Arduino Platform

Arduino was developed to teach Interaction Design [3], that meant that it required the ability to sense the surroundings and do something about it.

The platform is equipped with simple digital and analog input/output interfaces, that can be programmed to sense or react to some events. Figure 2.1 shows the Arduino Duemilanove board.

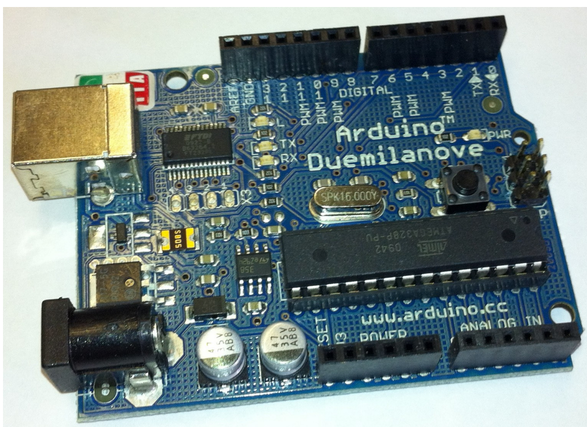

Figure 2.1: Arduino Duemilanove board

There are numerous sensors and actuators that work with Arduino. In relation to sensors: temperature, air pollution, light, GPS modules and sound are among the popular; as LEDs, speakers and digital/analog outputs are common actuators. Also, interfaces like buttons can be programmed and used as a human interactive input.

The design and electrical components of the Arduino board are available for anyone [2]. Figure 2.2 shows the connections layout of the Duemilanove model (see Figure 2.1).

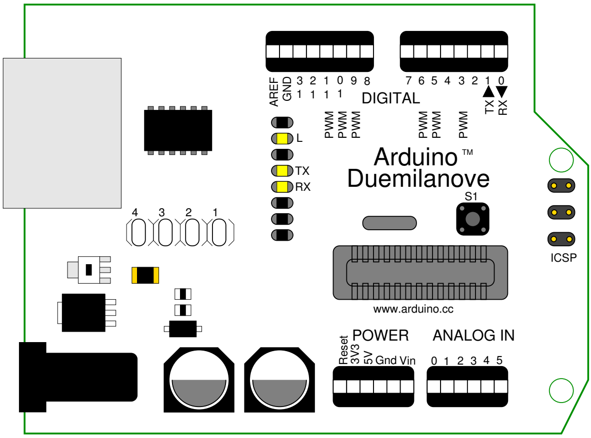

Figure 2.2: Arduino Duemilanove board: layout

2.2.1 Developing for Arduino

2.2.2 Example of a simple Arduino application

Chapter 3

An introduction to Zigbee and IEEE 802.15.4

Zigbee and IEEE 802.15.4 are specification for the higher and lower network layers of WSNs.[1] Zigbee devices are designed for low-range, low-complexity, low-cost and low-consumption applications. Zigbee includes low duty-cycle capabilities, which mean that the devices can sleep for most of the time. A zigbee product can potentially run on batteries for years. In our particular course, we use zigbee combined with Arduino. As Arduino is not designed for low consumption, it kills the possibility of running our prototypes on batteries for such a long time.

Zigbee defines the network and application layers of the devices and relies on IEEE 802.15.4 for the MAC and PHY layers. The specification of Zigbee standards is carried out by the Zigbee Alliance, which comprises all kind of companies (semiconductor industry, OEMs, software developers, etc.).

IEEE 802.15.4 operate in three different bands: 868 MHz, 915 MHz and 2.4 GHz. The 2.4 GHz band is a band for indus-

[1]This chapter is based on [4] and [7]

trial, scientific and medical uses (ISM). This band is shared by many technologies, including the IEEE 802.11 family. Spread spectrum techniques are used to alleviate the problem of interference. The modulations that are used are BPSK, ASK and O-QPSK.

Direct sequence spread spectrum (DSSS) or parallel sequence spread spectrum (PSSS) is used.

3.1 Devices

There are two different types of devices: full-function devices (FFD) and reduced-function devices (RFD). FFDs are more powerful and can assume any role in the network. RFDs are simpler, and can assume only the role of "Device" in the IEEE 802.15.4 terminology (equivalent to "ZigBee End Device" in the ZigBee terminology.

The other two possible roles, that can be assumed only by FFDs are "Coordinator" and "PAN Coordinator" in the IEEE 802.15.4 terminology. These are equivalent to "ZigBee Router" and "ZigBee Coordinator" in the ZigBee terminology.

A ZigBee End Device can be the source or destination of a packet, but it cannot forward packets for other nodes. A ZigBee Router can relay packets of other nodes. And a ZigBee Coordinator is the head of the network. In every every network, there is one and only one Coordinator.

Regarding the topologies, we can consider three different cases: star, tree and mesh.

3.2 Channel Access

In IEEE 802.15.4 there are two methods for networking: beacon-enabled networking and non-beacon networking. In beacon-enabled networking the coordinator transmits periodical beacons to synchronize the network. In these beacons, it is possible to define a super-frame structure. The super-frame structure spans the channel time between two consecutive beacons, and differentiates three periods: Contention Access Period (CAP), Contention Free Period (CFP, optional) and Inactive Period (IP, optional). The time is divided in slots, and multiple slots can be assigned to each of the different periods. In the CAP, the devices use CSMA/CA to contend for channel access. In the CFP, the channel time is reserved and only the devices that own the reservation can transmit. In the IP, the devices can go to sleep to save energy.

In non-beacon networking, the devices must always contend for the channel. The contention uses the CSMA/CA protocol. Before transmitting, the devices sense the channel for the presence of an ongoing transmission. If a transmission is detected, the devices backs off and re-attempts after a random period of time.

There are two mechanism to detect an ongoing transmission. Energy detection in which the device simply measures the amount of energy on the channel, and carrier detection in which the device looks for the presence of an IEEE 802.15.4 carrier. Either one or the other, or even a combination of both, can be used.

3.3 MAC handshake

In a beacon-enabled network, a node that wants to transmit data to the coordinator waits for the CFP if it has a reservation. Otherwise, it waits for the CAP. Then it transmits the packet (using CSMA/CA if it is in contention mode). The sender can request acknowledgement and, in this case, the coordinator may acknowledge the reception.

When a coordinator wants to transmit data to a device it indicates in the beacon the destination of the packet. The device processes the beacon and learns that there is data pending for it. After that, the device sends a “data request” message to the coordinator that the coordinator must acknowledge. Then the coordinator sends the data packet to the device which may acknowledge the correct reception.

In a non-beacon network, a device can transmit to the coordinator whenever the channel is sensed empty. If a coordinator has data for a device, it waits until the device sends a “data request” which must be acknowledged. Then the coordinator sends the data to the device, that may acknowledge the correct reception.

3.4 MAC problems in multi-hop wireless networks

There are two well-known problems that recurrently appear in multi-hop wireless networks. The first one is the “hidden node problem” and is represented in Figure 3.1. Imagine that nodes A and C cannot carrier sense each other. Then it can happen that C starts a transmission while A is transmitting to B. This results in a collision and the receiver may not be able to recover any of the packets.

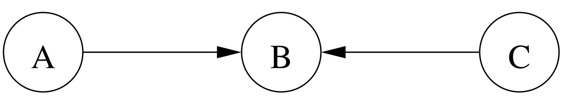

Figure 3.1: Hidden node problem

The other problem is the “exposed node problem”, illustrated in Figure 3.2. In the example, E is transmitting to D. Node F wants to transmit to G, but it senses the channel busy (as it detects E's transmission). F does not transmit when, in fact, it could transmit without interfering with the other transmission.

Figure 3.2: Exposed node problem

3.5 Addresses and Network Layer

Each device has two addresses: a 16 bit short address and a 64-bit extended address.

Bibliography

[1] I. Akyildiz and M.C. Vuran. *Wireless sensor networks.* John Wiley & Sons, Inc., 2010.

[2] Arduino Team. Arduino homepage. http://www.arduino.cc, October 2012.

[3] Banzi, M. *Getting Started with arduino.* Make Books, 2008.

[4] P. Baronti, P. Pillai, V.W.C. Chook, S. Chessa, A. Gotta, and Y.F. Hu. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. *Computer communications*, 30(7):1655–1695, 2007.

[5] C. Thompson. Build it. Share it. Profit. Can Open Source Hardware Work? http://bit.ly/ShWD43, October 2008.

[6] R. Faludi. *Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing.* O'Reilly Media, Incorporated, 2010.

[7] S. Farahani. *ZigBee wireless networks and transceivers.* Newnes, 2008.

[8] Á. Lédeczi, A. Nádas, P. Völgyesi, G. Balogh, B. Kusy, J. Sallai, G. Pap, S. Dóra, K. Molnár, M. Maróti, et al. Countersniper system for urban warfare. *ACM Transactions on Sensor Networks (TOSN)*, 1(2):153–177, 2005.

[9] J.A. Mazurek, J.E. Barger, M. Brinn, R.J. Mullen, D. Price, S.E. Ritter, and D. Schmitt. Boomerang mobile counter shooter detection system. In *Proceedings of SPIE*, volume 5778, page 264, 2005.

[10] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson. Analysis of wireless sensor networks for habitat monitoring. *Wireless sensor networks*, pages 399–423, 2004.

[11] Sanabria, L. Localization Procedure for Wireless Sensor Networks. Master's thesis, Universitat Pompeu Fabra, Barcelona, 2012.

[12] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, 2004.