Connecting Contiki enabled Versatile Sensor Nodes via CC1101 radio

Zoltan Padrah, Ales Verbic, Marko Mihelin, Carolina Fortuna, Mihael Mohorcic
Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
firstname.lastname@ijs.si

Abstract—In this demo, we show message exchange between Versatile Sensor Nodes (VSN), which run the Contiki operating system and use the Texas Instruments CC1101 radio. We explain our port of the Contiki on VSN with the emphasis on the wireless communication. VSN is a modular sensor node platform featuring several communication interfaces. In the demo we use the CC1101 radio to connect VSNs and RS232 to connect VSNs to PCs. The application used to illustrate the functioning of the system sends messages from a computer and displays them on the screen of all other computers attached to VSNs that received that message.

I. INTRODUCTION

Wireless sensor networks are increasingly used to monitor certain phenomena or activities for longer time periods and/or in remote locations. Such networks are composed of wireless sensor nodes and sensors that are attached to them. These sensors measure various physical quantities which are transmitted via sensor nodes to the processing infrastructure.

Generally, such nodes consist of a microcontroller, a radio transceiver, power source, sensors, and in some cases external memory. The microcontroller runs the software of the sensor node and controls all the other components of the node. Usually it contains the CPU, embedded RAM and flash memory, and a large number of peripherals, including analog to digital and digital to analog converters. The latter components are useful for retrieving the data from analog sensors. The radio transceiver is used for communication between the nodes. As sensor nodes are expected to operate autonomously for long periods of time, power efficiency is critical. Most common power sources are batteries and solar cells.

There are two main categories of software used on wireless sensor nodes: custom software and operating systems. The latter ones provide a set of hardware and application independent functionalities at the expense of the maximum theoretically achievable performance. As examples of most widely used operating systems for wireless sensor nodes are TinyOS and Contiki.

Currently there are several commercially available sensor nodes: custom software and operating systems. The latter ones provide a set of hardware and application independent functionalities at the expense of the maximum theoretically achievable performance. As examples of most widely used operating systems for wireless sensor nodes are TinyOS and Contiki.

There are two main categories of software used on wireless sensor nodes: custom software and operating systems. The latter ones provide a set of hardware and application independent functionalities at the expense of the maximum theoretically achievable performance. As examples of most widely used operating systems for wireless sensor nodes are TinyOS and Contiki.

II. VERSATILE SENSOR NODE

The wireless sensor node presented in this demo has been designed to be a multipurpose sensor node platform: it can be easily adapted to many applications, due to its modular design and the available expansion connectors. The main module of the node is called the Versatile Sensor Core module (VSC) which hosts the microcontroller. The block diagram of this module is depicted in Figure 1.

The versatile sensor node employs a STM32F10x microcontroller, having a 32-bit ARM Cortex-M3 processor core. The microcontroller has 64 kB of RAM and 512 kB of flash memory, and supports a variety of interfaces: USB, RS232, UART, IrDA, SPI, I2C, 12 bit ADC, DAC.

A presentation about the VSN can be found online at http://videolectures.net/wsn2010_mihelin_vsn/
The core module of the versatile sensor node has two expansion connectors: one designed for radio modules, and one for various purposes. As radio expansion modules, sub-gigahertz and 2.4 GHZ-band transceivers are available (i.e. using CC1101 and CC2500 radio chips), but alternatively external XBee or Bluetooth modules can be used as add-on boards to the radio transceiver modules. As examples of various expansion boards we mention the debugging and programming board, Ethernet to serial converter, WiFi to serial converter, prototyping modules and the additional power supply module.

Software development for the versatile sensor node can be performed by using standard debug protocol (JTAG) and open-source tools (OpenOCD, CodeSourcery G++ lite and Eclipse). In this demo, we use the radio module with CC1101 chip from Texas Instruments, in the 868 Mhz ISM band.

III. PORTING CONTIKI OPERATING SYSTEM TO VSN

On the VSN platform we decided to use Contiki OS [2], which offers many services and functionalities to the applications running on it, two of which are particularly important for this demo. First, because the operating system supports multi-tasking, various functionalities can be separated to different tasks. For example a radio driver can be separated from the main application. This simplifies the structure of the program. The other aspect is the large range of supported communication protocols, starting from a simple serial line driver, to various networking protocols such as uIP and TCP.

As we mentioned before, Contiki is highly portable. This portability is achieved by separating all the hardware dependent code from the rest of the system. In order to support new hardware, just a set of interfaces have to be implemented. Implementing the hardware-specific functions of the clock module and integrating the port with the Contiki’s build system makes the base system functions (processes and basic timers) work. Input from the RS232 line is handled by a hardware dependent code path feeding the generic serial-line module. Output to RS232 is implemented by defining the hardware-dependent functionality of a function (_write()), used by the C library of the compiler. Contiki has a dedicated interface for radio drivers. This interface provides functions for sending packets, signaling if a packet has been received, reading any received packet and turning the radio on and off. Our CC1101 driver implements such interface, so in the future it can be easily used with different communication stacks.

IV. THE DEMONSTRATION

The demo uses the CC1101 radio to connect VSNs and RS232 to connect VSNs to PCs. The application used to illustrate the functioning of the system sends messages from a computer and displays them on the screen of all other computers attached to VSNs that received that message. In this way we demonstrate the communication capabilities of our node and the working Contiki operating system port. The block diagram of the setup is depicted on Figure 2.

The application, that runs on the top of Contiki passes each message received on one communication channel (RS232 or radio) to the other communication channel.

Figure 3 depicts the demo setup. The user interface of the chat application consists of a terminal emulator running on a PC. Initially only a prompt character(>) is displayed. Messages can be entered by using the keyboard of the PC, and they can be sent by pressing the Enter key. Each message will be sent to all the nodes in the transmission range of the radio. After a message is sent, a new prompt character should appear. Received messages appear on the terminal as new lines of text. However because the application does not use a protocol that guarantees message delivery, it is possible that a message will be lost.

In order to make distinction between different sources of messages, a node identifier is automatically added to each message. This helps the users of the system in communication, by allowing them to easily identify the sender of each received message.

REFERENCES
