Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Development Of An Outdoor Mobile Robot For Teleoperation As An Agent For A Robot Network

Christof Hille, Ahmad Kamal Nasir, *Adrian Aria Abreu, *David López Piñeiro, Hubert Roth

Lehrstul für Regelungs und Steuerungstechnik (RST), Department of Elektrotechnik und Informatik, Fakultät IV, Universität Siegen, 57068, Siegem Germany (e-mail: {christof.hille, ahmad.nasir, hubert.roth}@uni-siegen.de)
**Dept. de Ingeniería de Sistemas, Universidad de Vigo, Campus Lagoas-Marcosende, 36200 Vigo, Spain (email: adrianariasabreu@hotmail.com, lopezdlp@gmail.com)*

Abstract: In this paper the development of hard and software modules for a mobile robot used for outdoor telecontrol is presented. The mobile robot is supposed to work as an agent in a network of mobile robots and its design is mainly for exploration in rough outdoor terrain or damaged buildings. Although the mobile robot is designed for outdoor missions, its components are designed in a way that they can be reused on most other mobile robot configurations. They allow a fast development of new robot agents, e.g. for specific tasks, with a quick and easy integration into the robot network. An Android app is presented to offer easy remote control of the mobile robot for an operator.

1. INTRODUCTION

Teleoperation has been an important research topic for several decades. Its focus is mostly the use of robots in areas where a human can't operate himself for a longer time like in polluted or dangerous areas or simply undersea (Ruangpayoongsak et al. (2005), Jamshidi and Eicker (1993), Sheridan (1992)) or in locations where humans are too large to operate like sewage water pipes (Li et al. (2007)). A lot of control techniques for this applications have been proposed and can be found e.g. in a survey from Schilling and Roth (2001).

One important aspect is how a mobile robot, e.g. for polluted or dangerous environments, should be designed for efficient operation during its missions. Although mobile robots for this purpose have been designed for years now, it's still not a solved area. The reason for this is the fast ongoing development of hard and software technologies. On one hand more and more powerful hardware is available like the new ARM CPUs which offer high computational speed at very low power consumption and small size and weight as discussed by Zhonghong et al. (2012) and on the other hand new data transmission technologies like Universal Mobile Telecommunications System (UMTS) and Long Term Evolution (LTE) offer less latencies in the data streams and much higher bitrates compared to older technology as stated by Sauer (2011). Therefore it's wise to reengineer hard- and software solutions.

In Nasir et al. (2012) a heterogeneous capability multi-robots cooperative framework to control numerous mobile robots of different structure and capability was presented. It enables users to remote control different types of mobile robots. Since this part is already solved, we focus in this paper on the evolution of our mobile robot Tracked MERLIN II (Mobile Experimental Robot for Locomotion and Intelligent Navigation), shown in figure 1 on the left side, to improve its functionalities and usability towards the framework and to offer more benefits to an operator. The existing design is evolved in its mechanical, electrical and software systems for an improved capability of outdoor operations. The new robot is developed as a standard ground agent for outdoor missions supervised by the software framework from Nasir (2012). The main idea of this development is that hard- and software modules are developed in a way that they can easily be integrated in a wide range of different mobile robots as well as in the software framework, to reduce development time and costs for future designs of mobile robots. The old design doesn't allow an easy and flexible extension or exchange of hard and software.

This paper is structured as follows. Chapter 2 gives a brief overview of the software framework the robot should be integrated in. Chapter 3 explains the mechanical structure of the new robot. In chapter 4 the electronic devices as well as their connection to each other is described. The software of the robot is presented in chapter 5, chapter 6 shows its capabilities in a tele-experiment.

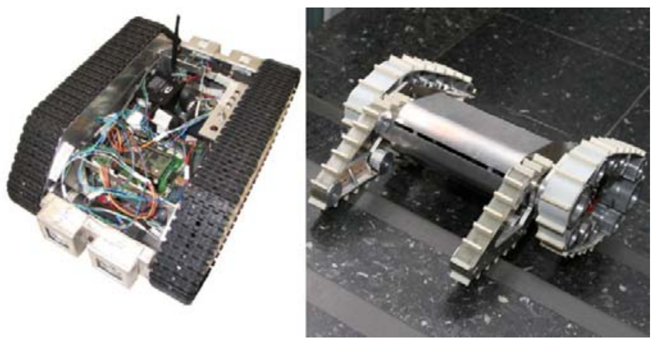

Fig. 1. Mobile robots: Tracked MERLIN II (left) and III (right).

2. ARCHITECTURE OF THE ROBOT NETWORK

In the current architecture three levels are proposed as depicted in figure 2: agents, coordinators and a server. The agents are mobile robots which are of general purpose or even for a very specific task. It doesn't matter if they drive, walk, fly or swim as long as the mobile robots use a digital protocol for transmission of commands and data. Their communication interface to the coordinator is also not relevant as long as a hardware driver for the physical interface exits on the coordinator side.

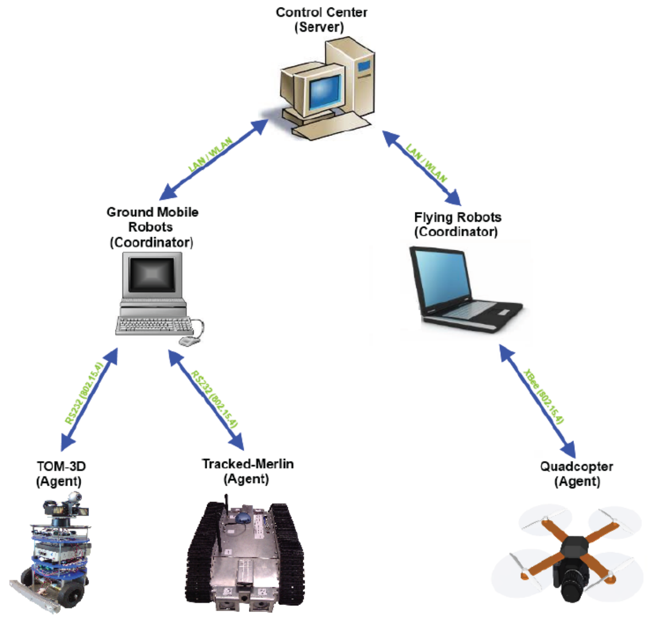

Fig. 2. Robot network.

During a mission an operator might want to move an agent from position a to b. It's clear that moving commands for flying and ground robots can be very different, depending on their design. For the operator it will be still the same instruction. The coordinator acts as a translator for this commands between server, operator(s) and the different mobile robots. All data from the different robots is stored into the server's database, where it can be further processed for map building or remote control of the agents.

3. MECHANICAL SYSTEM OF THE ROBOT

The mobile robot to be evolved is Tracked MERLIN III. Figure 1 and 3 show the mechanical structure of the new robot. It has two main tracks for its movement. In each track the return rollers and the road wheels are connected to the body of the robot with a spring damper system to provide smoother movement in outdoor environments. To improve the outdoor capability it has two flipper arms with tracks, which enable the robot to climb over outdoor obstacles like stones, wooden logs, etc. The arms can be rotated around their fixing points, which are the front wheels of the main tracks. During normal operation the tracks are inside the body of the mobile robot. In case they're needed they can be moved out to push up the robot and to extend its track length. Each of the two main tracks and two flipper tracks is driven independently by a high performance dc motor. The orientation of each flipper arm can also be changed by two high performance dc motors with a high gear ratio to produce enough torque to push up a 20 kg mobile robot. This two dc motors have additional brakes on their shafts which are always closed in their passive status and have to opened by applying a voltage to its electro magnets. Once the flipper arms are in the desired orientation, the brakes can close again and the orientation control can stop working. The whole chassis as well as the arms are out of aluminium to ensure robustness in outdoor applications. It can be closed and sealed against splash water. Dimensions of the mobile robot are 47 cm for its length, 44 cm for it width and around 20 cm for its height. It can archive a speed of 20 cm/sec during operation. This is limited by the motors under consideration of power aspects.

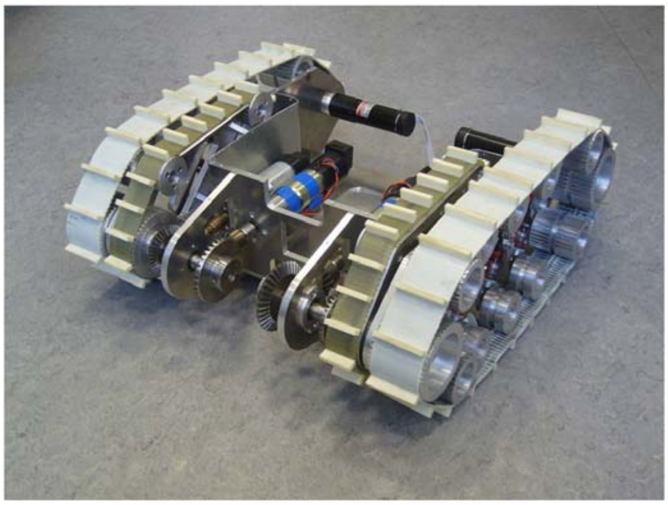

Fig. 3. Mechanical structure of Tracked MERLIN III.

4. ELECTRICAL SYSTEM

Typical electrical systems in a mobile robot are controller, sensors, batteries and data transmission devices. Traditionally the controller handles all control, sensor processing, trajectory control and communication tasks. If one intends touse a digital camera sensor for remote controlling or even image processing a simple microcontroller is not sufficient for this task anymore. In the Tracked MERLIN II design camera and microphone data are transmitted via an additional wireless transmission hardware to the operator. Apart from the additional costs and the technical limitations of the system it needs a lot of extra space in the robot, which is very rare. The biggest challenge for such a mobile robot is the control of the motors. Most microcontrollers don't offer an interface for incremental encoders, some do have one interface and only a few do have two. In contrary MERLIN III has six dc motors, which have to be controlled using a closed control loop.

For efficient modularisation following electronic groups are identified: main microcontroller, microcontroller for closed loop speed control, sensors, batteries, embedded pc, camera, network interface.

4.1 Master Controller

The main tasks like sensor reading, sensor data processing, trajectory control are done by an embedded module using a dsPIC33EP512MU810 microcontroller. Due to an internal Phase-Locked Loop (PLL) the user can choose the speed of the microcontroller by software. Here it operates with 40 MHz. It offers a great range of supported interfaces like 2 encoder interfaces, 4 Universal Asynchronous Receiver Transmitter (UART), 4 Serial Peripheral Interfaces (SPI) and 2 Inter-Integrated Circuit interfaces (I2C), 9 Timers, Direct Memory Access (DMA) for most internal interfaces, up to 32 analog inputs, 2 Controller Area Network (CAN) interfaces and much more. A double H-bridge (L298) on the board allows to control two dc motors using the PWM of the controller. In this configuration it is used for the orientation control of the flipper arms. Since the gear ratio for the flipper arms is high, a maximum current of 2 A is enough for each motor.

The key to use all the above mentioned interfaces with this 100 pin microcontroller is in its multiplexing unit where internal devices can be mapped to remappable output pins. This makes it possible to design a circuit board having all connections and to choose which, interfaces are connected to the pins of the microcontroller. More details can be found in Microchip (2013). Therefore this hardware is very flexible and can be used for different robot and sensor configurations.

4.2 Slave Controllers

Because of the already mentioned number of dc motor to be controlled, the idea was to design a small embedded system which controls the speed or the position of a dc motor with an attached incremental interface. Therefore a circuit board using a dsPICFJ128MC804 was designed. Since the controllers have two incremental interfaces its software is designed for two closed control loops. The board itself is designed as a shield board for the commercial power electronic board RN-VN2 DualMotor, supplying two dc motors with 5 A each. Both boards combined are shown in the following figure 4.

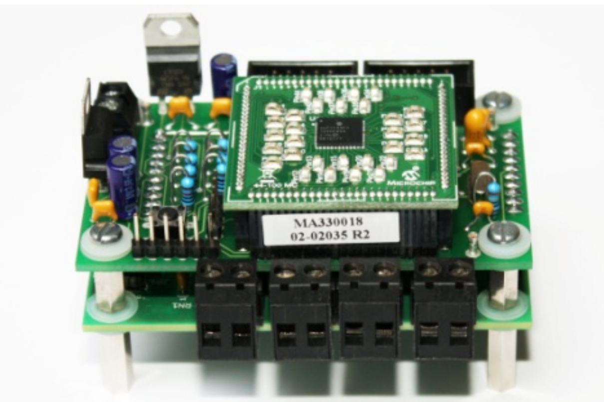

Fig. 4. Controller board for closed loop speed control with bridge driver board below.

The software of the PIC contains two closed loop speed controls with a cycle time of 1 ms. Both get their parameters, instructions and reference values via I2C bus from the master board. This board can be used on any mobile robot or application which needs a speed control and offers a standard incremental interface and uses 6-16 V input voltage for the motor and which doesn't consume more than 5 A. This board is used twice in the Tracked MERLIN III robot: for the speed control of the main tracks and the tracks of the flipper arms.

The firmware of this controller is programmed in a way that everything within the control loop can be configured by the master (cycle time, controller gains, dimensions of the wheels, gear ratios, etc.). It is connected to the I2C-1 interface of the main controller with a bitrate of 400 Kbit/sec.

4.3 Sensors

Tracked MERLIN III contains a huge variety of sensors. It uses 5 SFR02 ultra sonic sensors, which are connected via I2C-2 with 100 Kbit/sec to the main board. This keeps the computational burden of the creation and the evaulation of the ultra sonic sounds away from the microcontroller. Their measureing range is 15 cm to 6 m.

Furthermore it uses 6 GP2D12 infrared distance sensors for ranges between 10 - 80 cm. Although the ultrasonic sensors measure in the same range, the update rate of the infrared sensors is much better, which is important for real time obstacle detection while operating. They are connected to the analog pins of the microcontroller. Four of them are mounted in the front of the robot, a pair of each in different heights. This makes it easy to descide if an obstacle it small enough to be climed over or not. Figure 5 depicts this situation with a stair. If IR1 & 2 detect an obstacle and IR 3 & 4 detect nothing or an obstacle of farer distance, it's interpreded as a stair by the robot and the obstacle can be climed over instead of being avoided.

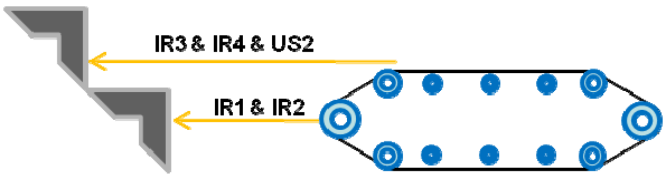

Fig. 5. Front sensor configuration of the mobile robot.

To know the home orientation of the flippper two CNY70 optical sensors are used. For safety reasons two ACS712 current sensors, which measure the current of the motors which move the flipper arms. With this configuration it is possible to detect an overload of the flipper arm motors. The optical sensors are connected to an I/O pin of the controller and generate an interrupt when they're 'high'. The outputs of the current sensor are also connected to analog pins of the controller.

An UM6 Inertial Measurement Unit (IMU) uses a tri-axis gyroscope, a tri-axis accelerometer and a tri-axis magnetometer and fuses internally by an extended Kalmanfilter, resulting in the attitude and position of the robot. If offers an UART and SPI as interfaces. Here the UART1 port is being used with a baudrate of 57600 baud/sec.

Finally a standard GPS completes the robots outdoor navigation capabilities. This GPS can eighter be connected to the IMU or to the microcontroller. Here it's connected to the IMU with a baudrate of 9600 baud/sec.

4.4 Batteries

Three packages of nickel-metal hydride batteries of 12 V each are connected in parallel to increase the capacity of the battery system to 10.5 Ah.

4.5 Embedded PC

Due to the rise of smart phones and tablets, especially with the phone operating system Android, inexpensive embedded PCs using an ARM-based CPU, running with Linux are available on the market. They are powerful in terms or calculation and need very little energy. The usage of such embedded systems offer new possibilities for the integration of new devices in a mobile robot. In the past telecontrol was mostly done via wireless serial interfaces, which have limited range and very low bandwith. Using an embedded PC offers the possibilty to use standard PC equipment to connect the mobile robot via USB to WLAN, UMTS or even LTE networks. This offers totally different perspectives to the mobile robot since now the bandwith is much higher and the operating distance significantly increases. The choice for the embedded system was made for a Raspberry Pi board. It offers a 700 MHz ARM CPU, USB connectors, a camera interface, serial port und much more. Important for the choice was its small size of ~9 times 6 cm, since space is rare inside the body of Tracked MERLIN III. Its processor includes realtime video encoding/decoding and an Image Sensor Pipeline (ISP) for cameras (Broadcom (2013)). Because the board only offers 2 USB ports, providing only ~100 mA, a USB hub is used. Right now a WLAN stick is used to connect the Raspberry to a router, but it can be replaced easily with LTE sticks. Due to LFM (2011) a cell size of an LTE network depends on the used frequency. Using 800 MHz can result in a cell radius of ~10 Km, 2.6 GHz in ~5 Km. The frequency is related to the bandwith and therefore with the amount of data one can transmit. In all configurations it will be more than enough to transmit a videostream.

4.6 Camera

One other benefit of the Raspberry Pi is its Camera Serial Interface (CSI). It's possible to attach any kind of camera, using this interface. Here the Raspberry Pi Camera Board was chosen. It offers pictures up to 5 Mpixel and video streams of different resolutions and framerates. It's very compact and offers also a software API to acquire images or videos easily and with high performance because of hardware support in video encoding.

4.7 Network

Right now the robot is connected to the remote control device via WLAN through a router. The WLAN adapter is standard WLAN-USB stick and can easily be replaced with a UMTS or LTE stick as long as the Raspberry can be accessed by any kind of interface which offers the usage of the TCP/IP protocol.

5. SOFTWARE SYSTEM

To offer extended flexibility in the entire system, the software also has to be developed in a modular way.

5.1 Software of the main controller

The PIC controller on the main controller board is the brain of the mobile robot. On the one hand it receives commands from outside and transfers this into driving commands for the speed control boards and on the other hand it receives information from the robot's sensors and transfers it to an operator. The operator is usually a human, controlling the mobile robot from another computer. The data transfer between mobile robot and operator's computer is defined as described in table 1. This protocol can be used through any other protocol and interface. It's designed to connect all agents and servers with each other.

Table 1. Data protocol.

No.	Packet Field	Size [Bytes]	Description
1	Header	2	Initiates the protocol and defines if it is a report (from μC to PC) or a command (from PC to μC).
2	Dest.ID	1	Defines the destination id, since there could be different operators in the network.
3	Payloadsize	1	Number of bytes contained in the message from no. 4-6.
4	SourceID	1	Identifies the sender.
5	Packettype	1	Command or Packettype.
6	Payload	0...n	The data to be transmitted.
7	Checksum	1	Modulo 256 checksum.

Its software is developed in modules, which can be switched on and off, just depending on the needs of the hardware configuration. It has to communicate with a lot of different interfaces as depicted in figure 6. Therefore the modules are also developed in a way that it's easy to switch the used physical resource, e.g. to switch from UART1 to UART2. The communication protocols between controller and sensors are defined by the sensors, which are usually also designed for easy and efficient handling from the producers. The protocol between the master controller and the speed controllers is designed like a standard I2C protocol, so it's also easy and efficient to handle.

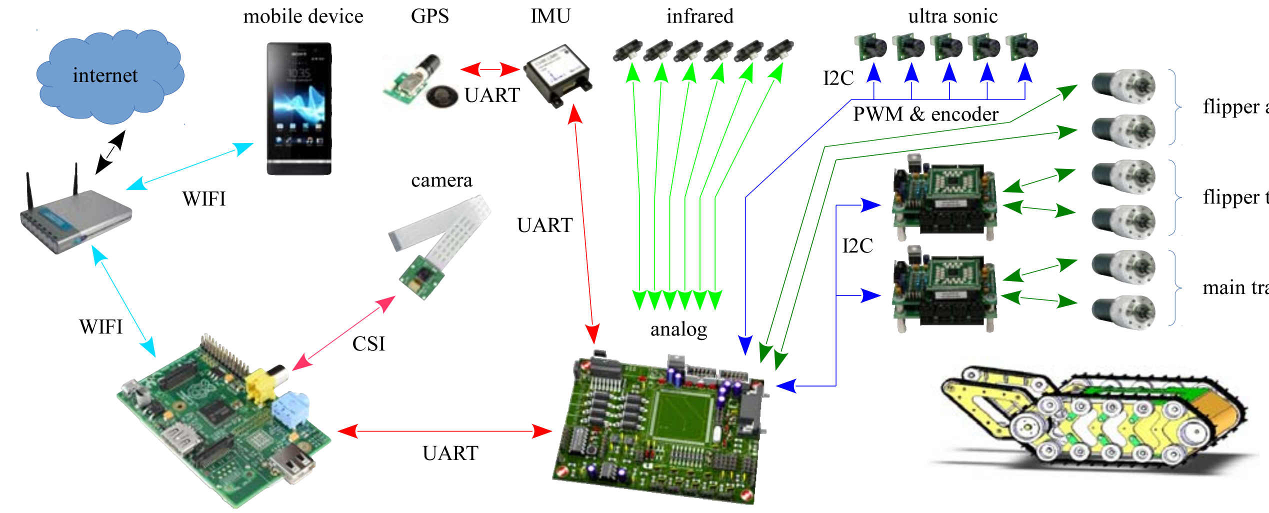

Fig. 6. Hardware scheme of Tracked MERLIN III

The software of the microcontroller makes heavy use of the Direct Memory Access (DMA) operations, which transfers data from special registers to the memory (and vice versa) without interrupting the CPU, to be able to handle all the different interfaces without stopping the main tasks of the software.

Since the main controller board is supposed for remote operation, its software implementation includes a very easy but sufficient handling for delays or communication losses. Every movement command of the mobile robot is stopped automatically after a certain time or distance. In case of a communication loss or a big delay, the robot simply slows down till zero velocity. State of the art methods to compensate the delay in transmission are not implemented, yet. Finally the controller software also offers some assistance driving functions to the operator, like automatic obstacle avoidance.

The controller board it connected to the Raspberry Pi board via UART interface using a baud rate of 115200 baud/sec.

5.2 Raspberry Pi

The above described protocol is received by the serial port of the Raspberry. It simply extracts the messages from the protocol out of the serial buffer. Meanwhile it listens on a port waiting if another application connects to it and sends or requests data. The application works in both directions. Data is send and received through the serial port and a network coordinator like a pc or mobile device can access the Raspberry's data via TCP/IP and the port. This software module works independent from the hardware configuration of the robot since the protocol (table 1) will always remain the same, no matter what instructions are being exchanged between microcontroller and coordinator.

Apart from providing access to the wireless networks, the Raspberry board also has the task to capture a real time video stream for the operator. This is done using the Raspberry Pi Camera Board, which also provides an API to capture and encode images and video streams using the GPU inside the ARM CPU of the Raspberry Pi. The resolution and the quality of the video stream can be controlled by software. As good as this hardware support is, this is also a drawback if the operator wants to use a mobile device like an android phone or tablet. The decoding of the h264 video standard is not yet implemented in Android and a work around depicted in figure 7 had to be implemented. The video stream is acquired from the camera using the Raspivid API, then it is wrapped with a mp4 container and streamed using the Real-Time Streaming Protocol (RTSP) using the ffmpeg and ffserver libraries (FFMPEG (2013)). Thus it's possible to display the video from the robot in an self written Android app.

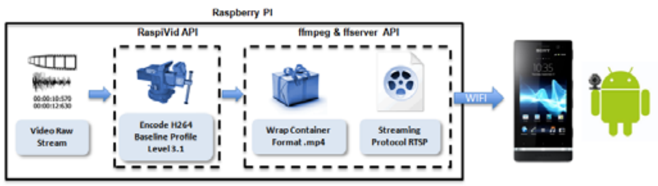

Fig. 7. Scheme of the container wrapping for the video stream.

6. TELEMATIC CONTROL OF THE ROBOT

An Android app was developed which enables the operator to control most things of the mobile robot Tracked MERLIN III. For the sake of simplicity this app communicates directly with the Raspberry Pi board and not through the complete robot network. The Android app can display the video stream of the robot like it is shown in figure 8 and has a virtual joystick to control the robots movement in the desired speed and direction. The resolution of the video stream is 640 times 360 pixel, its frame rate 25 fps with a bit rate of 500 Kbit/sec. It visualizes the robot's sensor data like all distance measurements or the robot's orientation. Due to the nature of the video format, the encoding, the wrapping and the decoding of the video stream right now the delay is quite high with 7-10 sec. Delays in the movement commands for the robot are much below a second and can hardly be recognized. The average network traffic, including remote operation and video stream, was measured as 65 Kbytes/sec.

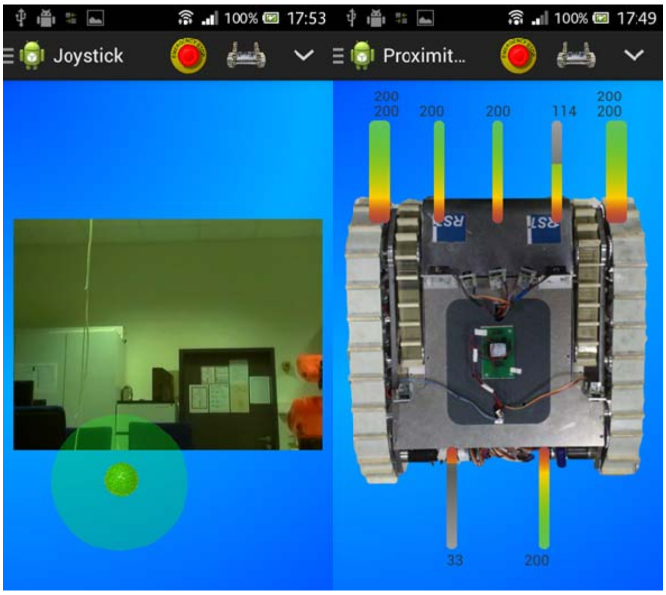

Fig. 8. Graphical User Interface of the mobile application.

A very promising application for this mobile robot would be a search and rescue scenario. One can imagine a destroyed environment, maybe a house. It's not know if there are still people inside the building but the entrance is partly destroyed and the building is already instable. It wouldn't be save for a human to enter the building and to look for people inside it. The robot is small enough to enter the house through smaller holes, can climb over some obstacles and can climb up and down stairs. With the camera the operator can check the situation inside the building. If the operator finds people to be rescued he can still decide about the risk of entering the instable building. In this application the remote controlled mobile robot would lower the risk for the humans significantly during a search and rescue operation.

7. CONCLUSIONS AND FUTURE WORKS

In this paper it has been shown the development of hard and software modules for a mobile robot, which allow the easy creation of different mobile robots, depending on the needs of the user just by a combination of the presented subsystems. All robots can also easily be integrated, coordinated and operated remotely in a robot network.

Further works will be the improvement of the delay in the video stream. Furthermore the Android app should be integrated into the framework and extended to enable the user to control all mobile robots in the robot network. It will be also important to find robust solutions for the behaviour of the mobile robot in case the connection suffers from packet losses or crashes completely. Another topics would address data security, since only the operator should have control over the mobile robot via our software framework.

REFERENCES

Broadcom (2013), *BCM2835* [online], http://www.broadcom.com/products/BCM2835 [accessed 17 November 2013]

FFMPEG (2013), *Webpage of the ffmpeg project*, [online] http://www.ffmpeg.org/, [accessed 17 November 2013]

Jamshidi, M., Eicker, P. J. (1993), *Robotics and Remote Systems for Hazardous Environments*, PRT Prentice Hall

LFM (2011), *LTE: Was ist das Eigentlich?*, [online] Landesanstalt für Medien Nordrhein-Westfalen, http://www.lfm-nrw.de/fileadmin/lfm-nrw/nrw_digital/Digitalkompakt_LTE.pdf [accessed 17 November 2013]

Li, P, Ma, S., Li, B. Wang, Y., Ye, C. (2007), *An in-pipe inspection robot based on adaptive mobile mechanism: mechanical design and basic experiments*, In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, San Diego, USA, pp. 2576-2581

Microchip (2013), *dsPIC33E Family Reference Manual, Section 10. I/O Ports* [online] http://ww1.microchip.com/downloads/en/DeviceDoc/70000598c.pdf, [accessed 17 November 2013]

Nasir, A.K, Hsino, A., Hartmann, K., Roth, H (2012). *Heterogeneous Capability Multi-Robots Cooperative Framework*. In: Proceedings of the 1st IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control, Würzburg, Germany, pp. 157-162

Ruangpayoongsak, N., Roth, H.; Chudoba, J. (2005), *Mobile robots for search and rescue*, In: 2005 IEEE International Workshop on Safety, Security, & Rescue Robots (SSRR), Kobe, Japan, pp 212-217

Sauer, M. (2011), *From GSM to LTE : an introduction to mobile networks and mobile broadband*, Wiley, UK

Schilling, K., Roth, H. (2001), In: Proceedings of the 1st IFAC conference on Telematics Application in Automation and Robotics, Pergamon Press, Weingarten

Sheridan T. B. (1992), Telerobotics, Automation and Human Supervisory Control, The MIT Press.

Zhonghong, O., Pang, B., Deng, Y, Nurminen J. K., Ylä-Jääski, A., Hui, P. (2012), *Energy- and Cost-Efficiency Analsysis of AMR-Based clusters*, In: Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and grid Computing, Ottawa, Canada, pp115-123