Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Communications 28 (2005) 1545–1551
computer communications
www.elsevier.com/locate/comcom

ScatterWeb: A wireless sensornet platform for research and teaching

Jochen Schiller*, Achim Liers, Hartmut Ritter

Institute of Computer Science, Freie Universität Berlin, Takustr 9, Berlin 14195, Germany

Received 31 August 2004; accepted 15 December 2004
Available online 29 April 2005

Abstract

This article presents ScatterWeb, a distributed, heterogeneous platform for the ad-hoc deployment of sensor networks offering hardware and open, fully documented software for research and teaching in embedded sensor networks. ScatterWeb consists of sensor nodes and several gateways connecting the sensor network to other wired or wireless networks. Already low-power by design, the sensor nodes offer additional energy conservation mechanisms and support energy efficient routing and applications. Even image capturing and transmission is possible with a minimum of energy. Together with auto-configuration and remote reprogramming techniques, these power saving mechanisms enable ScatterWeb nodes to survive many years in real-life scenarios without any in-place maintenance.

Keywords: Wireless sensor networks; Low-power; Auto-configuration; Self-organising networks

1. Introduction

Wireless sensor networks are an information gathering paradigm based on the collective efforts of many small wireless sensor nodes. The sensor nodes, which are intended to be physically small and inexpensive, are equipped with one or more sensors, a short-range radio transceiver, a small micro-controller, and a power supply in the form of a battery or mechanisms that try to scavenge energy from the environment. Sensor network deployments are envisioned to be done in large scales, where each network consists of hundreds or even thousands of sensor nodes. In such a deployment, human configuration of each sensor node is usually not feasible, and therefore, self-configuration of the sensor nodes is important. Energy efficiency is also critical, especially in situations, where it is not possible to replace sensor node batteries. In commercial sensor network installations, e.g. intelligent buildings, battery replacement is typically no option as the systems should allow maintenance-free operation.

Most sensor network applications aim at monitoring or detection of phenomena. Examples include office building environment control, wild-life habitat monitoring, and forest fire detection [13,19]. For such applications, the sensor networks cannot operate in complete isolation; there must be a way for a monitoring entity to gain access to the data produced by the sensor network. By connecting the sensor network to an existing network infrastructure such as the global Internet, a local-area network, or a private intranet, remote access to the sensor network can be achieved.

All tasks in wireless sensor networks, such as routing, sensing, data aggregation, or node updating, have to take into account the very limited resources of the nodes. While many projects assume that all nodes in a sensor network are battery-driven [1,11,20,21], nodes can also be powered by other energy sources such as vibration, temperature differences, or solar power [15]. Nodes powered by such a source can receive and transmit packets and sense parameters without consuming battery energy. Therefore, e.g. routing packets via such nodes is appealing.

Although there has already been done a lot of research in the area of sensor networks, most of it relies on simulations. We believe in a sound combination of simulations and real hard- and software. This is the only way to really investigate into the important topics of energy aware design and applications, energy aware protocols and routing. The key contributions of this paper are the development of ScatterWeb, an open and flexible sensor network platform, and the design and evaluation of low-power sensor nodes and applications.

* Corresponding author.
E-mail addresses: schiller@inf.fu-berlin.de (J. Schiller), liers@inf.fu-berlin.de (A. Liers), hritter@inf.fu-berlin.de (H. Ritter).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2004.12.044

The paper is organized as follows. Section 2 gives a short overview of the ScatterWeb platform and its hard and software. Section 3 describes basic power conservation mechanisms implemented in ScatterWeb's sensor nodes, while Section 4 discusses special extensions for low-power image capturing and transmission.

2. ScatterWeb

ScatterWeb [16] offers a distributed, heterogeneous platform for the ad-hoc deployment of sensor networks. The platform comprises both hardware and open, fully documented software. Fig. 1 shows a typical example for a sensor network based on ScatterWeb. Many sensor nodes, called ESB (Embedded Sensor Board, Section 2.1), are distributed in the environment for monitoring parameters such as temperature, vibration, light, etc. The sensor nodes may forward their data via other sensor nodes towards more powerful data sinks. ScatterWeb uses so-called embedded web servers (EWS, Section 2.2) as gateways between the sensor world and classical wired or wireless networks (Ethernet, Bluetooth, GPRS…). ESBs as well as EWSs can be deployed in an ad-hoc manner and after that the nodes perform automatic topology discovery and auto-configuration. The backbone of the sensor network, set-up between the EWSs, can be wireless or wired. Section 2.2 motivates the usage of simple web servers for this purpose.

2.1. Embedded sensor board (ESB)

As the name already implies, sensor networks need sensors, typically hundreds or thousands of sensors, to collect data from their environment. All sensor nodes may form an ad-hoc network with some nodes acting as data sources, some as relays, and some as data sinks collecting data. Nodes may act in all three roles at the same time. Typical communication scenarios of sensor networks based on the ESBs are:

- ESBs communicate via the serial port or USB with a standard PC/PDA for application development.
- ESBs communicate with GSM/GPRS modules to connect to wide-area mobile phone networks. This enables remote configuration of ESBs even via short messages (SMS) as well as reception of sensor data on mobile phones world-wide.
- ESBs communicate via their radio interface with other ESBs or an embedded web server (EWS) to form a truly embedded, highly flexible sensor network solution.

A full featured ESB (see Fig. 2) comprises besides a controller and transceiver the following components: Luminosity sensor, noise detection, vibration sensor, IR movement detection, precise timing, microphone/speaker, IR sender/receiver. This type of ESB is typically used for research and teaching. Furthermore, it serves as basis for application development and the creation of special purpose sensor nodes.

Sensors should follow the principle 'embed and forget', i.e. they must have an extreme low-power design. Typical average values for a full featured ESB running at 3.3 V are:

- ESB up and running with all sensors: 12 mA
- ESB transmitting data: less than 8 mA (avg.)
- ESB deep-sleep (clock only): 8 µA

Example scenarios with standard AA batteries (life-time varies depending on the sensors used):

- 5 years life-time with a duty-cycle of 1% (sensing and transmitting)
- 17 years life-time sending 25 B every 20 s

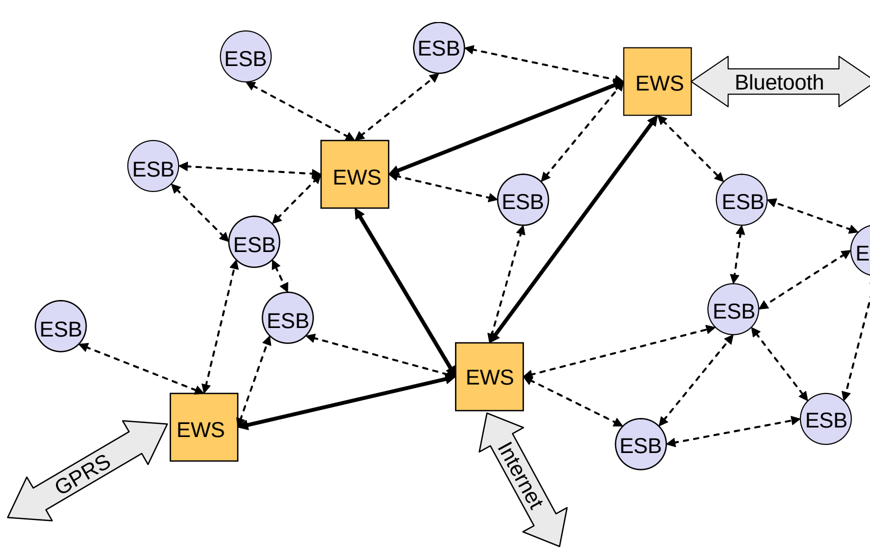

Fig. 1. ScatterWeb network example.

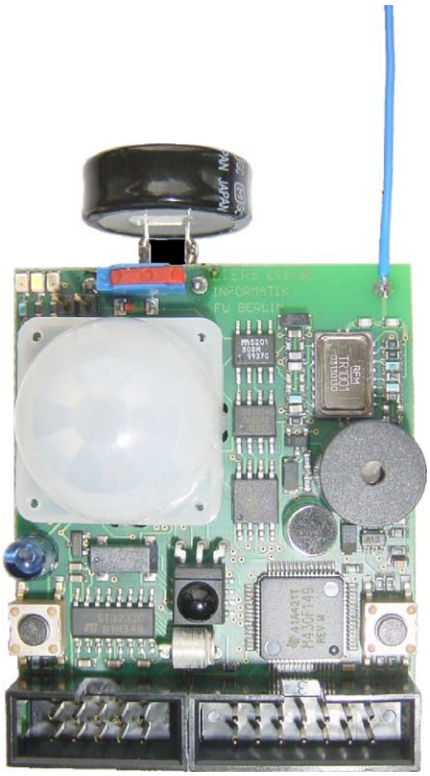

Fig. 2. Full featured embedded sensor board.

Surely, no normal battery will live that long and, thus, these values are rather theoretical. Due to its modular design, it is relatively simple to derive other modules from the ESB as shown in Fig. 2. Fig. 3 shows an almost radio-only module-it still contains a temperature and vibration sensor and is used in monitoring intelligent buildings. Other nodes designed within the ScatterWeb project comprise combined sensor/actor nodes for soccer playing robots, nodes for temperature measurement in the Baltic Sea (see Fig. 4), or nodes for the protection of jewellery.

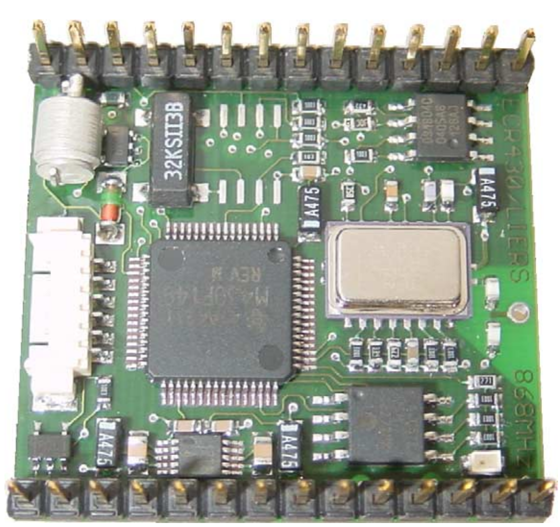

Fig. 3. Low-power radio only module.

Fig. 4. Temperature sensor for the Baltic Sea (not sealed yet).

2.2. Gateways and programming

A very convenient access to a ScatterWeb is the ScatterFlasher (see Fig. 5). A simple USB stick connects a PC to the sensornet and enables over-the-air flashing of all sensors, debugging, collecting sensor data, etc. The stick installs as USB device on the PC/PDA. This means, that no more cables are needed for the reconfiguration of the sensor network. No matter how many nodes have to be reprogrammed, the new software is distributed throughout the network; the nodes check the new software for correctness and then flash themselves.

Fig. 6 shows the internal memory layout of a sensor node. The firmware used for flashing is protected and, thus, no node is rendered useless even if the new software is faulty. In case of faulty software (e.g. endless loops, crashing software) the hardware can flush the defect software and waits for a new download. A node stores the new software in the EEPROM first for checking the correct reception. Flashing can be either synchronized by time or by command. All tasks on a node are linked against a certain firmware version and can use all functions. Tasks can also register callback functions.

The software generated by the compiler is fragmented into smaller packets for downloading over the air. This is done to minimize packet loss due to bit errors. If a node misses a packet due to interference, retransmission is requested locally first. With a very high probability neighbors received the missing packet(s) and can perform the retransmission without involving the gateway.

We use simple web servers (EWS, embedded web server) as gateways to the networks outside the sensornet. The main reason for this is the simplicity of access to the sensor network. No matter which network is used for accessing the sensor network, sensor data is provided as web service or simple web page. This allows for a plethora of devices accessing the sensed data as long as a web browser is running on the device. A very typical scenario is reading

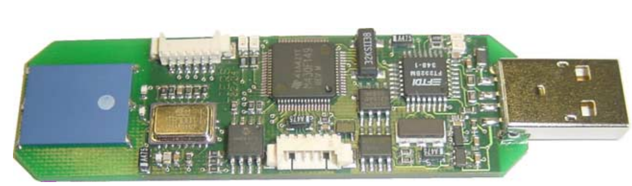

Fig. 5. ScatterFlasher for over-the-air programming.

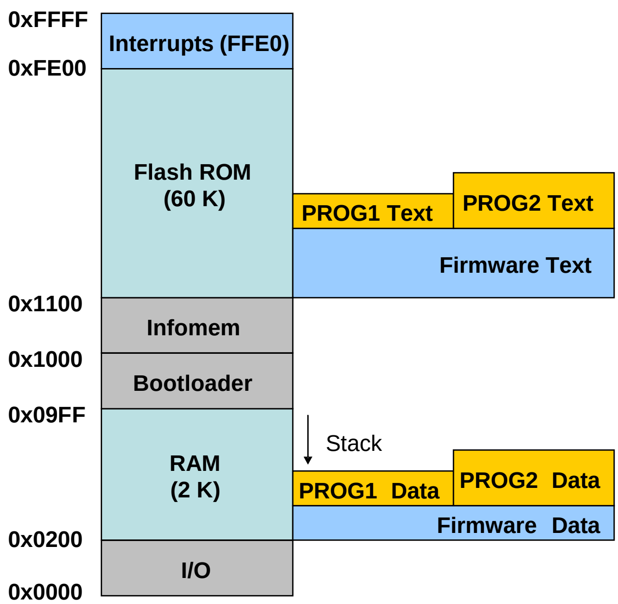

Fig. 6. Sensor node memory layout.

the temperature values of rooms in a building with a PDA connected by Bluetooth with the EWS collecting data from all sensor nodes. However, the web servers are not only used for reading values but also for setting up an (ad-hoc) infrastructure for the sensor network. Typically, an EWS is connected to power and not battery driven. Therefore, it can use more power for radio transmission and, thus, cover longer distances. The EWSs provide also the gateway for reprogramming the sensor nodes in addition to the direct access via the ScatterFlasher. Reprogramming can be done (after authentication) from arbitrary hosts in the Internet by accessing the web server and downloading new software.

Fig. 7 shows an EWS that bridges Ethernet and a sensor network. Power-over-Ethernet provides the energy necessary for the embedded ARM processor and the transceiver circuitry. Besides the standard Internet protocols the module performs auto-configuration with the help of DHCP and offers complete web pages and applets for accessing and configuring the sensor network.

In order to bridge longer distances without wires for monitoring devices using RS485 interfaces, we developed a gateway as shown in Fig. 8.

A Bluetooth-to-ScatterWeb gateway complements the current family of gateways. All components are easy to access via an open and fully documented C interface [16]. The platform provides software for the following features:

Fig. 7. Embedded Web Server (EWS) connecting the sensor network with Ethernet.

Fig. 8. RS485-to-ScatterWeb gateway.

- Configuration and analysis of all sensor data.
- Control of peripherals: timer, transceiver, serial port, EEPROM, IR-send/receive, LEDs, beeper, switches.
- Sending and receiving of data packets via the radio interface, transmission of sensor data, communication with other systems using the same radio interface (ESBs, EWSs).
- Sensing and receiving of short messages (SMS) via a GSM module/mobile phone. Just send the ESB an SMS triggering temperature monitoring every Tuesday morning from 7:00 a.m. until 11:43 a.m., if and only if there is movement in the room but no light—it is possible with a single SMS (or also via GPRS).
- Sending (e.g. to control home entertainment devices) and receiving (e.g. from conventional remote controls) of RC5 packets via infrared.
- Periodic polling of sensor data and (depending on settings) automatic transmission via the radio interface, the serial port (terminal or mobile phone/GPRS) or storage into the ring buffer of the local EEPROM.
- Simple and easy configuration and control of the ESB via terminal commands over the serial interface or a mobile phone.
- The standard TCP/IP protocol suite is available for ScatterWeb [6,7] including a simple web server directly running on the sensor node.
- The nodes are able to run operating systems such as TinyOS [17] and Contiki [5].

3. Energy conservation

Wireless sensor networks always have to take into account the very limited energy resources of the nodes compared to their wired counterparts. While many projects assume that all nodes in a sensor network are battery-driven [1,11,20,21], nodes can also hoard energy by scavenging their environment [10,15]. Typical environmental energy sources are solar power, vibrations, temperature differences, or mechanical forces (opening doors, flipping light switches, etc.). While energy collected from the environment is considered free, the nodes still have to manage the scarce amount of energy stored, e.g. in capacitors, in similar ways it has to do for battery power.

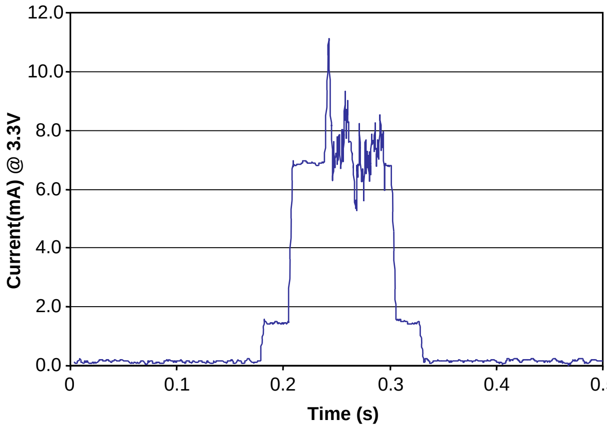

Fig. 9. Power consumption: sending.

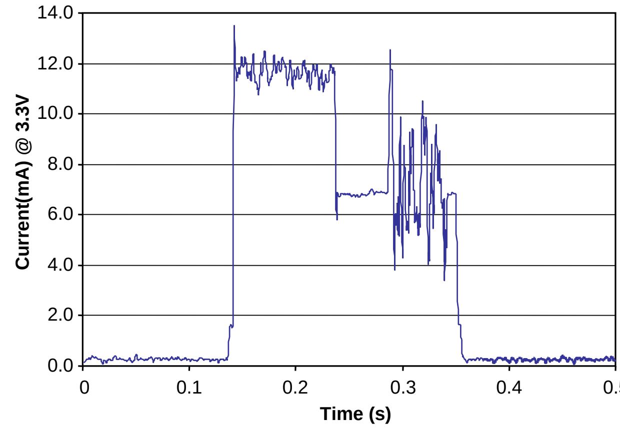

Fig. 11. Power consumption: sensing and sending.

First of all, ScatterWeb's sensor nodes are low-power by design. We use TI's MSP430 as micro controller that requires less than 2 μA in deep-sleep mode. The sensors can interrupt and wake-up the system without any polling required. The controller offers several low-power modes by activating/deactivating certain function blocks, e.g. the A/D converters.

For many sensor node simulations, it is assumed that sending is always much more expensive in terms of power consumption than receiving. This is not necessarily the case as Figs. 9 and 10 show. One version of the sensor board uses very simple 868 MHz transceivers (RFM TR1001, [14]). Fig. 9 shows a typical sending cycle. The circuitry wakes up from a low-power mode and wakes up the transceiver. After some time the transceiver circuits need for stabilizing, sending of data starts (first sync patterns then data) consuming less than 8 mA on average. Fig. 10 shows that receiving data is similar expensive with 7 mA. Therefore, using a transceiver like we do changes many simulation scenarios that rely on the fact that sending is always much more expensive. Clearly, sending power consumption very much depends on the output power requested at the antenna. With the simple transceiver as shown we can reach approximately 300 m in open space and 30 m inside buildings. The newer nodes using more advanced transceivers (e.g. CC1021 from Chipcon [2]) offer roughly 10 times the range (e.g. transmission through 15 floors in a building is possible).

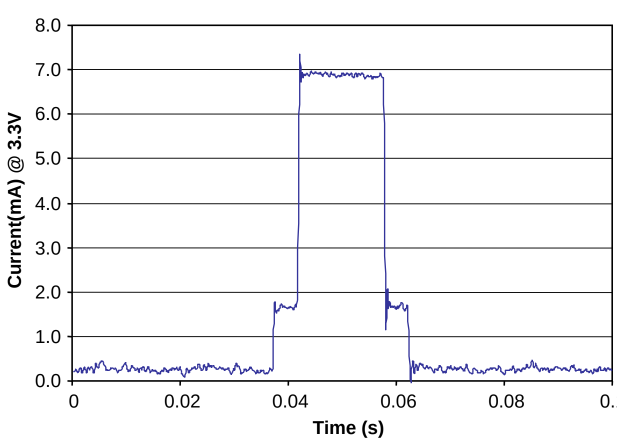

Fig. 10. Power consumption: receiving.

A typical cycle for a sensor is the following: deep sleep for more than 90% of the time, waking up and start sensing the environment, letting the values stabilize, switch off the sensors and start transmitting the sensed values. Fig. 11 shows such a typical cycle. 200 ms are typically more than enough to sense and transmit data plus receiving a short acknowledgement.

No matter how low-power the design is, energy is always limited on sensor nodes, particularly if they run on scarce environmental energy sources. It is crucial for reliable operation that nodes automatically switch to low-power modes before running out of energy or even perform a safe shut-down if all energy is consumed. Furthermore, power-up must be performed in a way that does not leave the node in a useless state.

ScatterWeb's sensor nodes come with an elaborated power sensing mechanism that allows operation with batteries or fluctuating environmental power sources. As soon as enough energy is available to run the controller, the nodes perform periodic power sensing and start the transmission of data if and only if the energy stored in a capacitor is sufficient for the complete transmission plus the reception of an acknowledgement. Also the period of transmitting sensor values depends on the available energy.

Fig. 12 shows the (simulated) behavior of a power source with a first increasing then again decreasing voltage. The (measured) behavior of an ESB is as follows (input current shown). As soon as the voltage reaches a certain level, the circuitry performs a clean reset to enter a well-known state. After that, the sensor node waits until the available voltage is high enough to start periodic sensing of the stored energy. This is done by periodically drawing a certain current and measuring the behavior of the energy source (short peaks in the lower line). If the stored energy seems to be large enough, the node starts sending (high peaks). If the voltage

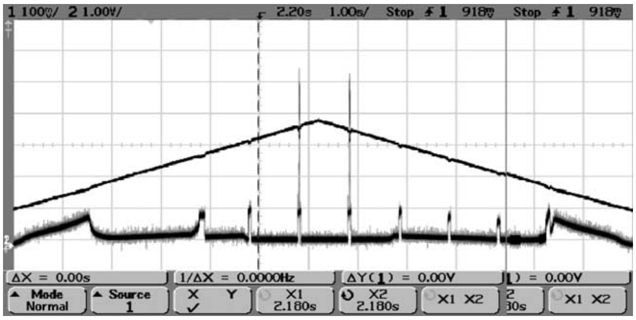

Fig. 12. Adaptive power management.

drops again the node enters the periodic checking mode again. If the voltage further drops below a certain threshold the node switches itself off (final peaks in the lower line).

We use gold-cap capacitors for storing energy and solar cells, piezo crystals, or thermo-elements for energy scavenging. With the 1 F capacitors we use, e.g. we can perform over 420 sensing and sending cycles without any additional power supply. Thus, the stored energy in such a capacitor is more than enough to keep the sensor alive over night for typical monitoring applications.

4. Low-power image transmission

Up to some years ago, no one would have given a thought to low-cost, low-power image transmission. However, due to the tremendous success of mobile phones with built-in cameras and the advances in CMOS technology the price for low-power VGA-resolution cameras is continuously dropping. Thus, it seems quite natural to include image capturing and transmission capabilities in low-cost sensor networks as monitoring an environment quite often also requires a visual impression. One hurdle was the integration of camera modules plus the image compression software needed to produce compressed images for storage and transmission. For our sensor nodes including a camera we use the C328-7640 VGA camera modules from COMedia [3]. The board comprises a VGA camera chip, a JPEG compression engine, and a serial interface. Via the serial interface, users can send out a snapshot command in order to capture a full resolution still picture. The picture is then compressed by the JPEG compression engine and transferred back via the serial interface. The whole board operates at 3.3 V and measures 20×28 mm. The maximum resolution is 640×480 pixels; however, the module can be configured to downsample the image to 80×64, 160×128, or 320×240 pixels. The transfer rate over the serial interface is 115.2 kbit/s. The built-in color conversion hardware provides 2/4/8 bit gray scale or 12/16 bit color images. An additional interesting feature of the module is the capability to stream a 160×128/8 bit preview video with 0.75–6 frames per second.

We attached the camera module to our ESB430 and ECR430 boards via a simple serial interface running at 115.2 kbit/s. Fig. 13 shows a sample image taken with an

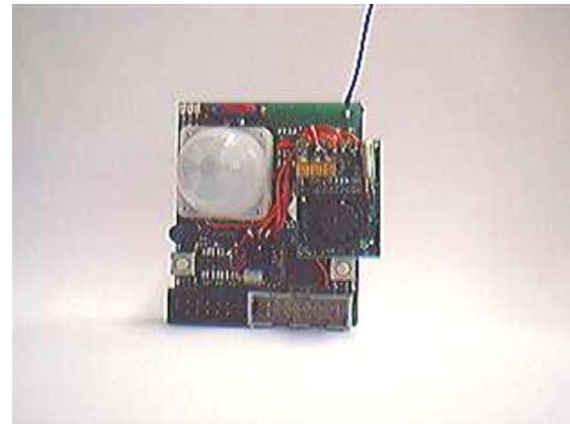

Fig. 13. Sample image taken from the prototype ESB/Camera module facing a mirror (320×240 pixel, 6304 B).

ESB/430 module facing a mirror. Up to now we have only built prototypes and we are currently designing new sensor nodes integrating the camera module.

An active camera module consumes 50 mA at 3.3 V. Our software allows for the activation and deactivation of the camera module. The power consumption of the camera module drops to 100 μA in power down mode. Using our ESB or ECR sensor nodes we can even further save energy by switching off the complete camera module by simply switching off the power supply. As we connect the camera module over a serial interface with 115.2 kbit/s, downloading compressed pictures from the camera to the sensor board takes only about 2 s even at the maximum resolution. JPEG compression on board of the camera module results in picture sizes of 20–30 KB for 640×480/16 bit.

One solution implemented on our sensor nodes to transfer images copies the image data from the camera module to the built-in 64 KB EEPROM first and then starts transmitting the image over the air including retransmissions of lost image data if requested (it is possible to use larger EEPROMs if required). Fig. 14 shows a typical cycle of image transmission. First of all, the sensor node commands the camera to start the image transfer into the EEPROM. This transfer takes about 2.5 s and consumes 56 mA. After that, the sensor node puts the camera module in low-power mode or switches

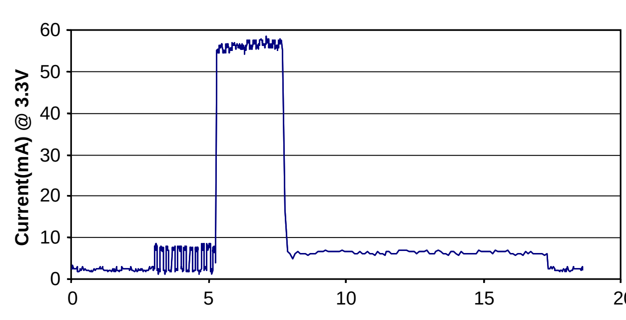

Fig. 14. Power consumption of the sensor node with camera while taking and transmitting a picture.

the module off. Then the transmission takes place at 7 mA and takes about 9.6 s. This results in an over all power consumption of 0.058 mA h per transmitted picture. Assuming a typical rechargeable AA battery with a capacity of 2000 mA h and a usable capacity of 80% a sensor can transmit about 27,500 pictures. As sending 20–30 KB images requires about 12–17 s between 3 and 5 pictures per minute are possible to capture and to transmit.

5. Related work

Since sensor network are one of the current ‘hype’ topics, several systems evolved over the last years-each with specific pros and cons. The most prominent system in academia is the Berkeley family of motes that are marketed and manufactured by Crossbow [4]. Closer to industry requirements are sensor network systems from Helicomm [12], Ember [9], or Ekasystems [8], just to name a few. These companies offer different gateways to industry control systems and more robust components compared to nodes used for research and development.

6. Conclusion and outlook

This paper introduced ScatterWeb, an open and flexible platform for teaching sensor networks and implementing real world sensor applications. Major goals of ScatterWeb are the design of low-power sensor nodes, the simple management of large-scale sensor networks, and the seamless integration into other wired or wireless networks by using Internet protocols and by providing several gateways. Furthermore, ScatterWeb serves as platform for the implementation of energy-aware routing algorithms [18]. This allows for the derivation of real-world parameters for simulation models. One example of such energy-aware algorithms is the extension of directed diffusion we implemented on our sensor nodes. Simulations and real implementations both show the benefits of energy-awareness. Furthermore, we showed that even the integration of a camera module and image transmission are possible with very low-power consumption.

Currently, we compare the usage of new, faster transmitters allowing for higher transmission speeds with slower, but more robust transmission. Additionally, we extend the features of the sensor nodes depending on project requirements: special modules for energy scavenging, new RF transceivers, voice output and speech recognition, etc. Furthermore, we extend the existing software by applications for localization, for integration into building management systems, or for graphical programming of sensor networks.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.
[2] Chipcon AS, www.chipcon.com, 11-2004.
[3] COMedia Ltd, www.comedia.com.hk, 11-2004.
[4] Crossbow Technology Inc., www.xbow.com, 11-2004.
[5] The Contiki Operating System, www.sics.se/~adam/contiki/, 11-2004.
[6] A. Dunkels, uIP—a TCP/IP stack for 8- and 16-bit microcontrollers, dunkels.com/adam/uip/, 11-2004.
[7] J. Dunkels, A. Alonso, T. Voigt, H. Ritter, J. Schiller, Connecting wireless sensornets with TCP/IP Networks, Wired/Wireless Internet Communications, Second International Conference, WWIC 2004, Frankfurt/Oder, Germany, 2004.
[8] Eka Systems, www.ekasystems.com, 11-2004.
[9] Ember, www.ember.com, 11-2004.
[10] EnOcean GmbH, www.enocean.de, 11-2004.
[11] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, International Conference on System Sciences, Hawaii, USA, 2000.
[12] Helicomm, www.helicomm.com, 11-2004.
[13] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, Wireless sensor networks for habitat monitoring, First ACM Workshop on Wireless Sensor Networks and Applications (WSNA 2002), Atlanta, USA, 2002.
[14] RF Monolithics, Inc., www.rfm.com, 11-2004.
[15] S. Roundy, P. Wright, J. Rabaey, Energy Scavenging for Wireless Sensor Networks, Kluwer, Boston, 2003. ISBN 1-4020-7663-0.
[16] The ScatterWeb project, www.scatterweb.net, 11-2004.
[17] TinyOS Community Forum, www.tinyos.net, 11-2004.
[18] T. Voigt, H. Ritter, J. Schiller, Solar-aware routing in wireless sensor networks, Personal Wireless Communications, PWC 2003, Venice, Italy, 2003.
[19] A. Willig, R. Shah, J. Rabaey, A. Wolisz, Altruists in the picoradio sensor network, Factory Communication Systems, Västerås, Sweden, 2002.
[20] F. Ye, G. Zhong, J. Cheng, S. Lu, L. Zhang, PEAS: a robust energy conserving protocol for long-lived sensor networks, International Conference on Distributed Computing Systems, Rhode Island, USA, 2003.
[21] Y. Zhao, R. Govidan, D. Estrin, residual energy scan for monitoring sensor networks, Wireless Communications and Networking Conference, Orlando, USA, 2002.

Prof. Dr.-Ing. Jochen H. Schiller (schiller@inf.fu-berlin.de) is head of the working group Computer Systems & Telematics at the Institute of Computer Science, Freie Universitaet Berlin, Germany. Dr. Schiller studied Computer Science at the University of Karlsruhe where he also got his PhD in 1996 (summa cum laude). Afterwards he joined Uppsala University, Sweden, for teaching and cooperating in several research projects. After being a guest professor at ETS Montreal, Canada, and Kiel University, Germany, Dr. Schiller got his habilitation and published the book “Mobile Communications” that is currently used by over 200 universities as text book. Since April 2001 he is full professor at the Freie Universitaet Berlin, Germany, and since 2002 he is Vice Director of the Institute of Computer Science. In April 2003 Dr. Schiller became Dean of the Department for Mathematics and Computer Science. His research focus is on wireless, mobile, and embedded devices, communication protocols, operating systems for devices with small footprint, and quality of service aspects in communication systems. Up to now, Dr. Schiller published 5 books and more than 70 international papers. His book “Mobile Communications” is available in 5 languages.