

# Mobile Prototyping Platforms for Remote Engineering Applications

[doi:10.3991/ijoe.v5s1.997](https://doi.org/10.3991/ijoe.v5s1.997)

K. Henke, St. Ostendorff, Th. Volkert and A. Mitschele-Thiel  
Ilmenau University of Technology, Ilmenau, Germany

**Abstract**—This paper describes a low-cost mobile communication platform as a universal rapid-prototyping system, which is based on the Quadrocopter concept. At the Integrated Hardware and Software Systems Group at the Ilmenau University of Technology these mobile platforms are used to motivate bachelor and master students to study Computer Engineering sciences. This could be done by increasing their interest in technical issues, using this platform as integral part of a new ad-hoc lab to demonstrate different aspects in the area of Mobile Communication as well as universal rapid prototyping nodes to investigate different mechanisms for self-organized mobile communication systems within the International Graduate School on Mobile Communications. Beside the three fields of application, the paper describes the current architecture concept of the mobile prototyping platform as well as the chosen control mechanism and the assigned sensor systems to fulfill all the required tasks.

**Index Terms**—Remote Engineering, Remote handling, Remote sensing, Mobile robots, Mobile communication, Rapid prototyping

## I. INTRODUCTION

The Integrated Hardware and Software Systems Group, led by Prof. A. Mitschele-Thiel, at the Ilmenau University of Technology is attempting to use flying “Quadrocopter” robots<sup>1</sup> in meanings of deploying self-assembling ad-hoc wireless networks. They can be used in case of disasters, when any wired communication infrastructure is destroyed. For that, the robots create a network of mobile nodes that provide both mobile-phone and standard wireless network access which allow people in distress to call for help. And it's all achieved using off-the-shelf parts.

The system is based on the Quadrocopter concept of autonomous flying helicopter-style models, which are often described as “unmanned aerial vehicles” (UAV) or as “Microcopter”. This concept was enhanced by different off-the-shelf parts, including VIA's Pico-ITX hardware, an FPGA and a GPS unit, which finally leads to universal mobile communication platforms providing both mobile phone and WiFi access. The platforms can provide these functions far more quickly than a technician on the ground might ever be able to do. Furthermore, they could spread themselves around an area and establish a radio network without human intervention.

Despite all promising advantages of Quadrocopters there are limitations though. The biggest issue is the

power supply because the batteries deliver power for only about 20 minutes. However, these 20 minutes are for flight time, giving the robots enough duration to position themselves and land on high ground or a building somewhere, and meanwhile provide network coverage for several hours. [1]

## II. ARCHITECTURE OF THE PROTOTYPING PLATFORM

To simulate and examine many challenging operational scenarios, partially focusing on cooperative coordination and control of multiple mobile platforms for different missions, several test beds will be designed in the near future. These test beds have to reflect the complexity expected in these application scenarios and consist of many (semi-) autonomous (possibly heterogeneous) mobile platforms. That's why the main design philosophy was to use simple platforms with commercial off-the-shelf add-ons, like Quadrocopters in our case. Therefore, many of them can be easily operated at the same time. This provides a good combination of flexibility, agility and mobility and further allows the usage of these test beds in a wide range of applications, as described in section III [2,3].

In the following we will briefly describe the architecture of these multipurpose mobile prototyping platforms.

### A. The Quadrocopter concept as basic idea

The basis for the mobile prototyping platform is the Quadrocopter concept. It uses a four-rotor aerial platform with two different rotor-rotations (see Figure 1).



Figure 1. The first prototype for the aerial platform

<sup>1</sup>A Quadrocopter is a four rotors, radio controlled or autonomous operating, electronically stabilized flying platform (for details see next section).

How can a Quadcopter fly at all? The front and back rotor turn clockwise, while the left and right rotors spin counter-clockwise. To hover, all rotors rotate at the same speed. When doing so, the forces between the clockwise rotors on one hand and the counter-clockwise rotors on the other hand, are balanced out. This enables the Quadcopter to hang “steady” in the air.

To be able to fly in one direction, the Quadcopter will be brought out of balance. The speed of the rotor that opposes the desired direction is increased. This makes the Quadcopter tip over in a certain direction. Example: to fly forward, the back-rotor has to turn faster. This is called “pitch” or “nick”, respectively. The later name is the preferred name for the forward/backward-movement. Left and right movements are usually called “rolls”.

Turning around its vertical axe is called “yaw”, but in German language it is called “gieren”. To be able to yaw you need a force to turn the platform around. This is produced by changing the speeds between the forward/backward rotors and the left/right rotors. Example: to be able to yaw clockwise, the forward/backward rotors will turn faster and the left/right rotors will slow down a little bit. This makes the Quadcopter turning clockwise, while the same height is maintained [4].

However, to fulfill all the required application scenarios, as described in section III, this simple platform has to be enhanced by different hardware and software components which will be described in the following.

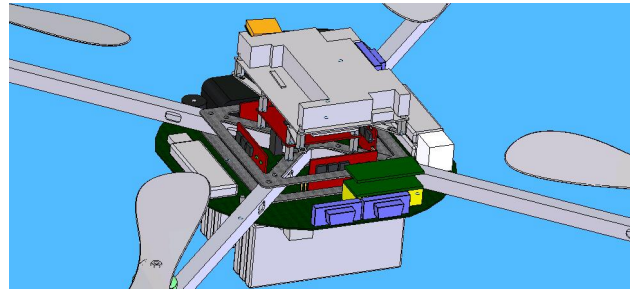


Figure 2. Arrangement of all additional add-on boards

### B. Hardware components of the mobile platform

The current hardware concept (shown in Figure 3) is based on a separation of the whole functionality into four different hardware components:

1. the basic flight control with all basic sensors,
2. a software add-on: a Linux based service platform,
3. a hardware add-on: a FPGA board for real-time autonomous navigation tasks and
4. the central board for signal routing between all platform components.

Furthermore, it is possible to connect additional peripheral components, e.g. monitor, display as well as a keyboard and a mouse. Figure 2 gives an impression about the actual arrangement of these components.

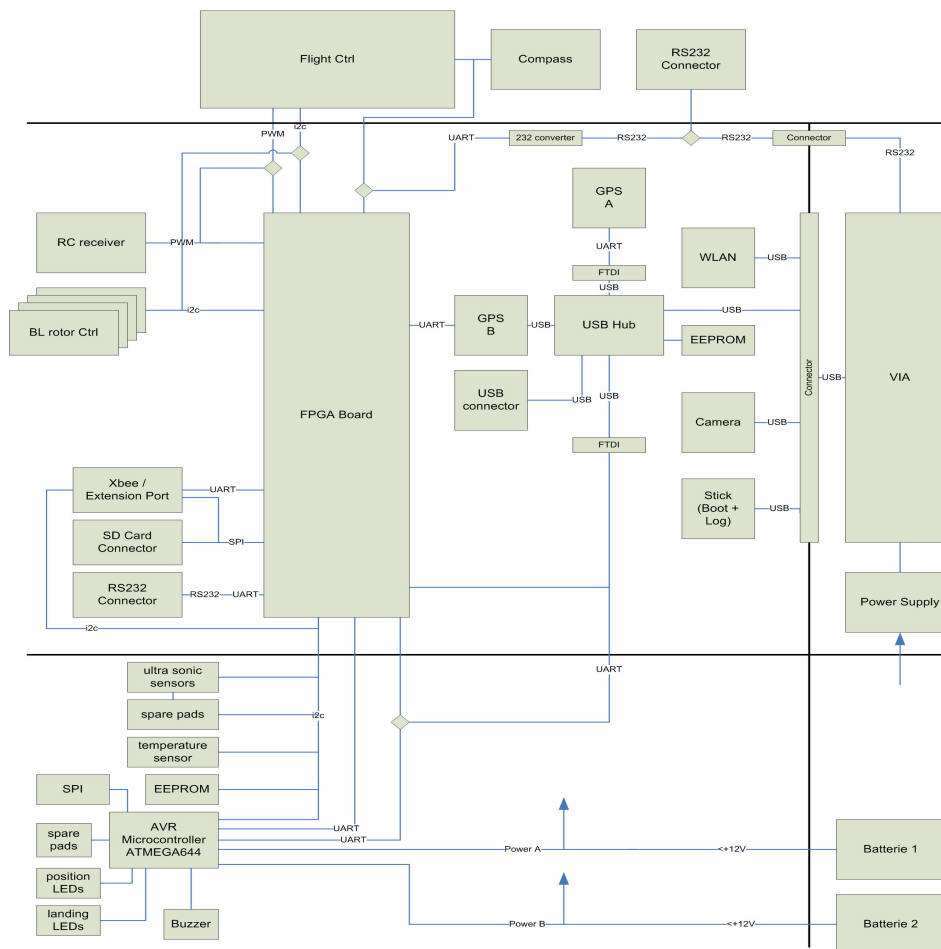


Figure 3. The hardware concept for the mobile rapid prototyping platform

### 1) Basic flight control with all basic sensors

For a stable flight a so called *flight-controller* is necessary. The main task of this controller is to make sure that the Quadcopter stays “steady” in the air. It uses data from several onboard sensors and calculates the speed for each individual rotor. Basically, the controller also compensates for external disturbances like wind [4]. All of these calculations are done by the onboard ATMEGA644 processor from ATMEL [5].

Of all sensors, the rotation speed sensors are the most important ones. The software uses them to determine the position in the air and to compensate external influences. For every axis (x, y and z) one rotation speed sensor is needed, so three of them all together. Usually these rotation speed sensors are called *gyroscopes* or *gyros* in short. They measure in degrees per second.

Another applied sensor type is the *acceleration sensor*. It measures the translation acceleration in all three possible directions. The vertical acceleration sensor is also able to measure the angle of the Quadcopter towards the earth. You can fly without acceleration sensors, but when they are attached to the controller, you are able to get the Quadcopter back to “position zero”. Hence, you can release the control stick and nevertheless the Quadcopter will stay at its position. Without the mentioned sensors, the Quadcopter will continue to fly in the prescribed angle. This kind of flying is called “heading hold”.

Furthermore, an *air pressure sensor* is placed on the flight-control, which enables the platform to keep flying on a static height all the time.

Additionally, the flight-control handles four *brushless controllers*. Each controller handles one brushless motor. These motors don't use any brushes for energy-distribution to the rotor. In contrary to brush-based motors, the magnets are rotating while the coils are steady. That's why DC current can't be used. Instead, an AC current source with precise pulses is needed to drive the motors. This source is provided by the applied brushless controllers, which are connected to the flight-control via an I<sup>2</sup>C bus system, routed by the central board. Each controller has an address, which is used by the ATMEL main processor, placed on the flight-control to realize a unicast communication.

### 2) The software add-on: A linux based service platform

Beside the flight controller, a separate x86-compatible C7 processor-based board from VIA [6], as depicted in Figure 4, is placed onboard the mobile platform and will be used for

- additional computation-intensive applications, e.g. the analysis of different communication protocols (for Ad-hoc net applications or self-organizing algorithms for disaster scenarios),
- Wired LAN connection during the development phase,
- Wireless LAN and/or XBee connections for the mobile phase [7],
- processing and streaming of the video data, captured from the onboard camera.



Figure 4. The software add-on: x86-compatible VIA board for all computational-intensive applications [6]

### 3) The hardware add-on: an FPGA board for real-time autonomous navigation tasks

The FPGA hardware is so far mainly used to

- **observe the flight-control input signals.** If radio transmission is lost, the system switches to a fail-safe mode and lands autonomously on the ground.
- **serve as black box.** All relevant flight data, like max. height, air speed and GPS position as well as flight-control inputs are logged and stored on the onboard SD card.

We use a Xilinx Spartan-3A evaluation kit - an easy to use low cost platform, especially for experimenting and prototyping applications based on the Xilinx Spartan-3A FPGA family (see Figure 5) [8].

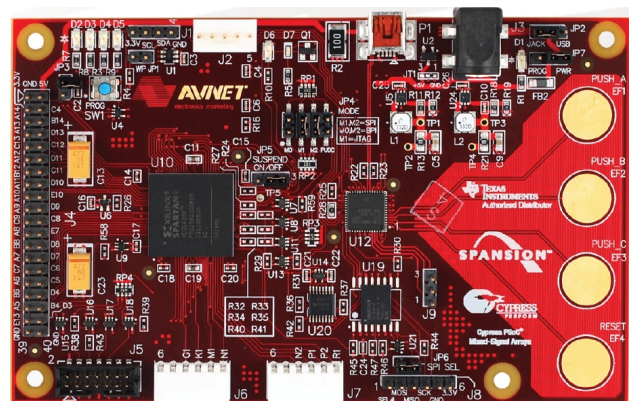


Figure 5. The hardware add-on: an FPGA board for real-time autonomous navigation tasks [8]

### 4) The central board for signal routing between all platform components

The four-layered central board is used to connect all the off-the-shelf add-on parts of the mobile platform, such as

- various onboard USB devices (GPS receiver, WLAN stick, Memory stick and camera),
- the X-Bee unit for ZigBee communication between the ground station and/or other mobile platforms,
- the ultrasonic sensors for distance measurement,
- the RS232 UART unit for serial communication,
- the temperature sensor and
- the SD memory card.

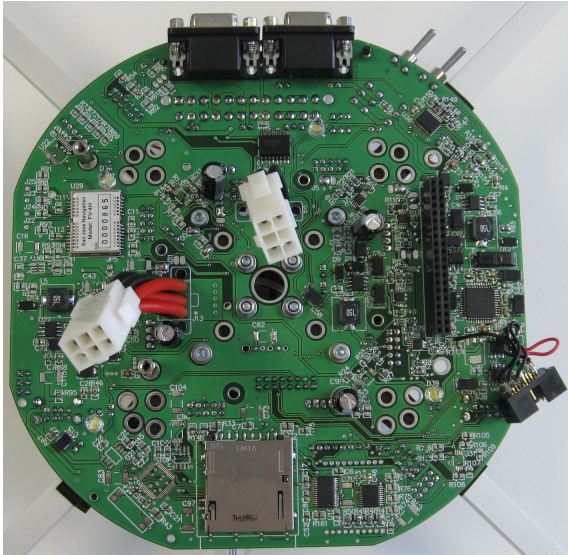


Figure 6. The central board for the signal routing

### C. Software modules of the mobile platform

Similar to the hardware concept, the software functionality is also separated into several software modules:

- (1) basic flight control functions (via flight control unit),
- (2) monitoring server for the flight-control unit (via x86-board),
- (3) microcontroller firmware for battery voltage monitoring and landing lights (via central board),
- (4) graphical monitoring via head-up display with overlay flight monitoring elements (data is sent via WLAN towards the ground station),
- (5) black box system for flight data recording (via FPGA & SD card),
- (6) camera streaming server (via x86-board),
- (7) sensor data multiplexer for x86-board and FPGA (via FPGA).

#### 1) Basic flight control functions

At the current state of the work we use all the built-in software functions of the flight-control unit [4]. In the future, we plan to substitute some of these software modules by other control mechanisms, e.g. fuzzy control.

#### 2) Monitoring server for flight-control functions

For the various application scenarios the monitoring of all onboard signals is necessary. For that we have developed a monitoring server, running on the VIA board [9]. The main tasks are:

- processing and storage of all flight-control signals,
- processing and storage of all sensor signals,
- processing and storage of the GPS data,
- observation of the radio transmission data stream,
- transfer of all the data via TCP.

The monitoring server can work autonomously or it can be controlled remotely via Ethernet, WLAN or ZBee [9].

#### 3) Graphical monitoring via head-up display

For a remote monitoring of relevant flight data a GUI was developed [10]. The main aspects of it are:

- JAVA-based, so it is platform-independent and it is currently used under Windows and Linux,
- graphical display of all flight metrics,
- implementation of various views,
- expandability with new pop ups or views,
- communication with the onboard monitoring server via TCP.

The look-and-feel of the GUI was influenced by the design of head-up displays of modern aircrafts. Figure 7 gives a short impression of the GUI elements.

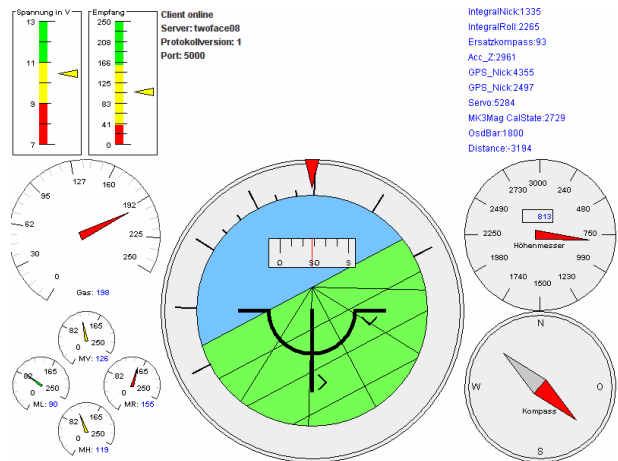


Figure 7. The head-up display for a remote monitoring

The next steps are the enhancement of the GUI functionality by combining the onboard captured camera stream with a simultaneous GPS tracing. The GPS trace is currently displayed in form of calibrated maps. These maps can be placed on the local system (offline), or they can be loaded dynamically from the network, e.g. “Open Street Maps” (online). In Figure 8 the used software concept is depicted. Furthermore, Figure 9 gives an impression of the described combination of camera and GPS data representation [11, 12].

#### 4) Black box system for flight data recording

As already mentioned, all relevant flight data, like height, airspeed, and GPS position as well as flight-control inputs are logged via the FPGA board and are stored on the onboard SD card.

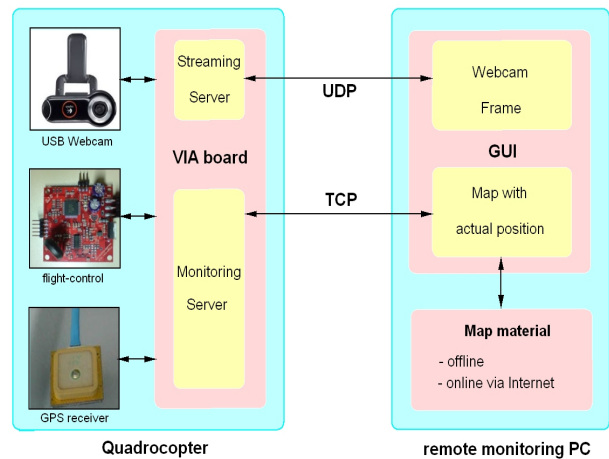


Figure 8. Combining camera and GPS data as new GUI function

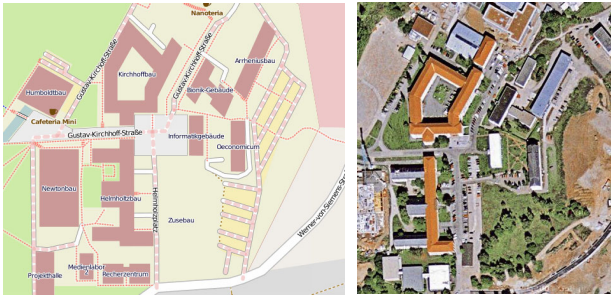


Figure 9. Combining camera and GPS data as new GUI function

### 5) Camera streaming server

As already mentioned the platform also includes an x86 compatible board with an onboard C7 processor. With the help of this configuration it is possible to develop software with a short deployment phase. This aspect was important for the first time when the camera streaming software was created. All development work was done by using usual PC components. Afterwards the software was easily copied to the x86 compatible board of the platform and it worked out of the box. This removed the time overhead when programming exotic embedded hardware.

The camera streaming server was completely written in C language. It supports several video streaming formats like H261 and H264. However, the resulting video resolution and quality are limited by the performance of the onboard C7 processor. It lacks current instruction sets like Intel's fourth iteration of their "Streaming SIMD Extension" (SSE) called SSSE3. Only support for the instruction sets MMX, SSE and SSE2 is provided by this main processor. Especially, live MPEG4 encoding as used in the H264 specification is a hard job for the processor. Furthermore, the used wireless network is another bottleneck for the video stream. Hence, only smaller resolutions like CIF or QCIF at a low frame rate of about 5 frames per second are recommendable to be used in this context. For the applied live encoding and decoding of the video frames an open source library called "ffmpeg" is applied [29]. This library is a cross-platform solution to record, convert and stream audio and video supporting several formats.

The implemented multi-threaded software architecture includes a video multiplexer which enables the system to distribute the video stream among several video sinks, like a responsible platform controlling person and some spectators. Until now, simple unicast communication is used to multiplex the traffic. In the near future it is imaginable to use multicast communication based concepts for traffic reduction on the wireless link. This could be implemented by a stream relay server at the ground station which distributes the video frames among all video stream subscribers.

### 6) Hardware-independent communication framework

For connecting various interfaces with different protocols a universal framework has been developed to

- exchange unicast and multicast packets for both control and data streams,
- simplify data exchange between a single master and multiple slaves with different interface specifications [13].

This framework is implemented within the FPGA (see Figure 10).

### 7) Microcontroller firmware for battery voltage monitoring and landing lights

For all other none-critical sensor data acquisition and control functions an additional AVR microcontroller is used, e.g. for

- battery voltage monitoring,
- handling of the landing lights,
- ultrasonic sensor processing,
- temperature monitoring as well as
- disturbance/warning buzzer.

The microcontroller uses an SPI and I<sup>2</sup>C interface to communicate with additional sensor modules on demand.

### D. Implemented Core Applications

Beside the planned various application scenarios, a set of core functions must be available to guarantee the whole functionality of the aspired tasks.

Some examples for such core functions for the operating environment are

- Auto pilot, including
  - autonomous navigation based on GPS and additional sensor inputs, like ultrasonic sensors (see Figure 11, [12, 14])
  - autonomous landing,
  - In case of radio transmission failure of the remote control system, the system switches to a fail-safe mode and lands safely by itself [16, 17]
  - autonomous homing (return to the Quadcopter ground station after completion of all tasks or for battery recharging)
  - way point navigation, fly autonomously along pre-defined way points (according to the users requirements)
- Intelligent navigation based on mission guidelines
  - optimal WLAN or X-Bee field intensity [7, 15],
  - optimal functional placement as mobile device for ad-hoc networks,
  - enhancement of WLAN coverage providing relay functionality
- General reconnaissance and coordination
  - distributed reconnaissance in coordination with base stations,
  - video observation, transmission and recording

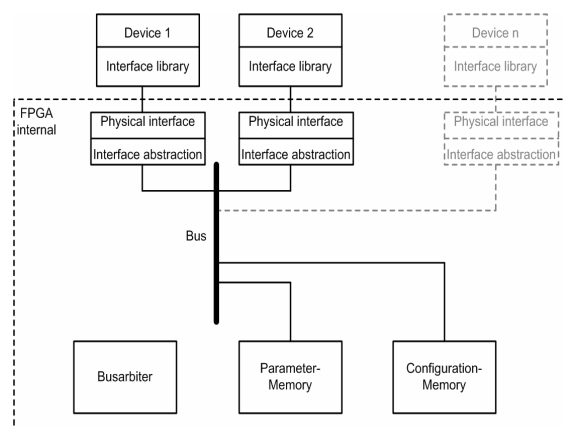


Figure 10. FPGA-based hardware-independent framework [13]

- Assistance of communication services to other mobiles and seamless integration into the remaining communication infrastructure as a whole in case of emergencies.
- Mission planning can be carried out online or offline. Beyond that, missions can be stopped and resumed at anytime.

### III. FIELDS OF APPLICATION

Worldwide there exist a lot of Quadcopter research applications, which are mainly done by universities and the military [18 - 24]. In the following we provide several application examples, showing how to use the mobile Rapid Prototyping platform

- to motivate bachelor and master students to study Computer Engineering sciences by increasing their interest in technical issues,
- as integral part of a new ad-hoc lab to demonstrate different aspects in the area of Mobile Communication as well
- as universal rapid prototyping nodes to investigate different mechanisms for self-organized mobile communication systems within the International Graduate School on Mobile Communications at the university.

#### A. Make “Computer Engineering” accessible

The main task of the newly created Joint European Master Degree Program Remote Engineering (MARE) – which is being implemented by partners from five European Universities – is to elaborate a complete new joint curriculum in the field of Remote Engineering [25]. At the Ilmenau University of Technology this program starts in April 2009 as specialization within the “Master of Science in Computer Engineering”. General objective of the curriculum is to mediate fundamentals, applications and experiences in the field of Remote Engineering by an interdisciplinary approach in combination with “learning by doing” phases. The Integrated Hardware and Software Systems Group is involved in this European Master Degree Program with lectures dealing with the design of complex digital control systems as well as lectures in the field of Mobile Technologies.

In different courses students learn various methods and tool concepts to create, implement and validate complex integrated hardware and software systems. For a better understanding some accompanying practical examples are beneficial. Unfortunately, such exemplary designs are too complex for a single lecture or 4-hour laboratory. That’s why we organize so called “project seminars” during a whole semester. The time for the project seminar is calculated with an amount of 4 ECTS. From our point of view, it is necessary to motivate bachelor and master students to study Computer Engineering sciences by increasing their interest in technical issues. In our contribution we would like to present a universal mobile prototyping platform for different application scenarios which is based on the Quadcopter concept. From the educational point of view, a Quadcopter is a mix of electrical and mechanical engineering as well as computer science and therefore ideally suitable to bring students closer to “Computer Engineering” [26, 27]. Current research tasks in our project seminars are suited to address students interested in one of

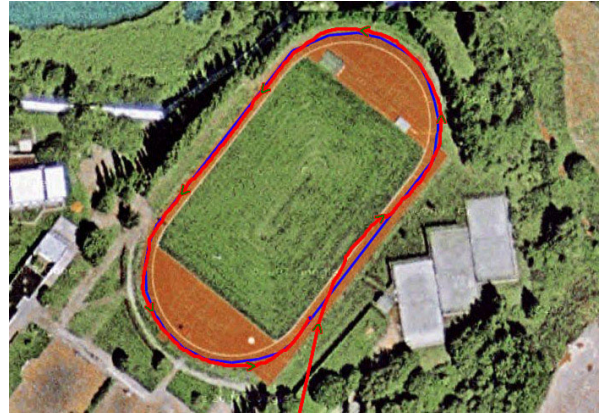


Figure 11. Resulting GPS track of the first autonomous GPS-based navigation experiments on a sports field

these topics and bring them together with other students with other fields of interest. Teamwork and interdisciplinary communication skills are important soft-skills that can only be developed and improved by practical experience and projects. Most of the work on the Quadcopter platform has been done by undergraduated and graduated students.

#### B. Ad-hoc lab scenarios

The second field of application is the newly-created Ad-hoc Lab of the Integrated Hardware and Software Systems Group. Before the Quadcopter platform existed, we had been only able to offer static lab experiments in the area of Mobile Communication, e.g. in our Wireless Lab. For a better demonstration of different aspects of high dynamic communication systems, a lab platform with real mobile devices was required. Only a lab structure with dynamic mobile devices can represent real scenarios and bring the students closer to the complexity of protocols and wireless network access methods. Additionally, the mobile platforms should have an auto-navigation ability to ensure the reproducibility of lab experiments.

The mobile communication nodes should be able to move within a given area randomly or should move along a given route. IP based cameras can observe all the movements. Moreover, a stationary lab server can act as access point for the communication as well as center for controlling and monitoring of the mobile communication platforms. The usage of multiple mobile platforms allows further the implementation of multi-hop (several intermediate mobile nodes are possible on the route to the uplink access point) as well as ad-hoc (communication without central access points which leads to a flat network structure) lab experiments.

The arrangement of the new ad-hoc Lab additionally allows students to acquire practical abilities in the area of Mobile Communication which substantially exceeds the normal knowledge transfer during lectures.

#### C. Self-organized mobile communication systems

A third important application area of the described prototyping platforms is the usage within the International Graduate School on Mobile Communications at the Ilmenau University of Technology. It focuses on self-organized mobile communication systems for disaster scenarios and addresses some of the most challenging research topics, the self-organization of distributed radio

systems and networks and the recovery of such communication systems after disasters.

Self-organization is applied both to the whole system and also to parts of the network, protocols and services. The main task is to develop a system consisting of self-organized networks, which is able to be deployed in disaster scenarios and which offers necessary services, like low-cost multi-target surveillance and tracking, widespread search and localization of individuals, moving target location and tracking or emergency communication. This includes land based and airborne vehicles to provide communication as well as localization and surveillance services.

Within the Graduate School, these mobile rapid prototyping platforms will be used for the following research topics:

- self-organized service management,
- self-organized recovery of services,
- self-organized management of remaining transport resources after disasters,
- self-organized management of radio resources and radio interfaces including distributed MIMO systems,
- self-organized management of rights and security issues,
- self-organized operation of land-based and air-borne vehicles to provide communication and other services which are no longer provided by regular infrastructure.

#### IV. CONCLUSION

With the described mobile communication platform – based on the Quadcopter concept – we offer a low-cost test platform for rapid prototyping of

- different control algorithms (based on microcontrollers, a PC or an FPGA),
- various communication protocols as well as
- different off-the-shelf hardware components

for challenging application scenarios, like ad-hoc labs or self-organized mobile communication systems.

With these multipurpose mobile prototyping platforms we provide the basics for the application of mobile airborne communication platforms. These mobile platforms are used within the International Graduate School on Mobile Communications of the Ilmenau University of Technology [28].

Since the communication platform is very robust, it is possible to execute multiple missions in a short period of time with minimal setup and organization between the experiments.

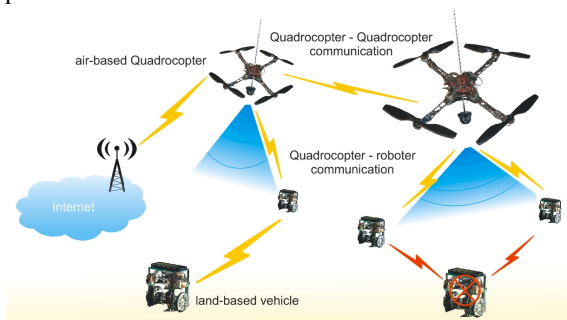


Figure 12. Airborne and land-based vehicles in mobile ad-hoc nets

#### ACKNOWLEDGMENT

The authors would like to thank many students which have supported this research, especially Sven Biegler, Thomas Dietrich, Kay Wenzel and Sandro Pax.

#### REFERENCES

- [1] D. Fearon, "Flying robots to provide Wi-Fi in disaster zones", *PC Pro*, <http://www.pcproweb.com>, 3 March 2009.
- [2] J.P. How, "UAV SWARM Health Management Project", MIT Aerospace Controls Laboratory, <http://vertol.mit.edu>, 2009.
- [3] J.P. How, "Multi-Vehicle Flight Experiments: Recent Results and Future Directions", AVT-146 Symposium on Platform Innovations and System Integration for Unmanned Air, Land and Sea Vehicles, Florence, Italy, May 2007.
- [4] Mikrokopter Wiki, <http://www.mikrokopter.com>, 2009.
- [5] ATMEL, <http://www.atmel.com>, 2009.
- [6] EPIA PX processor board, datasheet, [http://www.via.com.tw/en/products/embedded/ProductDetail.jsp?motherboard\\_id=472](http://www.via.com.tw/en/products/embedded/ProductDetail.jsp?motherboard_id=472), 2009.
- [7] Th. Dietrich, "A survey on alternative communication mechanisms for mobile communication platforms", student research paper, Ilmenau University of Technology, 2008.
- [8] Avnet Spartan-3A Evaluation Kit, [http://www.xilinx.com/products/devkits/aes\\_sp3a\\_eval400\\_avnet.htm](http://www.xilinx.com/products/devkits/aes_sp3a_eval400_avnet.htm), 2009.
- [9] S. Pax, "Configuration and monitoring of a mobile communication platform", student research paper, Ilmenau University of Technology, 2008.
- [10] S. Biegler, "Remote monitoring for a mobile communication platform", student research paper, Ilmenau University of Technology, 2008.
- [11] S. Biegler, "Graphical position monitoring for a mobile communication platform", student research paper, Ilmenau University of Technology, 2009.
- [12] K. Wenzel, "Possibilities and limitations of GPS auto-navigation", student research paper, Ilmenau University of Technology, 2008.
- [13] Th. Hertwig, "Hardware-independend communication framework", student research paper, Ilmenau University of Technology, 2009.
- [14] F. Herzog, D. Kling, "Sensor system extension of the mobile communication platform", student research paper, Ilmenau University of Technology, 2008.
- [15] S. Lange, N. Sünderrhauf, P. Protzel, "Autonomous Landing for a Multirotor UAV Using Vision", International Conference on Simulation, Modelling and Programming for Autonomous Robots SIMPAR2008, 3 – 4 November 2008, Venice, Italy.
- [16] G.P. Tourmier, M. Valenti, J.P. How, E. Feron, "Estimation and Control of a Quadrotor Vehicle Using Monocular Vision and Moiré Patterns", AIAA Guidance, Navigation and Control Conference, 21 – 24 August 2006, Keystone, Colorado.
- [17] M.C. Berntssen, H. Haehnel, "Evaluation of Cost-Sensitive Wireless Transmission Systems for Time Critical Mobile Systems", International Conference on Remote Engineering and Virtual Instrumentation, REV2008, Düsseldorf, Germany, 23 – 25 June 2008.
- [18] H. Surmann, D. Holz, S. Blumenthal, Th. Lindner, P. Molitor, V. Tretyakov, "Teleoperated Visual Inspection and Surveillance with Unmanned Ground and Aerial Vehicles", International Journal of Online Engineering i-JOE, Vol 4, No 4 (2008).
- [19] AWARE Project, "Platform for Autonomous self-deploying and operation of Wireless sensor-actuator networks cooperating with AeRial objEcts", <http://www.aware-project.net>, 2009.
- [20] Quadcopter project, TU Chemnitz, <http://www.tu-chemnitz.de/etit/proaut/forschung/Quadcopter.html>, 2009.
- [21] AMFIS project, Aufklärung mit mobilen und ortsfesten Sensoren im Verbund, <http://www.iitb.fraunhofer.de/servlet/is/18599>, Fraunhofer IITB, 2009.
- [22] A. Undurti, "UAVs helping a human supervisor map an area that has just experienced a Chemical/Biological/Radiological/Nuclear (CBRN) incident", MIT Aerospace Controls Laboratory, <http://acl.mit.edu/projects/cbrn.html>, 2009.

- [23] Project "Autonom Fliegende Quadrocopter", Max-Planck-Institut für biologische Kybernetik, <http://www.kyb.mpg.de/de/projects.html?prj=346&compl=0>, 2009.
- [24] L. Ji, "Increasing Accuracy of Location Determination", Master Thesis, KUNGLIGA TEKNISKA HÖGSKOLAN, Royal Institute of Technology, COS/CCS 2007-14, Stockholm, Sweden 2007.
- [25] M.E. Auer, I. Grout, K. Henke, R. Safaric and D. Ursutiu: "A Joint Master Program in Remote Engineering", International Journal of Online Engineering (iJOE) [Online], Vol. 2, No. 2, 30 Apr (2006)
- [26] N. Sünderhauf, Th. Krause, P. Protzel, "Bringing Robotics Closer to Students – A Threefold Approach", IEEE International Conference on Robotics and Automation ICRA06, Orlando, Florida, 15 – 19 May 2006.
- [27] N. Sünderhauf, Th. Krause, P. Protzel, "RoboKing – Bringing Robotics closer to Pupils", IEEE International Conference on Robotics and Automation ICRA05, 18 – 22 April 2005, Barcelona, Spain.
- [28] Mobicom - International Graduate School on MOBILE COMMUNICATIONS, Ilmenau University of Technology, <http://www.gs-mobicom.de>, 2009.
- [29] FFMPEG project, <http://www.ffmpeg.org>, 2009

## AUTHORS

**K. Henke** is with the Ilmenau University of Technology, Faculty of Informatics and Automation, Integrated Hardware and Software Systems Group, 98684 Ilmenau,

Germany, POB 10 05 65 (e-mail: karsten.henke@tu-ilmenau.de).

**St. Ostendorff** is with the Ilmenau University of Technology, Faculty of Informatics and Automation, Integrated Hardware and Software Systems Group, 98684 Ilmenau, Germany, POB 10 05 65 (e-mail: steffen.ostendorff@tu-ilmenau.de).

**Th. Volkert** is with the Ilmenau University of Technology, Faculty of Informatics and Automation, Integrated Hardware and Software Systems Group, 98684 Ilmenau, Germany, POB 10 05 65 (e-mail: thomas.volkert@tu-ilmenau.de).

**A. Mitschele-Thiel** is with the Ilmenau University of Technology, Faculty of Informatics and Automation, Head of the Integrated Hardware and Software Systems Group, 98684 Ilmenau, Germany, POB 10 05 65 (e-mail: andreas.mitschele-thiel@tu-ilmenau.de).

This work is partly supported by the German Research Council in the context of the Research Training Group program (Graduiertenkolleg) under contract GRK 1487 "Self-organized Mobile Communication Systems for Disaster Scenarios".

This article was modified from a presentation at the REV2009 conference in Bridgeport, CT, USA June 2009. Submitted, March 2, 2009. Published as resubmitted by the author(s) on July, 20, 2009.