Communication of autonomous agent systems based on wireless sensor motes

ANDREJ BENCÚR, M.Sc.[1]
Prof.MIROSLAV POKORNÝ, Ph.D.[1]
Dr.JAN SMID[2]

[1]Department of measurement and control
Technical University of Ostrava
17.listopadu 15, 708 33 Ostrava
CZECH REPUBLIC

[2]Morgan State University
Department of computer science
Cold Spring Lane, Baltimore 21251-0002
USA

Abstract: - This paper describes a project that deals with a network of autonomous agents. The agents are represented by Internet Embedded Devices (IED). Miniaturization of hardware and growing computing power of embedded devices allows us perform sensing and data conversion using microcontrollers and/or wireless motes. The embedded web server provides the interface for direct data transmission.

Key-Words: - Internet, sensors, communication, autonomous agents, language, intelligence, server, client

1 Introduction

The data sharing is important. Every device needs to communicate with other units, provide and serve its own measurements and sensed events, and receive information from distant systems.

The whole network can contain various nodes with different capabilities. One of them performs basic data readout from sensors on Parallax's Board of Education development board and their broadcast over the Internet. For this purpose Red-I web server is used.

The second level in the current concept is based on MICA motes from CrossBow Technologies, Inc. This platform and its application will be described below.

Also we will comment on more complicated devices, such as embedded Linux-based Stargate developed by Intel Berkeley.

2 Intelligent agent networks

The main task of the project is to develop a network of autonomous agents that use language-based communication to perform useful tasks. Generally, this network contains servers, clients, agents, robots, sensors and effectors. A highly important property of such system is its flexibility in knowledge acquisition, exchange and incorporation of new members. Four layers have been defined as follows:

- Hardware layer
- Data transport layer
- Knowledge (re)presentation
- Application

At the beginning we have been dealing with the hardware layer first. Appropriate technologies had to be found for proper functioning such as standalone devices with sensing and communication capabilities. We will concentrate below on the Crossbow's MICA motes, their hardware, radio and sensing capabilities.

3 Wireless sensor networks

Well known problems with wired networks, such as interference, physical limitations etc. are the main reasons for growing demand for wireless networking and data communication.

3.1 MICA radio platform

Common hardware platform is used as node's base. Every base is constructed of similar parts. In the center lies Atmel ATMega128 8-bit microcontroller with a tunable Chipcon radio. Depending on the transceiver, different frequencies can be used. In our work so far

CC1000 and CC2420 chips have been used. Microcontroller is equipped with 128kB of Flash, 4kB of RAM and 512kB of serial EEPROM memories [1].

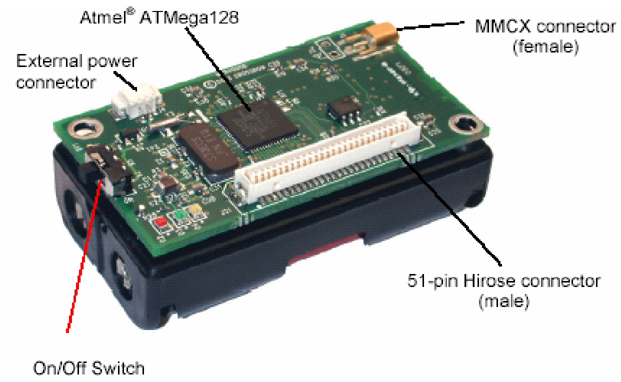

Fig.1: MICA2 communication platform

A block diagram of MICA2 platform can be seen in the next figure:

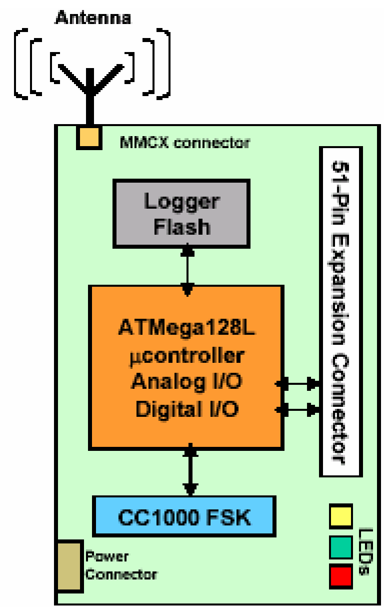

Fig.2: Block diagram of MICA2 mote.

3.2 TinyOS operating system

The installed processor and memory are powerful enough to run TinyOS, which is an open-source operating system developed at the Berkeley University and designed for wireless embedded sensor networks. It features a component-based architecture which enables fast implementation and innovation. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools, which can be used as-is or be further refined for a custom application. TinyOS's event-driven execution model enables power management yet allows scheduling flexibility necessary by the nature of wireless communications and physical world interfaces.

The TinyOS system, libraries and applications are written in nesC, a language for programming structured component-based applications. nesC is primarily intended for embedded systems, such as sensor networks. nesC has a C-like syntax, but supports TinyOS concurrency model, as well as mechanisms for structuring, naming and linking together software components into robust network embedded systems. nesC applications are built out of components with well defined, bi-directional interfaces. Also nesC defines a concurrency model, based on tasks and hardware event handlers, and detects data races at compile time [2].

3.2.1 Surge multi-hop networking

The TinyOS-1.1.0 release and later include library components that provide ad-hoc multi-hop routing for sensor network applications. The implementation uses a shortest-path-first algorithm with a single destination node (the root) and active two-way link estimation. The data movement and route decision engines are split into separate components with a single interface between them to permit other route-decision schemes to be easily integrated in the future. Use of the multi-hop router is essentially transparent to applications (provided they correctly use the interface).

Crossbow's mesh networking resides as a software component in TinyOS. Typical applications that run on the Motes use several different software components - *e.g.*, sensor components, data logging, and more. If the application is going to use mesh networking, it will link in the *MultiHopRouter* software component. The implementation of the *MultiHopRouter* component found in Crossbow's TinyOS contribution directory after installation includes some significant performance upgrades, and it is a state-of-the-art mesh networking application [3].

3.3 Sensor boards

MTS300/310CA are sensor boards with a variety of sensing modalities. They can communicate with Atmel microcontroller through GPIO's connected to a 51-pin HIROSE connector. Any user board can be connected with such connector and provide specific user functions. However, we use CrossBow's MTS300 and MTS310 boards with built-in peripherals:

- Magnetometer
- Accelerometer
- Microphone
- Sounder
- Temperature sensor
- Light sensor

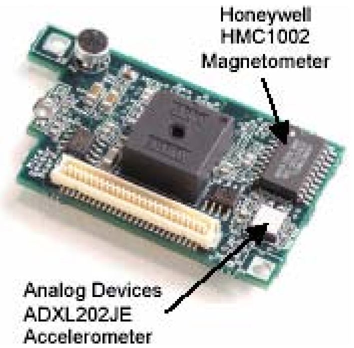

Fig.3: MTS310CA sensor board

3.4 Ethernet and programming gateway

For programming of above mentioned devices and for their access to local network, MIB600 Ethernet gateway from CrossBow is used. This board provides socket for connecting any of MICA family motes, which serves as gateway for all other motes in a wireless networks and broadcasts their data over TCP/IP connection.


Fig.4: MIB600 Ethernet and programming gateway

In chapters 3.1 to 3.4 basic components of wireless sensor networks are described. In actual work there is a change in radio platform used. MICA2 has been exchanged with MICAz radio running on Chipcon CC2420 transceiver, which is specifically designed for ZigBee networks according to IEEE802.15.4 standard offering significantly higher data throughput (250kbps compared to 76.8kbps), better power-saving features and new multi-hop networking algorithms [1].

4 Getting wireless sensor network online

To bring the wireless sensor network online, few steps have to be taken and the communication chain must be understood. In the Fig. 5 all the parts of such a system are shown.

Data from every node should be accessible by any other distant client. Since not all nodes in this sensor network are capable of direct communication with other “high level” nodes requiring data, a PC with a TCP client, php script and web server servers for this purpose.

4.1 MICAz radio platform, sensor boards and Ethernet gateway

See chapters 3.1, 3.3 and 3.4

4.2 PC interface

4.2.1 TCP/IP client

This application programmed in Microsoft Visual C# .NET has two main tasks to process.

The first one is connecting to a specific TCP port from which it can read sensor data coming periodically from MIB600 Ethernet board.

The second task is to filter the unnecessary data out, format them in appropriate way and store them into a text file for further processing.

4.2.2 PHP script

PHP script running on web server continuously reads data from a temporary database in the text file and reloads web page with the newest information.

4.2.3 Web server

Apache web server 1.3.27 with PHP 4.2.3 support provides Internet access to data from local sensor network.

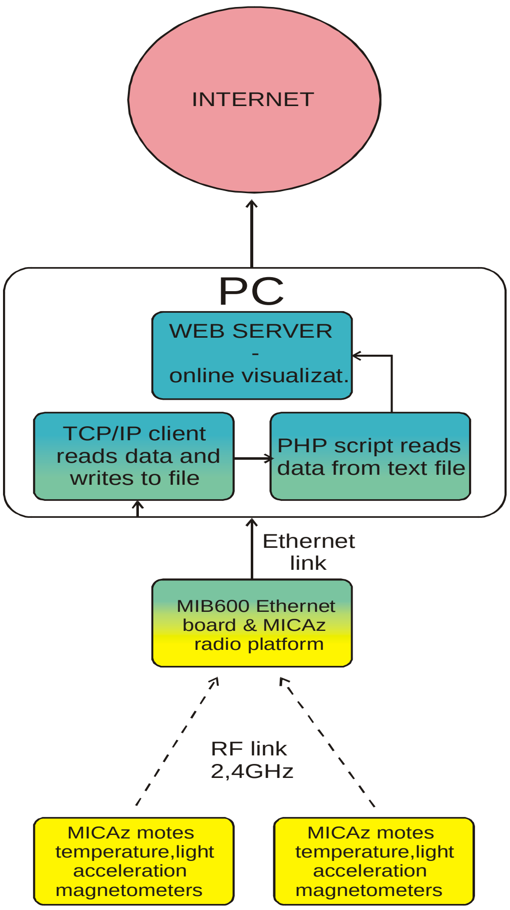

Fig.5: Communication chain

5 The four-layer model

Good and necessary practice in designing communication systems and protocols is to organize different system functions into layers.

The layers are described in more details below.

5.1 Layer 1 - Hardware layer

Another four classes can be placed here:

- Class 1 – Server with TCP/IP client-server functionality
- Class 2 – Active subserver hosted on server (not designed for the task, full functionality)
- Class 3 – Passive server (limited client functionality)
- Class 4 – Sensor server (only posts data)

Hardware components may consist of the following:

- MICA2 and MICAz sensor networks running TinyOS operating system and TinyDB database engine
- Basic Stamp 2 microcontroller with a memory and Basic-programming
- Javelin Stamp module – microcontroller with a memory and Java-subset programming
- Other microcontroller-based systems and Internet Embedded Devices

5.2 Layer 2 – Data distribution

This layer deals with transmission and data routing using TCP/IP, UDP server-client connections, PHP files, remote script execution, routers, Internet Embedded Servers etc.

5.3 Layer 3 – Knowledge re(presentation)

One important algorithm is based on the semantic mapping problem:

Utterance	Meaning representation
foo bar	LIST(**temperature**)
baz bar	LIST(**humidity**)
quux bar	LIST(**acceleration**)
quux bleen plugh baz	COPY(**humidity**,TO(**acceleration**))
baz bleen cruft foo	COPY(**humidity**,FROM (**temperature**))

Table 1: Mapping problem

The solution to this illustrative toy mapping problem is:

Utterance	Meaning representation
bar	LIST(x)
foo	**temperature**
baz	**humidity**
quux	**acceleration**
bleen	COPY(x,y)
plugh	TO(x)
cruft	FROM(x)

Table 2: Solution to the mapping problem

Consequently for example utterance "Meeting tomorrow at 8:00" can be processed by taking the following steps:

- Use lexicon
- Analyze semantic frame
- Apply information gathered from previous references
- Understand meaning

5.4 Layer 4 – Applications

A goal of this layer is to compile tables of temperatures and time tags, be able to answer queries regarding temperature values at different places and different time, and acquire new concepts such as trend of temperature.

These concepts are not hard-wired but learned using provided semantic information [4].

6 Conclusion

In this paper the first steps of our project that deals with what we call concept and sensor networks were presented.

We focused on the wireless sensor network of motes. Data provided by our sensors and agents are accessible On line on the Internet

The next step will be implementing database systems into low-level nodes and programming active servers on PCs and embedded computers (Stargate).

Acknowledgments:

This work has been supported by the GACR project 102/05/H525: "The Postgradual Study Rationalization at Faculty of Electrical Engineering and Computer Science VSB-TU Ostrava".

References:

[1] CrossBow technology, *Wireless Sensor Networks – MPR/MIB User's manual,* Revision A, August 2004

[2] UC Berkeley TinyOS Community, *www.tinyos.net*, *http://www.tinyos.net/tinyos-1.x/doc/tutorial/*
[3] CrossBow technology, *Wireless Sensor Networks – Getting started guide*, Revision B, August 2004
[4] Jan Smid, *McDaniel College Presentation – Robots, Minds and Networks*, January 3, 2005