International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

Keyboard Tele-operation for Autonomous Amphibious vehicle

Khaled Sailan, Siegen University; Marc Steven Krämer, Siegen Uninversity

Abstract

This paper presents the design of a controller intended for Tele-operation. It is capable of controlling an amphibious autonomous vehicle through a WLAN. The system uses the already widespread Wi-Fi technology as its wireless communications medium. The user can control the amphibious autonomous vehicle remotely and access its sensory feedback signals as well. This project covered the design of control circuitry for the amphibious autonomous vehicle. It also covered the design of the software to communicate signal from a pc controlling unit to the microcontroller unit through Raspberry Pi unit which has WIFI adapter integrated in the unit. This software is Robot operating system (ROS) which used in the Linux environment to set the communication between the Raspberry Pi and the PC.

Introduction

Popularity of the research on amphibious autonomous vehicle has been recently increasing, due to their possible use in different outdoor environments. Planetary explorations, search and rescue missions in hazardous areas [1], surveillance, humanitarian demining [2], The need for large -scale robotic vehicles arises in the exploration of hazardous sites as well as in the exploring on the land and on the water. Both applications require a vehicle that can operate continuously for several hours, cover large distances at speeds of up to 25 meter per second, carry a payload of several hundred kilograms, and drive at precisely controlled speeds. To meet these needs, institute of real time learning system in Siegen University has procured an all-terrain vehicles DORIS Robot figure 1 with the intention of converting them to driverless operation for use as amphibious autonomous vehicle. In addition to meeting the basic criteria stated above, DORIS Robot was selected for its payload flexibility, its stability on a variety of terrain, its ability to perform point turns, and the estimated ease of automating all onboard functions. The purpose of this project was to develop the keyboard Tele-operation speed and steering control that will enable the vehicle to driverless operation, to function as a robotic vehicle [3]. Many high-level tasks that are difficult to achieve autonomously can already be tackled when the intelligence of the user is combined with robot skills by means of tele-operation Nevertheless, users want personal robots as convenient tools that act as autonomously as possible, since steady low-level control of the robot would be tedious and time-consuming. Only in difficult situations, in which no autonomous solution exists yet, the user should need to take over control of the robot on the most convenient, i.e. most autonomous, level. We propose handheld user interfaces that allow persons to tele-operate an amphibious robot on various levels of autonomy[6].

the Tele-operation technique used the robot operating system (ROS) to send the command from the PC controlling unit to the microcontroller unit through the Raspberry Pi unit ROS software mounted on both devices and tow nodes created to send and receive the keyboard events.

The result of this research shows the control architecture for this vehicle and implements it on real-time computing hardware on the converted vehicle.

Figure 1 Doris Robot

ROS

Robotic middleware solutions migrated to a thin-design paradigm that supports the development of modular components and increases the ability to reuse existing code. Several frameworks follow this new paradigm with the Robot Operating System (ROS) being one of the most popular. Since its release in 2007, several hundred packages have been published. Despite the clear benefits that ROS introduced to the robotic community, the concept of ROS is not particularly new, but it is actually comparable to existing solutions such as, e.g., the Inter Process Communication

ISSN:2319-7900

(IPC) library. ROS provides mainly a communication interface between independent pieces of a framework together with a platform to share released code. In order to implement their algorithms on a particular robotic platform, robot cists are still forced to develop the appropriate drivers and controllers and to link them into a reliable framework. While drivers for several hardware components have been gratefully shared by other scientists through ROS, the development of an appropriate control framework still remains a challenging and time-consuming task, for example in the popular research field on *Unmanned Ground Vehicles* (UGVs) [4]. Robot Operating System (ROS) is used as the middleware for the robotic platform. Safety features, at both the software and hardware level, are pre-built into the robot thus providing the basic measures for ensuring its safe operation in human environments. The ROS middleware also provides easy access to the open-source library of various robotics algorithms, which can be used off-the-shelf or customized for the mobile base.

Hardware components

The hardware used to drive the amphibious vehicle autonomous shows in figure 3 which consist of:

Client PC :The client PC is a standard personal computer running Linux and robot operating system (ROS) with networking enabled, and the necessary drivers installed.

Server Pc :The Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools.

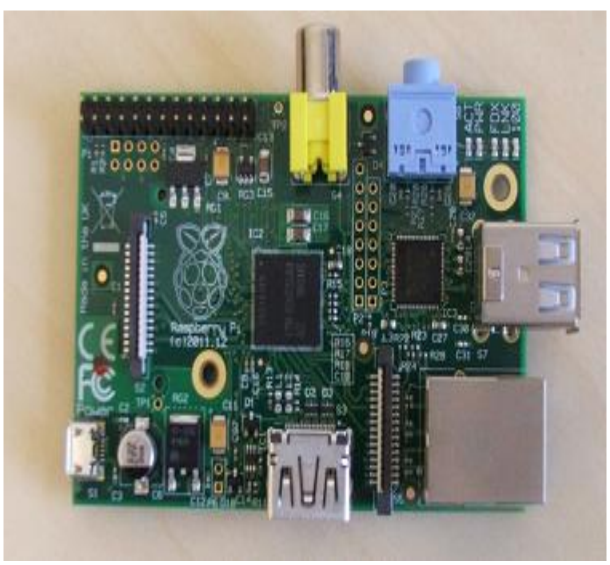

Figure 2 raspberry Pi platform

Software

A Tele operation system was chosen because it allows much more flexibility in programming; allows for easy warless control; provides a reduced need for specialist hardware; and wireless communications have already been well established [5]. The programs have been written in c. The same program runs on both the server and client computers; however one is set as the server and the other as client at runtime. The server listens to its set ports waiting for a connection from the client. The client needs the IP address of the server and the port number it is listening to, and then can establish a link. The wireless communications are handled by ADHOC through WIFI. Once communications have been established, the client and server computers can send each other the necessary control commands. The control command send from the pc to the client pc which is raspberry pi through WIFI and from the raspberry pi to the microcontroller through the serial communication as shown in figure 3.

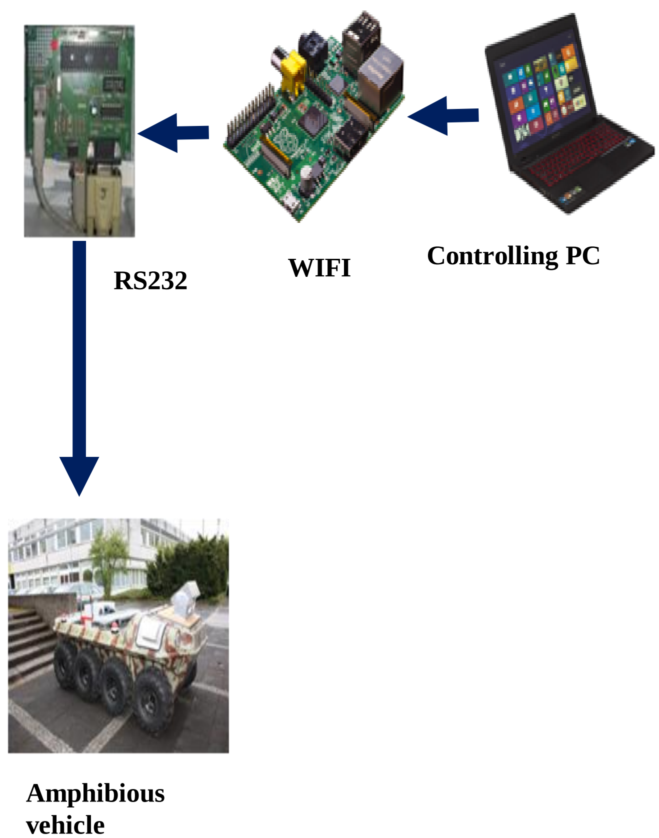

Figure 3 hardware structure of keyboard tele-operation

ISSN:2319-7900

Tele-Operation of the robot including 2 parts, one is a listener (listener.cpp) and another one is publisher (keyboard.cpp). The setting is like this, we have two nodes running; one is a publisher which takes keyboard input and kept sending a keyboard events command out. A second node, a listener, kept listen to keyboard events command then print it out on our computer screen and also send it to the microcontroller via RS-232 cable. In the microcontroller a UART communication sitting is made to get the keyboard events. The Dc Motors positions that produce hydraulic pump speed and wheel speed divided to three speed Low speed, medium speed and high speed for forward driving and for backward driving the same division is made Low ,medium and high speed. A stop position to stop driving sitting also with two spot turning left and right .the combination of the keyboard and speed are

For forward driving

F: low speed

G: medium speed

H: high speed

For backward driving

R: low speed

T: medium speed

Z: high speed

S: stop running

W: turn right

Y: turn left

The control commands sends from the keyboard to drive the robot and to steer the robot is shown in the figure 4 where the keyboard event used to control servo motor and the tow DC Motors. One node written in c language on the pc called PC node receive keyboard pressed buttons and publish this event though the keyboard publisher to the raspberry where another node written on the raspberry pi board called pccontroll. The subscriber receives the keyboard events and sends it to the microcontroller in the microcontroller a function treat the keyboard events and drive the motors and servo to drive the robot forward ,backward and steer the robot .Figure 8 shows the signal following for keyboard to the microcontroller. To start operate the vehicle the rose core should enable as shown in figure 5 and then the publisher node and listener node enabled consequently as in figure 6 and 7.

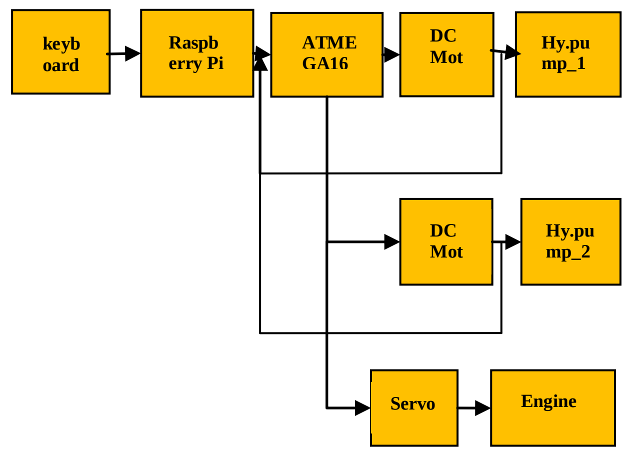

Figure 4 control structure

Figure 5 ROS core enable

ISSN:2319-7900

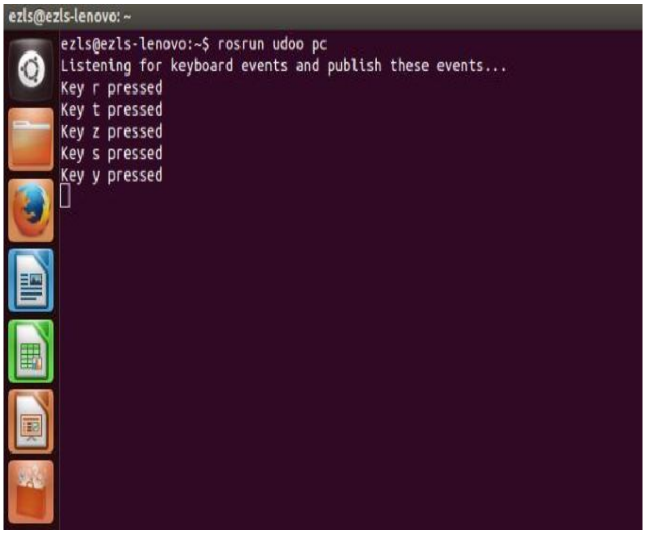

Figure 6 the first node on PC server

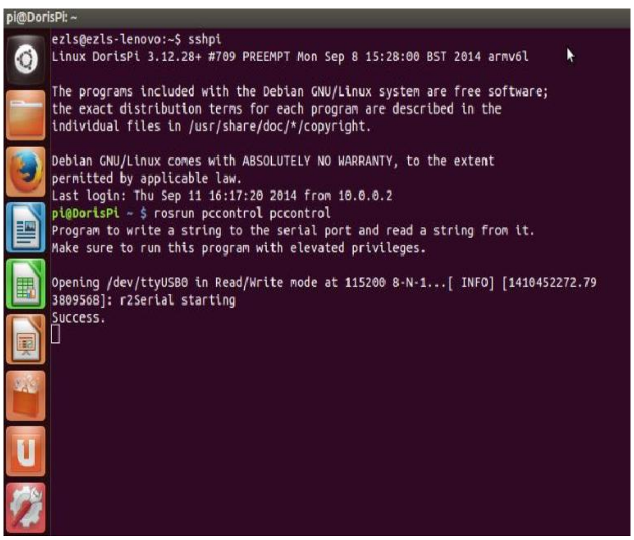

Figure 7 the second node on Raspberry pi board

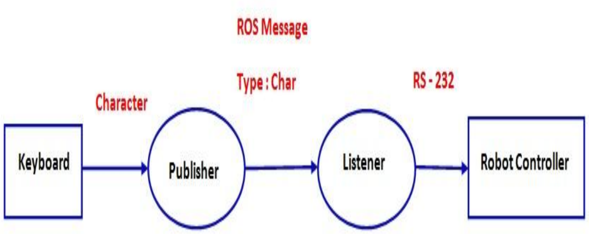

Figure 8 Ros message flow diagram

Results

The attached figures shows the DC Motor angular position for motor 1 and 2 where each Motor control the hydraulic pump swash plate position this positions give us different hydraulic pump speed and different robot speed .Figure 9 shows the both motors in stop position where the robot is not working ,figure 10 shows the DC Motors in different position where motor_1 drive the right wheel backward and motor_2 drive the left wheel forward this combination speed make the robot to rotate around its self . figure 11 shows the DC Motors in the same position but less angular position than the stop mode that make the swash plate open and the hydraulic motors and robot wheel will rotate forward with this combination the robot will move forward. Figure 12 shows the robot in the backward driving mode. Figure 13 shows the DC Motors in the middle position which gives more speed while figure 14 shows the servo motor in one position which open the gasoline engine throttle valve .Figure 15 shows the hydraulic pump swash plate positions for driving forward and backward in reality the swash plate operate within + 18 to -18 degree .

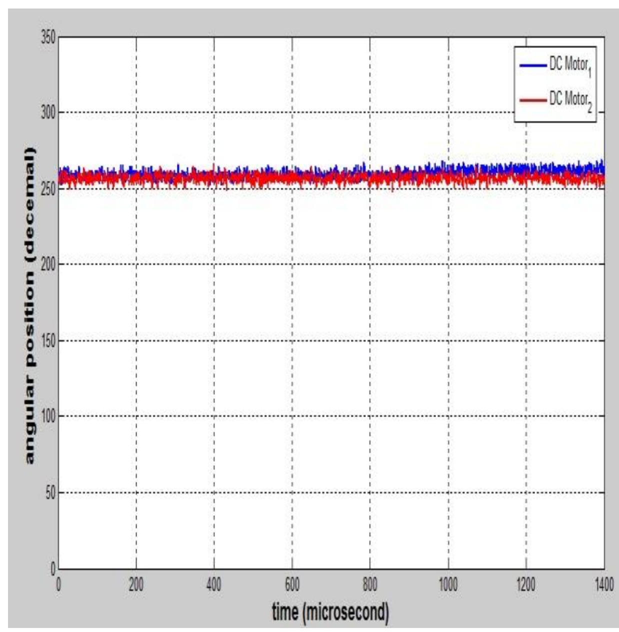

Figure 9 Robot in stope Position

ISSN:2319-7900

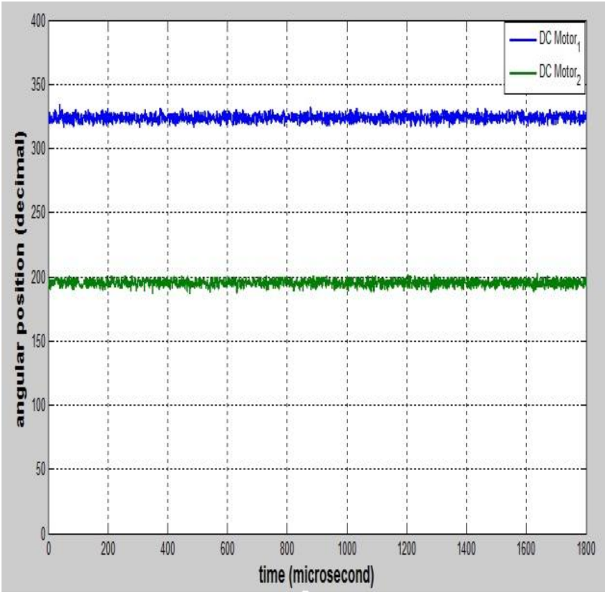

Figure 10 Robot in steer mode

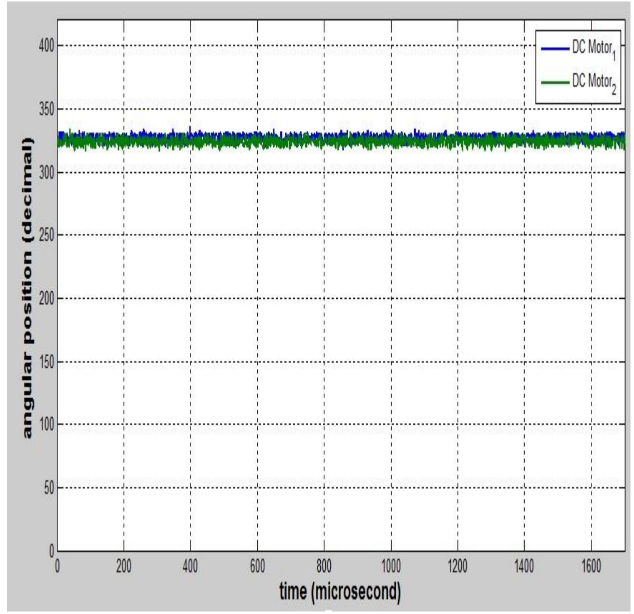

Figure 12 Robot in backward driving mode

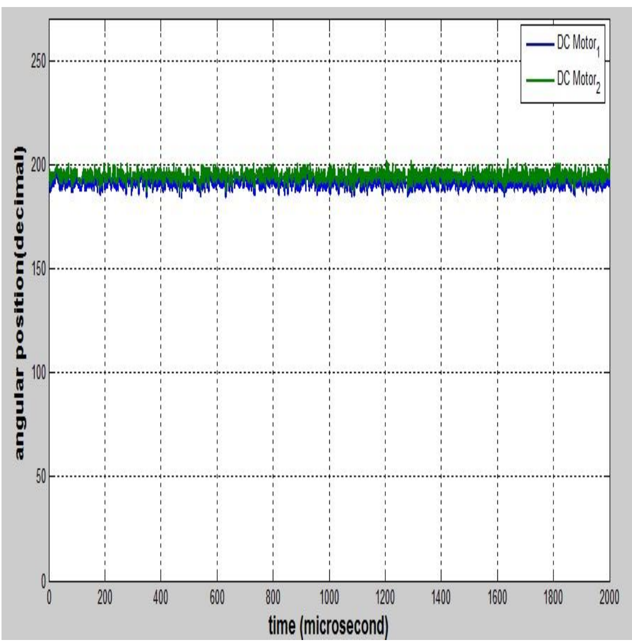

Figure 11 Robot in forward driving mode

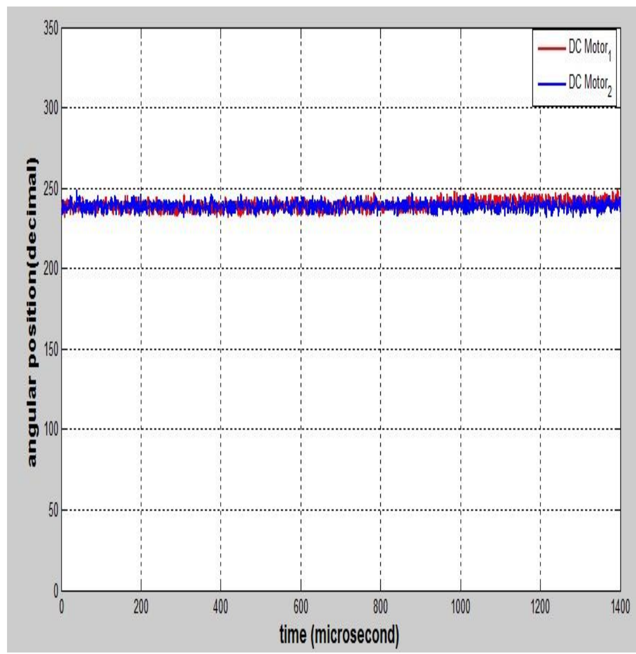

Figure 13 Robot in forward and middle speed

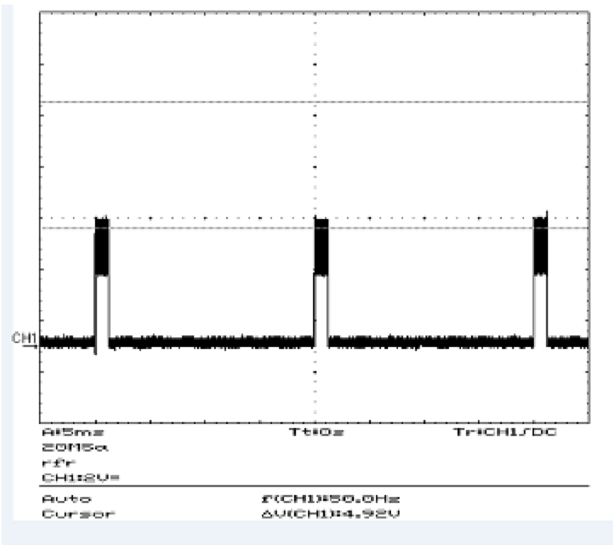

Figure 14 the servo motor position

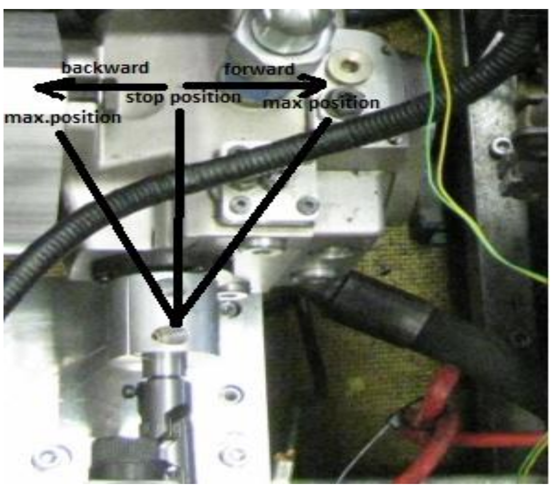

Figure 15 swash plate positions

Conclusions

The Tele-operation controller to drive the amphibious Vehicle implemented and programmed ,the results shows the actuators positions which activate the Vehicle to move forward ,backward ,stop and turns , the speed is set to be 3 speed for forward and 3 speed backward. The engine throttle controlled through the servo motor.

References

[1] Ashitey.Trebi-OllennuAn John M. Dolan, "Autonomous Ground Vehicle for Distributed Surveillance:CyberScout". Carnegie Mellon University.

[2] J. Luca Bascetta, GianAntonio Magnani, Paolo Rocco, Matteo Rossi and Andrea Maria Zanchettin. Teleoperated and Autonomous All Terrain Mobile Robot (TA-ATMR).

[3] S. John S. Jacob, ``Conversion and control of an All Terrain Vehicle for use as an autonomous mobile robot`` A thesis submitted in partial fulfillment ofthe requirements for the degree of MASTER OF SCIENCE in Mechanical Engineering 1998..

[4] Volker Grabe, Martin Riedel, Heinrich H. Bulthoff, Paolo Robuffo Giordano, Antonio Franchi. "The TeleKyb Framework for a Modular and Extendible ROS-based Quadrotor Control", 2013 European Conference on Mobile Robots (ECMR) Barcelona, Spain, September 25-27, 2013.

[5] Gourab Sen Gupta1,2, S.C. Mukhopadhyay1, Matthew Finnie ,"Wi-Fi-based Control of a Robotic Arm with Remote Vision", I2MTC 2009 - International Instrumentation and Measurement Technology Conference Singapore, 5-7 May 2009.

[6] Max Schwarz, Jörg Stückler, Sven Behnke, "Mobile Teleoperation Interfaces with Adjustable Autonomy for Personal Service Robots", In Proc. of the 9th ACM/IEEE International Conference on Human-Robot Interaction, Bielefeld, Germany, 2014.

Biographies

SAILAN KHALED. Was born in Sanaa, Yemen, in 1978.He received the B.S from Sanaa University in 2001 and M.S. degree in Mechatronics from Siegen University of in Germany in 2010 Since 20012, he is a PhD student in Siegen university institute of Real Time Systems .His research interests are control systems, embedded systems, Robotics and Autonomous Systems.

Mrak-stefen Kramer .Received his Dipl.-Ing. In Electrical and Information Technology from Siegen University. he is a research assistance in Siegen university institute of real time learn systems.