# AVIONIC CONTROL SYSTEMS FOR EDUCATION AND DEVELOPMENT

**Nils Gageik[1], Atheel Redah[2], Sergio Montenegro**

*University of Würzburg, Aerospace Information Technology (GERMANY)*
*1 & 2 contributed equally to this work (Würzburg January 2012)*
*gageik@informatik.uni-wuerzburg.de, redah@informatik.uni-wuerzburg.de, montenegro@informatik.uni-wuerzburg.de*

## Abstract

This paper describes, explains and evaluates two avionic control systems for education and development, which have been implemented as educational tools for subjects in the study path Information Technology for Aerospace. They are also being used in ongoing development and research projects of the chair.

Keywords: avionic, controlling, steering, PID, seesaw, quadrotor, quadrocopter, test case, real-time, imu, sensors, UAV.

## 1 INTRODUCTION

Traditionally Aerospace was part of Mechanical Engineering, but in the last years the greatest innovations came from the field of Information Technology. Therefore Information Technology for Aerospace is a new research branch at the University of Würzburg and unique in Germany. It combines beside other subjects Computer Science, Aerospace and Control Engineering, which is the focus of our work. To give the students a deeper understanding of these topics and teach them more practical skills, we developed two avionic control systems for the students to use. On this way the students can learn tasks like designing controllers, programming embedded systems as well as gathering and analyzing sensor information. The idea is to give the opportunity to apply theoretical background in practice on very interesting and motivating applications.

The first exercise system is a Seesaw Control System (SCS) where the students can observe, study and experiment on a seesaw with two brushed motors and propellers. Their task is to stabilize the system at desired tilt angle and angular rate, and later to improve the system response by tuning controller parameters and improving sensor data gathering and interpretation. All the system I/O signals are real-time monitored by a control panel implemented in LabVIEW [3,5].

The second exercise system is a Quadrotor Control System (QCS) which is a self-made quadrotor connected with a joint on a rod. There are two versions available of this system: One with full 3DOF (degree of freedom) movement and one which allows adjustable 1 DOF pitch, 1 DOF yaw or a combination of 2 DOF pitch and yaw movements. All the sensors (Accelerometer, Gyroscope) and the actuator's controllers (Brushless Motors) are connected via one I²C bus to simplify the communication. Using the QCS, the students can implement step by step and test their own control algorithm to stabilize the system and later to use it in real flying quadrotors. This has been proved by free flight tests.

Both systems have been demonstrated in the international aerospace supply fair AIRTEC 2011 with a very good response.

## 2 SEESAW CONTROL SYSTEM (SCS)

Creating a simple and effective training tool for the students to learn and experience with the principles and elements of closed loop control systems was the main motivation for the SCS. In this type of control systems, sensors are used to measure the actual states of the system (actual outputs) and feed this information back to the controller. The controller uses this information to adjust the actuating signal of the system accordingly.


*ISBN: 978-84-615-5563-5*

## 2.1 Hardware Components of the SCS

The system mainly consists of two brushed motors with propellers mounted on a seesaw construction and a 3-Axis accelerometer sensor (MMA7361L) used to measure the tilt angle of the system. A driver IC for dual DC motors (TB6612FNG H-Bridge) is used to drive the motors in both directions. In order to obtain sensor information and generate required Pulse Width Modulation (PWM) signals to drive the motors and stabilize the system, the actuators and the sensor are interfaced with the computer using the NI USB-6008 data acquisition unit, as shown in Figure 1 [4].

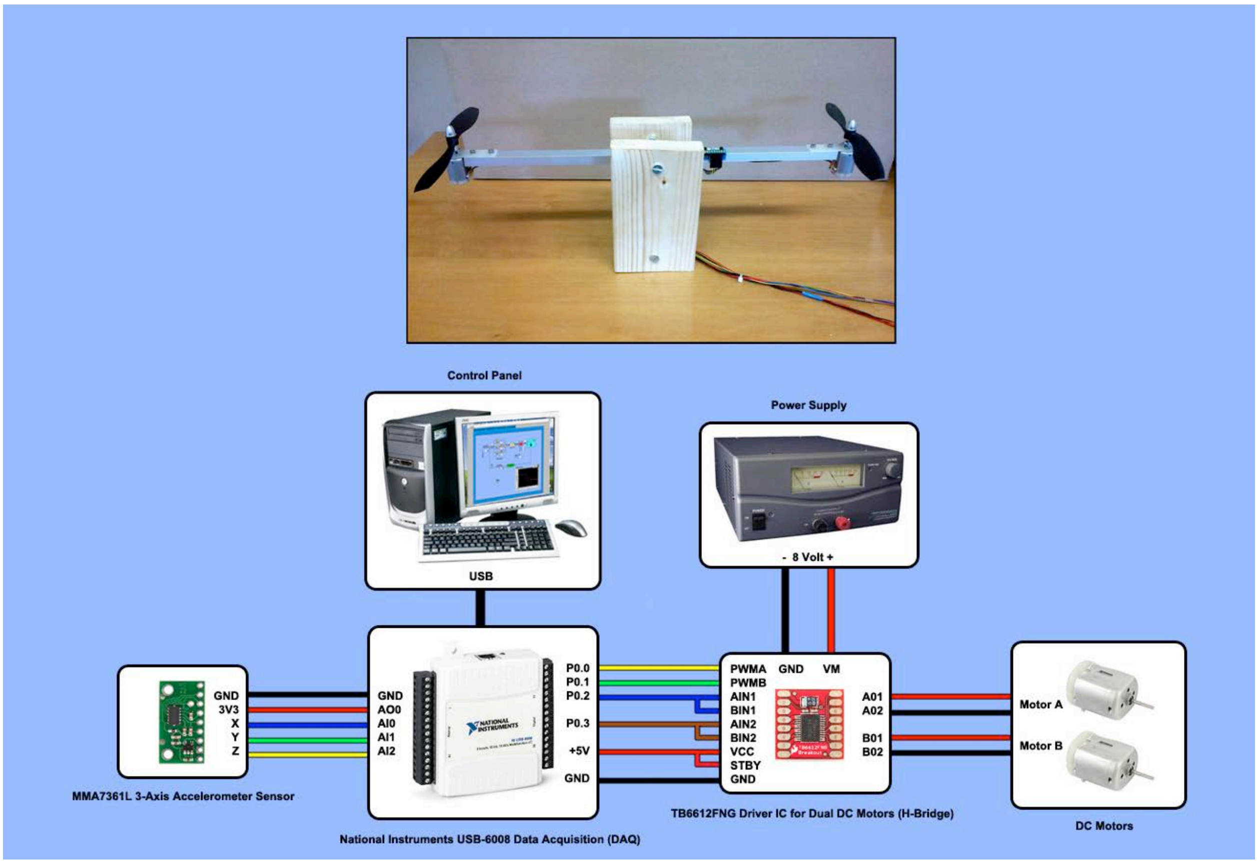


Figure 1: The Seesaw Control System (SCS)

## 2.2 Software Components of the SCS

The control panel of the system is completely implemented using LabVIEW, so the students can easily observe the transient response and all other parameters in the system in real-time. The seesaw system can be used in two main controlling tasks.

### *2.2.1 One Motor Control*

In order to steer and stabilize the system around a desired tilt angle, a Proportional-Integral-Derivative (PID) Controller is used to control the rotation speed of a single motor. The students can experiment with the PID controller parameters (P, I, D) in order to stabilize the system and then improve the system response by tuning these parameters for a desired transient response. Thus, they understand the effects of each parameter on the response and the behavior of the controller. Furthermore, parameters which can affect the robustness of the controller like sampling time, PWM frequency and filtering sensor information can also be studied and analyzed, as shown in Figure 2.

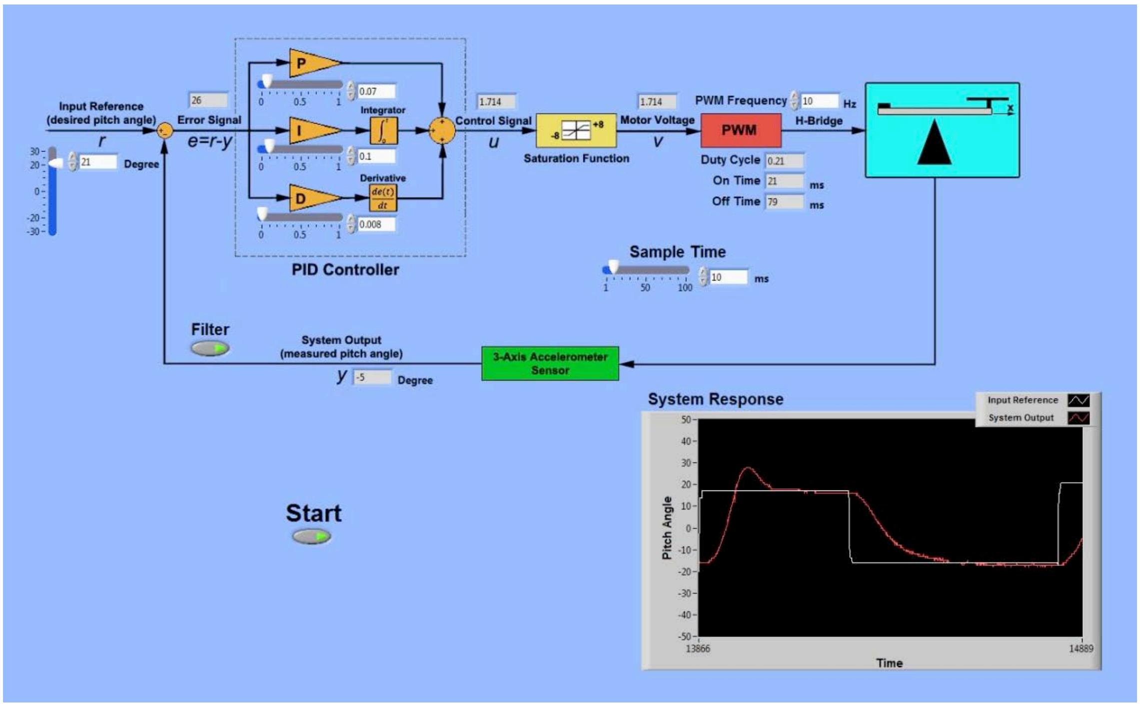


Figure 2. One Motor Control Panel of the SCS

### 2.2.2 *Two Motor Control*

In this task, the PID controller will calculate the required ratio needs to be increased and decreased in the two motors respectively for stabilizing the system around a desired tilt angle and also at desired system speed using voltage ratio algorithms. The applied voltages needs to be governed by a set of equations which ensure the sum of the voltages at a desired tilt angle is maintained while varying the input voltage to each motor, as shown in Figure 3.

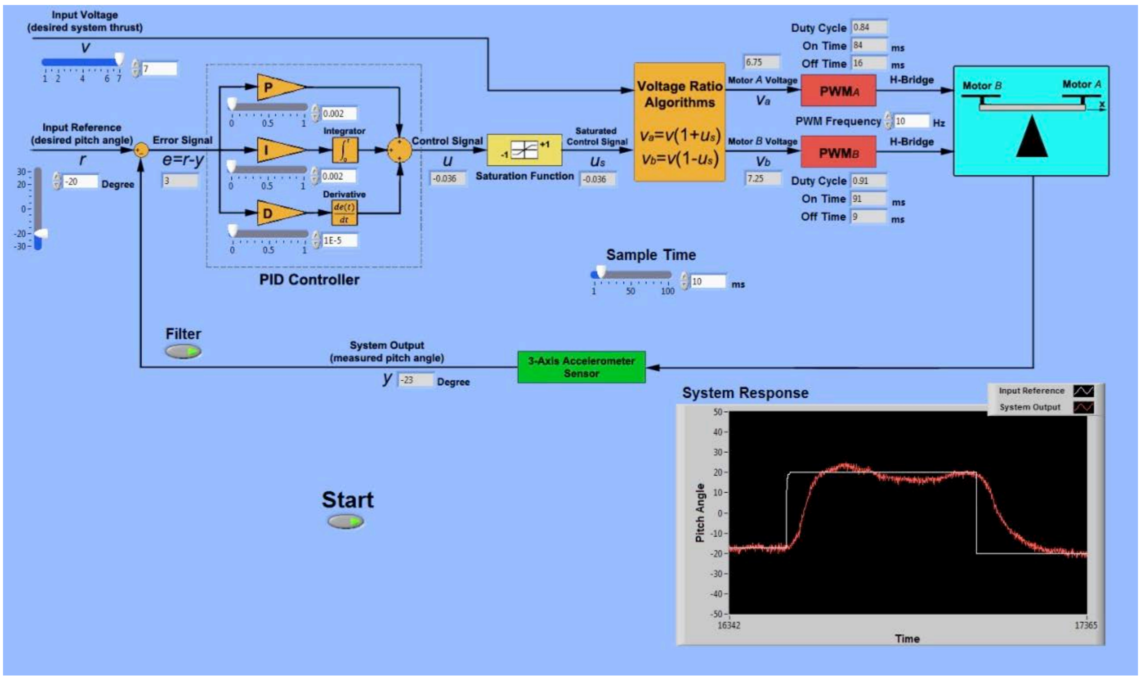


Figure 3. Two Motor Control Panel of the SCS

### *2.2.3 Calibration*

Before the students can continue with the controlling tasks, they need to be familiar with the accelerometer sensor and how it can be used to measure the tilt angles and also try to improve the quality of its data by calibrating the sensor at various different positions, as shown in Figure 4.

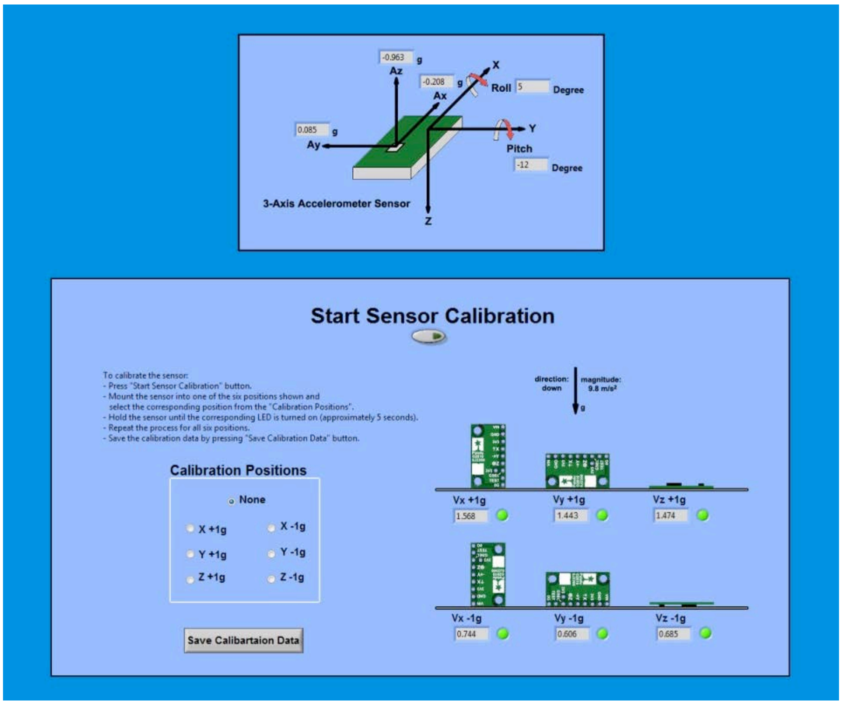


Figure 4. The Sensor Calibration Panel of the SCS

## 2.3 Conclusions for the SCS

Working with this type of simple and open system teaches the students the basic knowledge and experience they need to handle the design of more complex control systems. The main criteria in which the students benefit of using this system can be summarized in the following points:

**a.** Closed loop control system principle (feedback control).

**b.** PID controller principles and the effect of each controller parameter (P, I, and D) on the system response (experimentation).

**c.** Understanding how the output of the PID controller adjusts itself to the required input for stabilizing the system around a desired point.

**d.** Understanding the PWM signal and how it can be used to control an analog system by adjusting its duty cycle in real-time, and also the effect of the signal's frequency on the applied average value.

**e.** Calibrating 3-Axis accelerometer sensor and retrieving tilts angles information using readings from the 3-Axis.

**f.** Understanding the sampling time effect (the timing of the control loop) on the control system behavior (discrete control).

**g.** Understating the effects of filtering sensor information on the gathered data.

# 3 QUADROTOR CONTROL SYSTEM (QCS)

The main idea of the QCS was to design a simple tool which supports students to learn embedded programming in a real time environment using an application from Aerospace. But the QCS provides much more. It is also a very good example to make practical experiences for a very fascinating and motivating application and that way it can be used as a tool to learn in practical the interdisciplinary meaning of math and physics, computer science and embedded programming, measurement and controlling as well as mechanical and electrical engineering.

Beside the functionality for education the QCS can be used as a test case to develop a quadrotor. Here it already paid out as a first step to create our own quadrotor from scratch helping in debugging, testing and parameterizing.

## 3.1 Hardware Components of the QCS

The approach of the hardware design follows the idea to use a powerful (32bit) and easy to use microprocessor and a simple communication system with few wires. Therefore the decision came to use the UC3A from Atmel [6] because of the good software availability (see next chapter) as well as the provided microcontrollers. We use the EVK1100 development board because of the wide spectrum of useful debugging peripherals like two RS232 interfaces, a display, several push buttons and dedicated I²C and SPI connectors.

The communication of the system uses I²C (Inter-Integrated Circuit), because with I²C it is possible to connect all peripherals (sensors and actuators) on one bus. This means a minimum number of wires and drivers are needed to run the system.

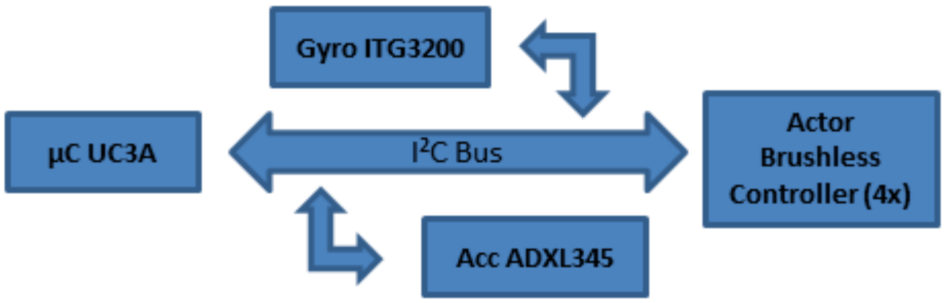


Figure 5. The Hardware Components of the QCS

The hardware components of the system (Figure 5) are the microcontroller UC3A, the IMU (Inertial Measurement Unit) consisting of the ITG3200 gyroscope and the ADXL345 accelerometer and the four brushless controllers, which drive the brushless motors. All devices are connected via I²C. Though using one sensor would be enough to implement a control system, both sensors provide us with the possibility to implement more advanced control routines like data fusion with Kalman filter or further approaches. Using at least two MEMS sensors is state of the art in a UAV (unmanned aerial vehicle) control systems [1, 2] and useful for the final system, when real flights are performed. Further components like magnetic compass, infrared, ultrasonic and pressure sensors can be added without effort using the I²C bus. Still the microcontroller provides enough I/O lines (USART, SPI, GPIO) to connect all other kind of peripheries like Bluetooth, Wifi, Camera, GPS and so on.

## 3.2 Software Components of the QCS

The Software of the system is designed using the Atmel Framework (Technical Library) within AVR32 Studio. The advantage of this approach is that examples (drivers) for nearly all interfaces (I²C, ADC, USART, SPI) and functionalities (TC, GPIO, DIP, IRQ) are given, so results can be achieved very fast. This allows the students first to learn the usage of all drivers in the first semester of our lecture and later use this knowledge to develop a complete embedded control system.

The final software version of the system merges basic drivers like timer counters (TC), I²C, GPIO (optional for buttons as user interface), SPI (optional for display of user interface) and USART (optional to send commands and debug information) with the application software containing of PID Controller, quaternions and filters. The control loop operates on a sample time (in the area) of 10ms, which is fast enough for a very stable control behavior in this application. The microcontroller could even handle a faster sample time, but using a higher sample time provides enough scope for sending a lot of debugging information, generating quaternions and filter calculations (Kalman), so the controller does not lacks of an unstable sample time.

Though finding the right sample time can of course be part of the exercise, this is not trivial to be figured out by inexperienced students beside many other parameters and then could be given as well as a template of the whole software system, filled out depending on the skill level of the students. Using the QCS students can learn to implement all level of software from low level drivers to create I²C, USART etc., over communication protocols and settings for peripherals (sensors and actuators) to the application software like algorithm for controlling and steering. A diagram of the final software version which is capable to fly is shown in Figure 6.

The system can be controlled using a PD (Proportional-Differential) Controller only, which is the first approach the students use. That way the students learn in practical the meaning of every control parameter and how to set them up with an empiric approach. The next step is then to add an I-Part to remove the stationary control deviation.

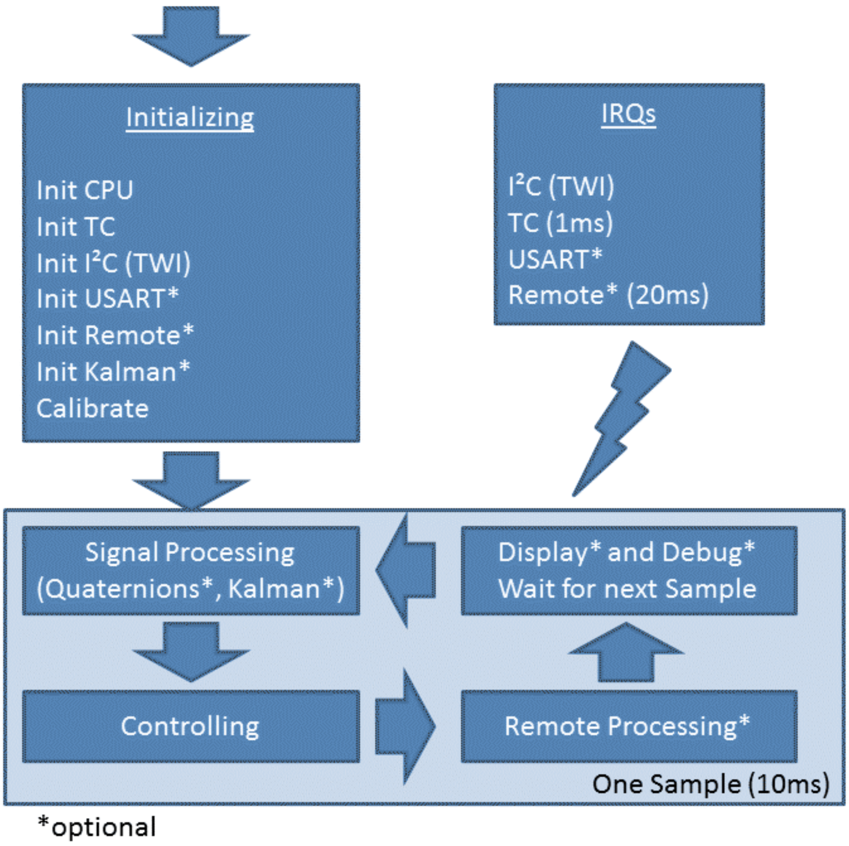


Figure 6. The Software Components of the QCS

## 3.3 Mechanical Design of the QCS

There are two different versions, which allow the development, implementation and parameterizing of each controller for all axes separately. Later these controllers can be merged using superposition to create the final triaxial controller.

The first version (2 DOF) uses a seesaw on a rode which can yaw (Figure 8). The seesaw as well as the rode can be fixed or set rotatable meaning the first version may perform a 1 DOF pitch or yaw movement or both at the same time. Using this device all parts of the triaxial controller including its parameters and the fusion algorithm can be tested separately and altogether. Furthermore it is possible to make the system lift until to a certain level, whereby it has up to 3 DOF and can be used to test a height controller.

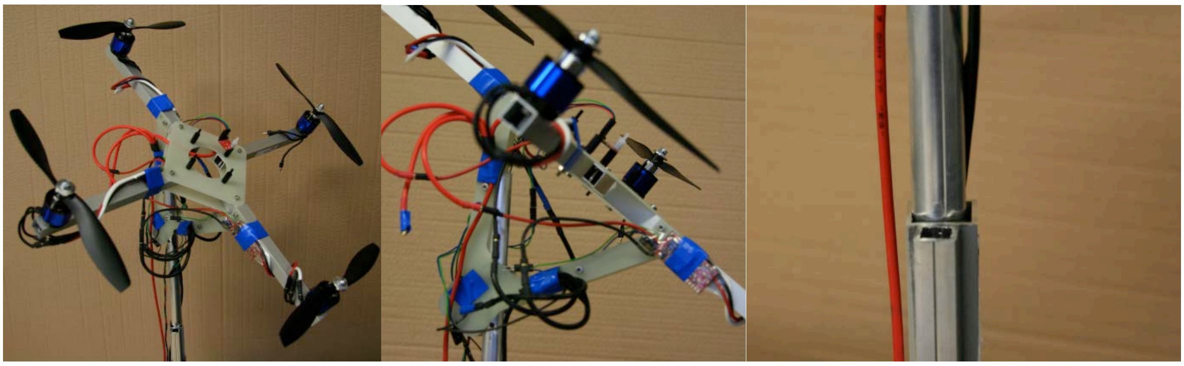

Figure 8: The Mechanical Design of the QCS (Version 1)

The second version (3 DOF) is an axial joint from IGUS (AGRM-08, [7]) on a fixed rode. Using the axial joint the quadrotor can manipulate its orientation in all 3 rotation axes without changing the position meaning the 3 translational axes. This  allows safe tests on the ground. Because the rode is inside a shaft the system even would be able to take off, if the motors are provided with enough power, enabling a flight test (4 DOF). In a carrying on step with this setup a height control could be implemented using infrared or ultrasonic sensors.

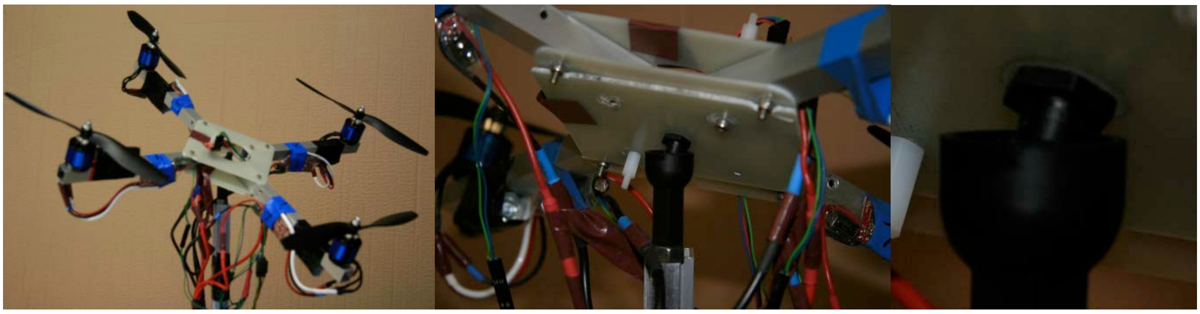

Figure 9: The Mechanical Design of the QCS (Version 2)

## 3.4 Conclusions for the QCS

The QCS proved to be a successful learning tool, because it motivated students to improve their skills and knowledge in interdisciplinary subjects (see above) and teaches them practical skills beyond theoretical knowledge.  Many students became interested to continue working with this fascinating topic in ongoing tasks doing a student, bachelor or master thesis with quadrotors.

Beside the usage for education, the QCS could also be used to completely test the software for a quadrotor. A on the QCS optimized software and control system was able to control and fly the quadrotor with exactly the same PID parameters. Nevertheless for an optimized flight the PID parameters had to be adapted, because the fixed system has a different transfer function then the quadrotor does when flying.

One big issue of this approach was the use of I²C for mainly all kind of input/output. Though I²C simplifies the communication structure, the system reacts very sensitive on errors on the I²C bus. For the QCS this is of little interest, as failure are not critical, but for real flights this was a big problem and could only be overcome by changing the I²C framework drivers adding check, timeout and reset functions.

## REFERENCES


[1] Autonomous Flying Robots, Spring 2010, Kenzo Nonami

[2] I4Copter: An Adaptable and Modular Quadrotor Platform, SAC 2011, Peter Ulbrich

[3] Introduction to Feedback Control Theory, CRC Press 1999, Hitay Özbay, ISBN: 084931867X.

[4] PWM Control of a Magnetic Suspension System, IEEE Transactions on Education 2004, Volume 47, Issue 2, pp. 165 – 173, W.G. Hurley, M. Hynes, and W.H. Wolfle

[5] LabVIEW for Everyone, Prentice Hall 2005, James Kring, and Jeffrey Travis, ISBN:0131856723.

[6] AT32UC3A Series Preliminary, Datasheet Online Accessed January 2012, http://www.atmel.com/dyn/resources/prod_documents/32058S.pdf

[7] igus GmBH, http://www.igus.de