Software and Hardware Design of a Miniaturized Mobile Autonomous Robot, operating in a Wireless Sensor Network

Tobias Glocker, Matti Tuomaala, Reino Virrankoski, Mohammed Elmusrati

Communications and Systems Engineering Group, University of Vaasa
Wolffintie 34, 65200 Vaasa, Finland

tobias.glocker@student.uwasa.fi
matti.tuomaala@student.uwasa.fi
reino.virrankoski@uwasa.fi
mohammed.elmusrati@uwasa.fi

Abstract—*Nowadays wireless nodes are becoming more and more popular in the field of localization. Thanks to the high research effort in this area, wireless sensors become more and more sophisticated. From year to year the accuracy in terms of distance estimation increases. In comparison to other localization devices like a Local Positioning System (LPS) or Global Positioning System (GPS), the wireless nodes are considered as a cheap alternative. The Finnish defence department, police and fire department support current research activities within this area, in the hope that they will get beneficial applications.*

Index Terms—*WSN, Miniaturized Mobile Autonomous Robot, IEEE 802.15.4, NanoStack*

1. Introduction

In recent years, the development of wireless sensor networks has caused an exponential growth. Wireless Sensor Networks (WSN) have led this trend due to the reduction of the noise level with high sophisticated antennas and better coding techniques, that achieve a better accuracy in distance estimation and localization. In addition, Wireless Sensor Networks are extendable. According to [1] "[they] can be deployed on a global scale for environmental monitoring and habitat study, over a battle field for military surveillance and reconnaissance and in emergent environments for search and rescue".

Common localization methods used for autonomous robots are Local Positioning Systems (LPS) and Global Positioning Systems (GPS). Local Position Systems and Global Positioning Systems have a quite good accuracy but both are very expensive in comparison to a Wireless Sensor Network consisting of many Wireless Sensor Nodes.

The aim of this thesis, was to design and build an miniaturized autonomous robot for the Wireless Sensor Systems in Indoor Situation Modeling (WISM) project. One part of this project deals with distance estimation and localization within a Wireless Sensor Network. To evaluate already developed Algorithms for distance estimation and localization based on their precision, the constructed miniaturized autonomous robot should be guided inside a WSN, as shown in Fig. 1. The static nodes are deployed in a two dimensional environment and they first localize themselves. When a sensor of a static node detects an event, it will send a message to the robot (mobile node). Since the estimated locations of the static nodes are known, the robot needs to compute a path which leads it to the target node.

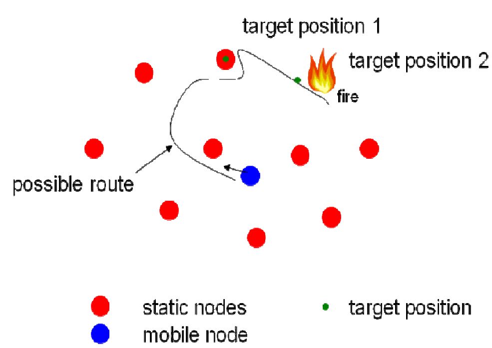

Fig. 1. Wireless Sensor Network with static nodes and mobile node.

2. Hardware Design Of The Robot

The robot consists of a mobile platform, a wireless sensor node and an embedded Linux PC. The Hardware of the Mobile Platform is described in section 2.1. In the following sections the hardware of the wireless sensor node and the hardware of the embedded Linux PC will be explained. Section 2.4 shows how the hardware parts are connected together.

2.1. Hardware of the Mobile Platform

The mobile platform (see Fig. 2) used in this thesis is a product of Matrix Multimedia [2]. It mainly consists of a PIC18F4455 microcontroller, a motor driver chip (L293D), a microphone with sound level amplifier circuit, three distance sensors, a light sensor, eight user definable LEDs, an external 5V power supply and a Universal Serial Bus (USB) interface for programming the microcontroller. Four AA batteries inside the plastic chassis supply the two motors and the circuit board with power. The maximum speed that can be reached is 20 cm per second.

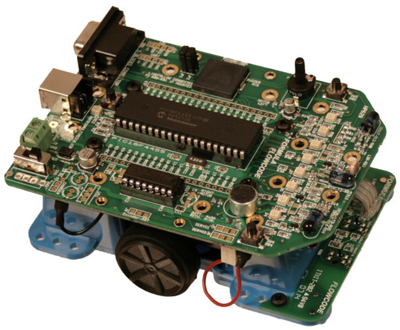

Fig. 2. Formula Flowcode Buggy.

The microcontroller (PIC18F4455) [3] is produced by microchip and handles up to 12 Million Instructions Per Second (MIPS). It provides an enhanced USART interface, a SPI interface and Inter-Integrated Circuit (I²C) interface. Four Timer modules, an Enhanced Capture/Compare/Pulse Width Modulation (ECCP) module and a 10 Bit Analog-to-Digital Converter (ADC) are integrated in this chip.

2.2. Hardware of the Wireless Sensor Node

The wireless nodes are manufactured by Sensinode in Oulu, Finland. A wireless node consists of an integrated Radiocrafts RC2301AT module [4] with the size of 12.7 x 25.4 x 2.5 mm. This module is complete ZigBee-ready and has an IEEE 802.15.4 transceiver. It is a so called System on Chip (SoC) module containing a high performance 8051 microcontroller core with 128 kilo byte (kB) flash memory, 8 kB Shadow Random Access Memory (SRAM), 4kB Electrically Erasable Programmable Read-Only Memory (EEPROM) and an 8 channel 14 bit Analog-to-Digital Converter (ADC). For the communication with the outer world, the module provides 19 digital and analog Input/Output pins, a Universal Asynchronous Receiver/Transmitter (UART), Serial Peripheral Interface (SPI) and a debug interface. Fig. 3 displays Sensinode NanoSeries N100 module.

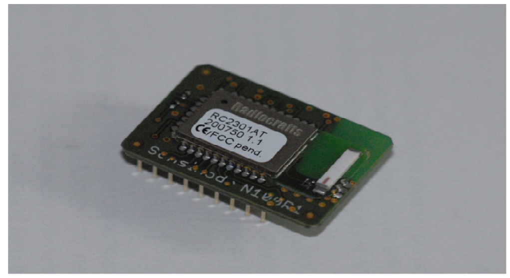

Fig. 3. Sensinode NanoSeries. Node Design N100.

2.3. Hardware Platform of the Embedded PC (Portux 920T)

Portux 920T [5] is a single board computer with an AT91RM 9200 CPU from Atmel. CPU-Modules with ARM architecture are suitable for embedded systems with high performances and low power consumption. The size of the Portux board is 100 x 71 mm. With a clock rate of 180 MHz the ARM processor can solve complex computations within a short time interval. It also offers a variety of integrated peripherals such as USB 2.0, Ethernet, Serial Peripheral Interface (SPI) and four Universal Synchronous and Asynchronous Receivers/Transmitters (USARTs). Both flash memory (16 MB) and Random Access Memory (RAM) (64 MB) are big enough to run embedded Linux on this ARM processor. The baseboard comes with a 10/100 Mbit/s Ethernet interface and with two serial interfaces. It contains a Secure Digital (SD)/Multimedia (MMC) Card Slot for memory extension. Additional peripherals can be connected with the Portux Extension Bus (PXB). Fig. 4 gives an overview of the main interfaces on the base board.

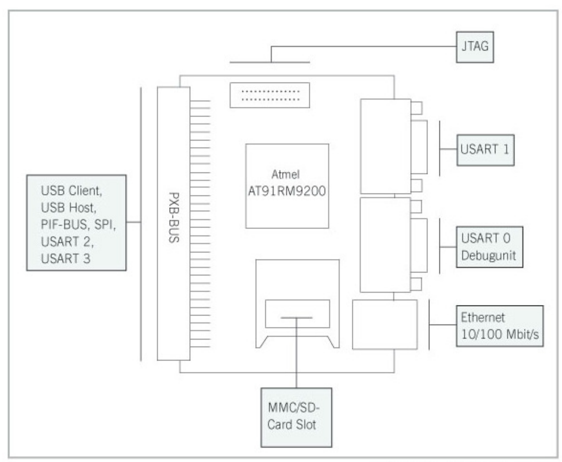

Fig. 4. Overview of the main interfaces on the base bord Portux920T.

2.4. Hardware Design of the Miniaturized Mobile Autonomous Robot

One task of this thesis was to design a circuit board that connects the hardware components mentioned in the previous sections. The Embedded PC acts as Master Device that communicates with the wireless node and the microcontroller on the mobile platform via Universal Asynchronous serial Receiver and Transmitter (USART). An overview of the circuit board is given in Fig. 5. USART 3 of the embedded PC is directly connected with the UART of the wireless sensor node. Both devices work with the same voltage level (3.3V). USART 2 is used for the connection between Embedded PC and the microcontroller (PIC18F4455) on the mobile platform. Due to different voltage levels, the transmitter of the microcontroller cannot be connected directly with the receiver of the Embedded PC. Thus it was necessary to build a voltage regulator that regulates the microcontroller's transmitter output from 5V to 3.3V. An 11.1V Li-PO battery with 1300mAh supplies the circuit board with power. The Embedded PC has its own voltage regulator but not the wireless node. For that reason a voltage regulator board was developed to reduce the battery's 11.1V output voltage to 3.3V. There are also four push buttons on the board that are connected with the Embedded PC. To display messages and errors the circuit board contains also a 16 character, 2-line alphanumeric Liquid Crystal Display (LCD).

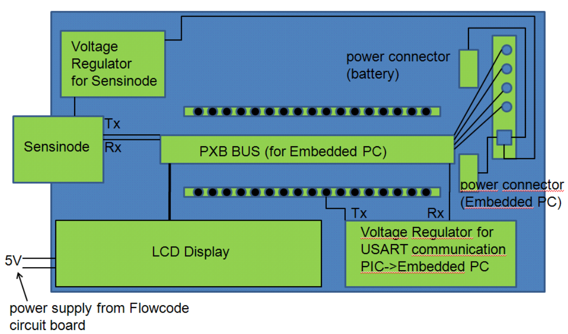

Fig. 5. Overview of the designed circuit board that connects the Embedded PC with the microcontroller (PIC18F4455) on the mobile platform and with the wireless sensor node.

3. Software Design Of The Robot

Each device mentioned in the sections 2.1, 2.2 and 2.3 requires a software implementation. The program running on the microcontroller PIC18F4455 is responsible for reading the values of the infrared sensors and for the control of the motor drivers. The program that runs on the wireless node located on the circuit board takes care of the communication between Embedded PC and other wireless sensor nodes (static nodes). On the Embedded PC, the program needs to deal with mobility computation, distance estimation and prediction filters. Fig. 6 represents the functionalities assigned to each device.

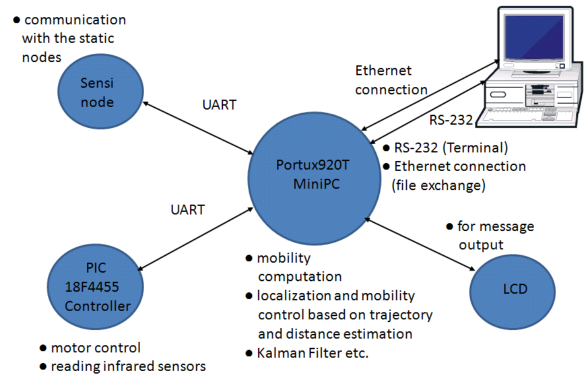

Fig. 6. Representation of the functionalities assigned to each device.

3.1. Software Design of the Mobile Platform

In comparison to the Embedded PC, the microcontroller on the mobile platform is quite slow. For that reason the program running on that microcontroller must be kept simple. Simple means that the program does not contain heavy computations, so that the execution time of one loop iteration is short. As already mentioned in section 2.4 the microcontroller is connected with the Embedded Linux PC via USART and it receives periodically a packet that contains a start byte, direction byte, two bytes for the speed of the motors and one byte for the checksum. Fig. 7 illustrates the packet types. After receiving a correct packet the program sets the duty cycle of the Pulse Width Modulation (PWM) according to the desired wheel speed and runs the motors. Before the robot starts to move the infrared sensor checks if there are no obstacles in front of it.

Packet Format Embedded PC → Microcontroller (PIC)

START	Direction Byte	Speed Motor Left	Speed Motor Right	Check-sum

5 Bytes

Packet Format Microcontroller (PIC) → Protocol Embedded PC

START	Infrared Value Left	Infrared Value Forward	Infrared Value Right	Check-sum

5 Bytes

Fig. 7. Packet types between Embedded PC and PIC18F4455.

3.2. Software Design of the Wireless Node

The main task of the program running on the wireless node is to receive packets from other wireless sensor nodes and forward them via UART to the Embedded PC. There are two different states in that program. First the program is in the configuration state in which the wireless node on the robot can communicate with the wireless node that forwards/receive packets from/for the Graphical Control User Interface (GCUI) shown in chapter 4. In the second state the wireless node on the robot is only able to receive packets.

3.3. Software Design of the Embedded Linux PC

3.3.1 Embedded Linux: On the embedded PC (Portux920T) runs an embedded Linux operating system. The source code for embedded Linux is completely open and can be changed by anyone. Programmers can develop Linux applications and sell it without paying any sort of royalty [6]. It is also possible to make changes to Linux but all the changes made to the Linux kernel must be shared with the rest of the world. Linux is also famous because of its stability and good documentation. Many documents are available on the internet and dozens of books can be found in book stores. Beside the previous mentioned benefits, Embedded Linux provides also benefits related to the application development with concurrency. Concurrency means that several computations are executed simultaneously. In Embedded Linux concurrency can be achieved by creating multiple threads. Another advantage of Embedded Linux is that the application can be programmed with an Object Oriented Programming (OOP) language such as C++ and Java.

3.3.2 Software Design of the Embedded PC: When the Embedded PC is switch on the boot loader (UBoot) loads the embedded Linux. As soon as this process is finished a bash script will be executed that starts the configuration program. The configuration program communicates with a GCUI. It reads the configuration commands send from the GCUI and sets the parameter in the local configuration file. After the configuration is done the program can be terminated by pressing the first push button or by sending a termination command with the GCUI. Then the main program will be executed. It waits until a packet is received from the wireless sensor node. After the packet was received correctly the program checks if there are no obstacles in front or beside the robot by reading the Infrared Sensor values received from the microcontroller of the mobile platform. Then it computes the direction and the wheel speeds. A rough overview of the software flow is given in Fig. 8.

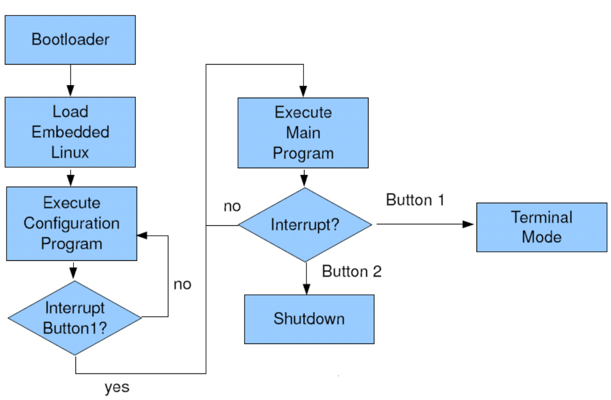

Fig.8. A rough overview of the software flow.

4. Graphical Control User Interface (GCUI)

To calibrate the robot a Graphical Control User Interface written in QT and C++ was developed. It provides a joystick control feature with that the robot can be controlled and which helps the user by adjusting the motor values. Later on there will be also features for receiving error messages and control messages from the robot. In addition, some configuration parameters can be set remotely. Fig. 8 shows the current state of the Graphical Control User Interface.

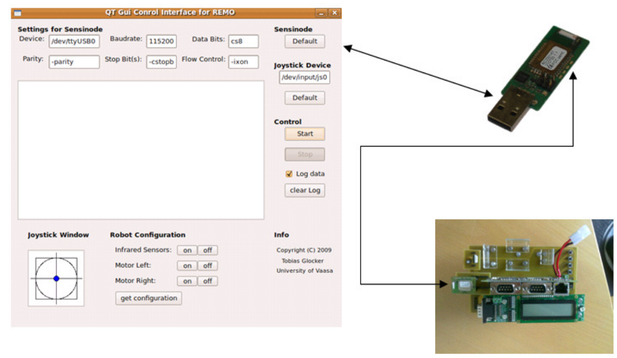

Fig. 8. Graphical Control User Interface.

5. Conclusion

Until now, already developed algorithms for distance estimation and localization are tested with Matlab and a Network Simulator called NS2. The target of this project is to develop a robot, acting as a mobile node in the indoor situation modelling system developed in the WISM project.

References

[1] Y. Yu, V. K. Prasanna, and B. Krishnamachari, *Information Processing and Routing in Wireless Sensor Networks,* World Scientific Publishing Co. Pte. Ltd., 2006, ISBN 981-270-146-X.

[2] Formula Flowcode, User guide. [Online]. Available: http://www.matrixmultimedia.com/datasheets/HP794-30-1.pdf

[3] (2009) PIC18F2455/2550/4455/4550 Data Sheet. 28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf

[4] (2007) ZigBee – Ready RF Transceiver Modules. RC230x Data Sheet (rev. 1.2) [Online]. Available: http://www.radiocrafts.com/uploads/rc230x_data_sheet_1_2.pdf

[5] Taskit Portux920T Overview. [Online]. Available: http://www.taskit.de/en/products/portux/index.htm

[6] J. Lombardo. Embedded Linux. New Riders, 2002. ISBN 0-7357-0998-X.