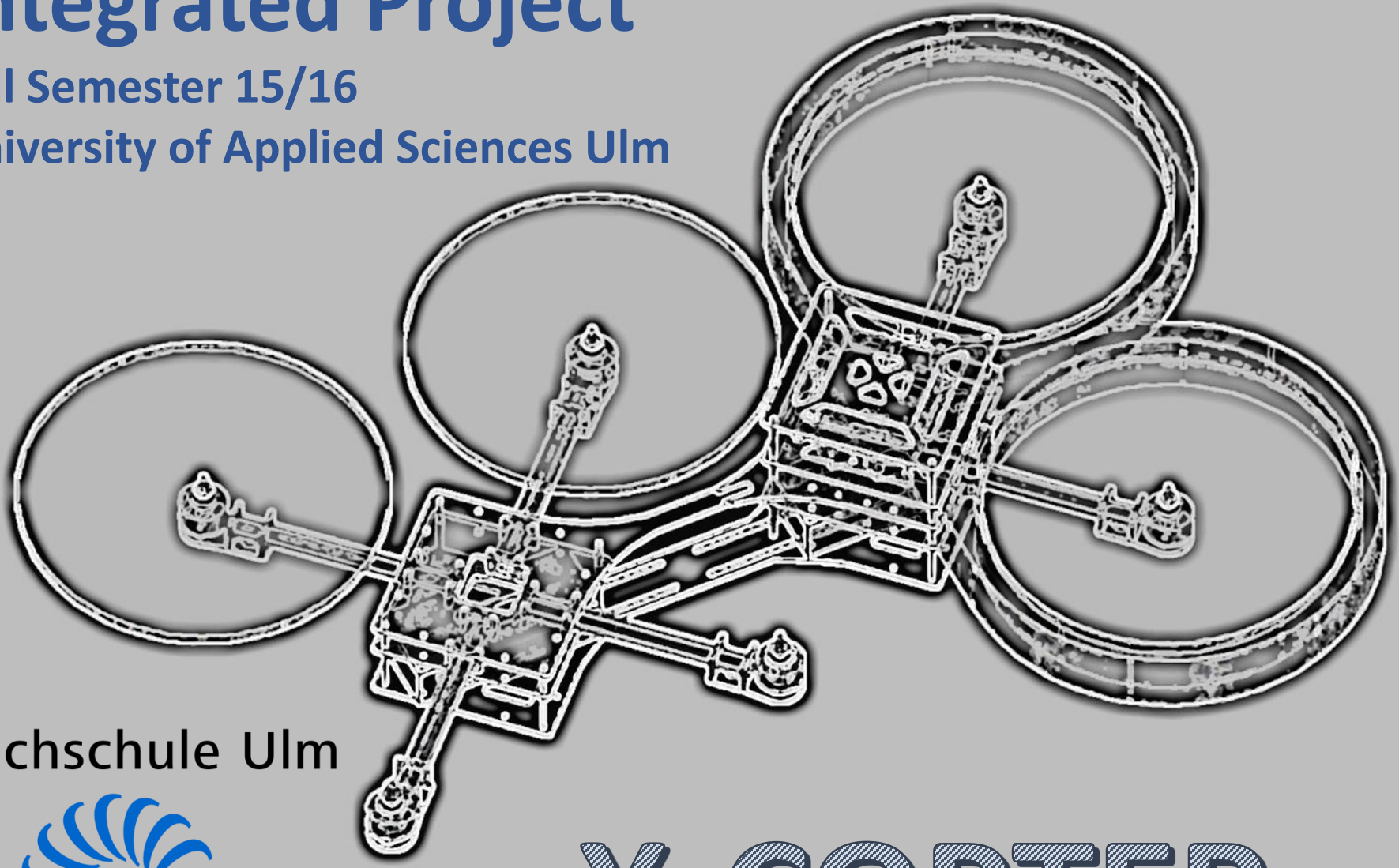
Integrated Project

Fall Semester 15/16

University of Applied Sciences Ulm



Hochschule Ulm

X-COPTER

Table of Contents

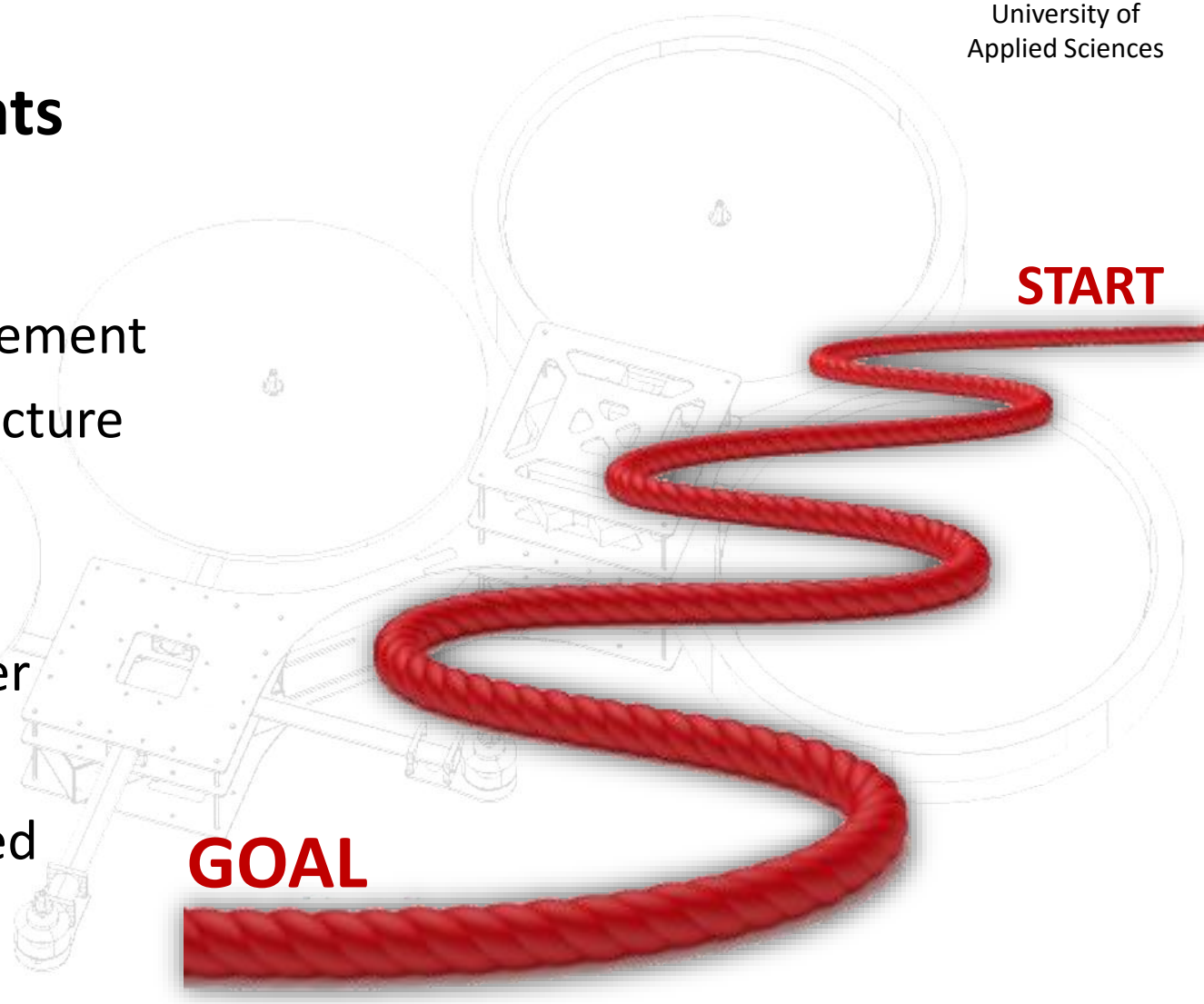
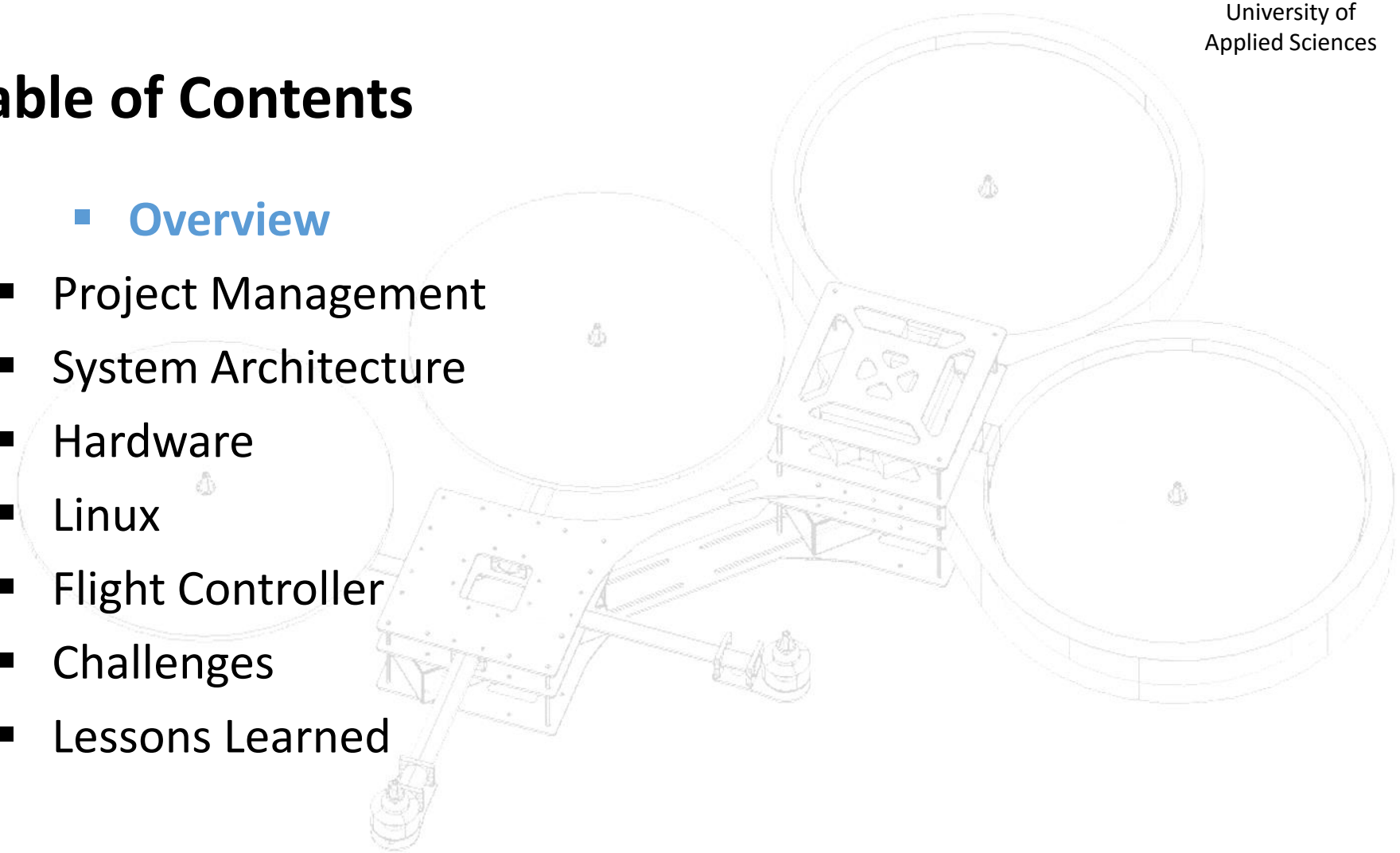
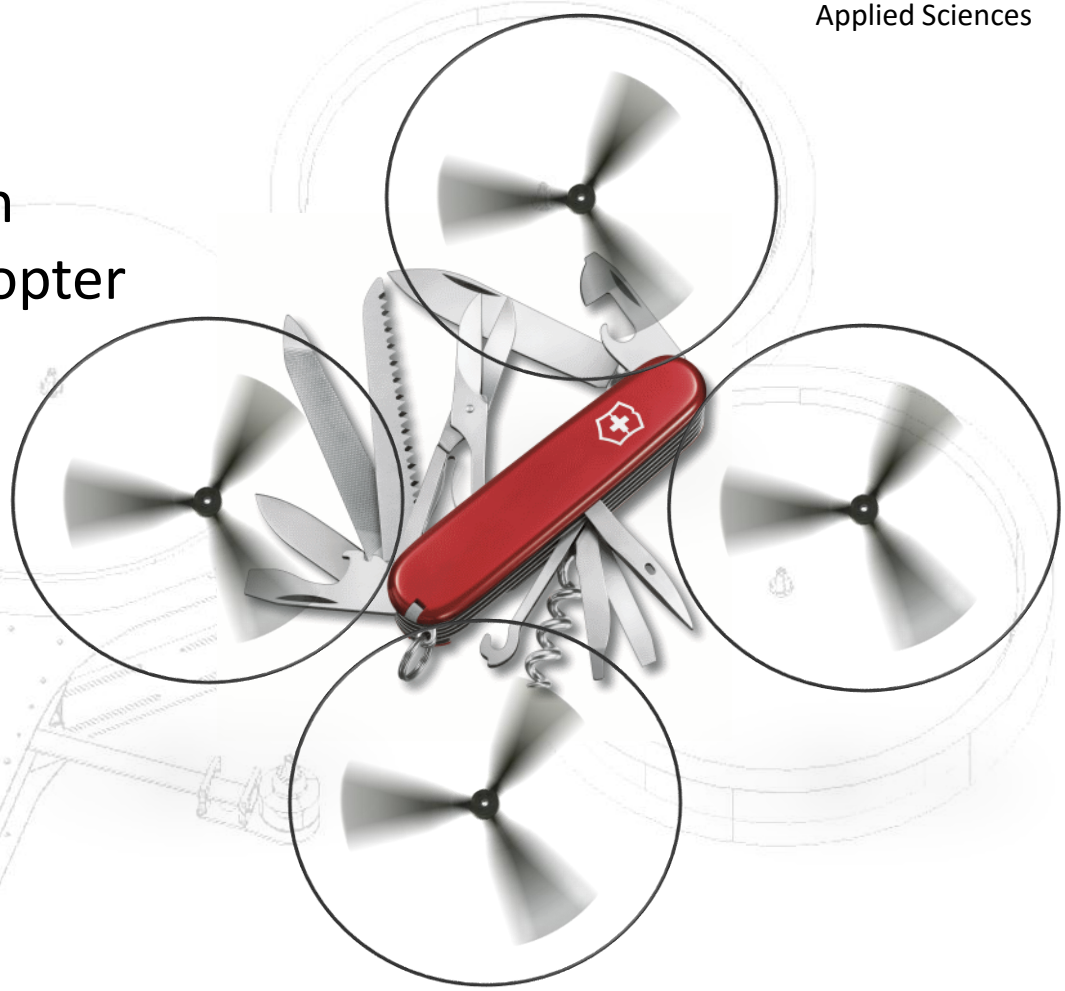


Table of Contents



Introduction

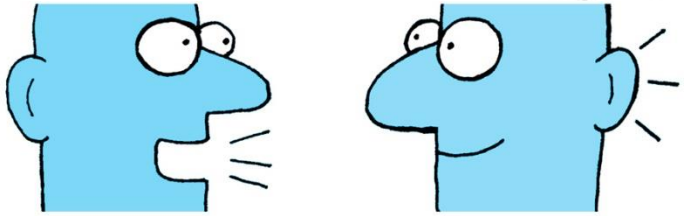
The X-Copter is planned as an autonomous modular multicopter



Fields of application

- 3D Mapping
- Rescue Missions
- Surveillance
- Parcel delivery
- Movie Production

Customer Needs



Basic Requirements

- Modularity (4/6 rotors)
- Stable power supply
- Sturdy construction
- Remote control
- Test flight

Flight Controller

- Test with commercial Flight Controller
- Own Flight Controller

Logging and Sensors

- Evaluating sensor data
- Logging of all sensor data
- Visualisation on Ground Station

Future Development

- 3D Environment Mapping
- Autonomous flight
- Collision control

Initial state of the Project

- Finished Physical model
- Custom power supply circuit
- Interfaces (I2C, PIOs, PWM)
- Interface circuit for sensors and actors
  - Sensors (gyroscope, accelerometer, compass)
- SoPC for the FPGA-part of DE1-SoC
- Interprocessor communication with FIFOs
- Drivers (partially tested)
- (Some sort of USB interface)

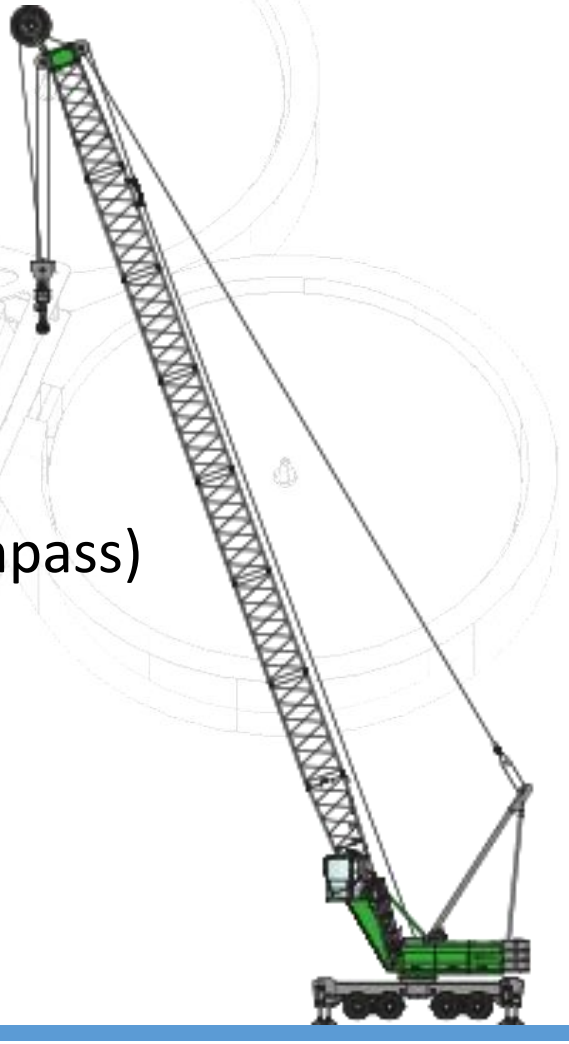
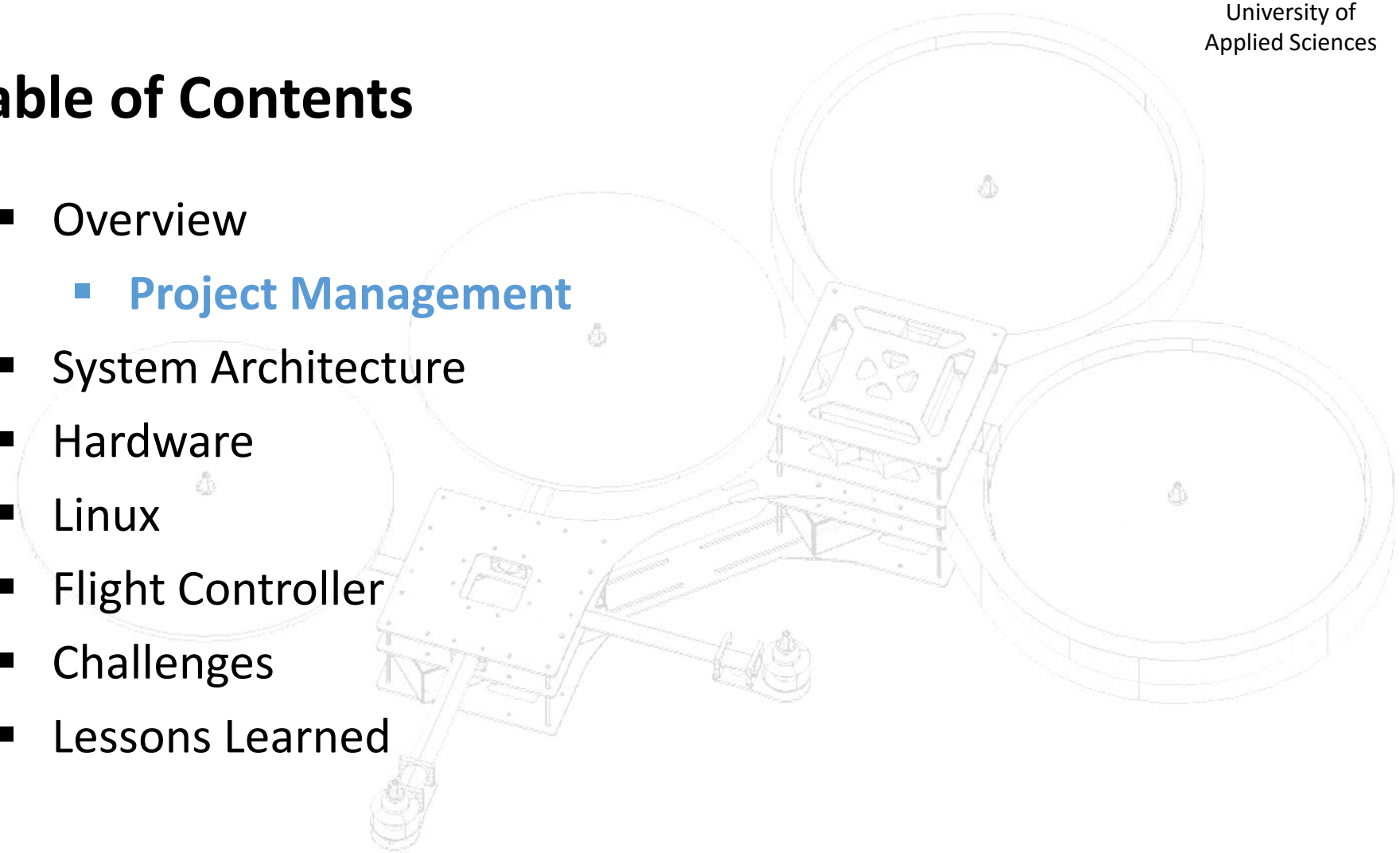


Table of Contents



Project Management

Scrum:

Changes in Project Management

Scrum:

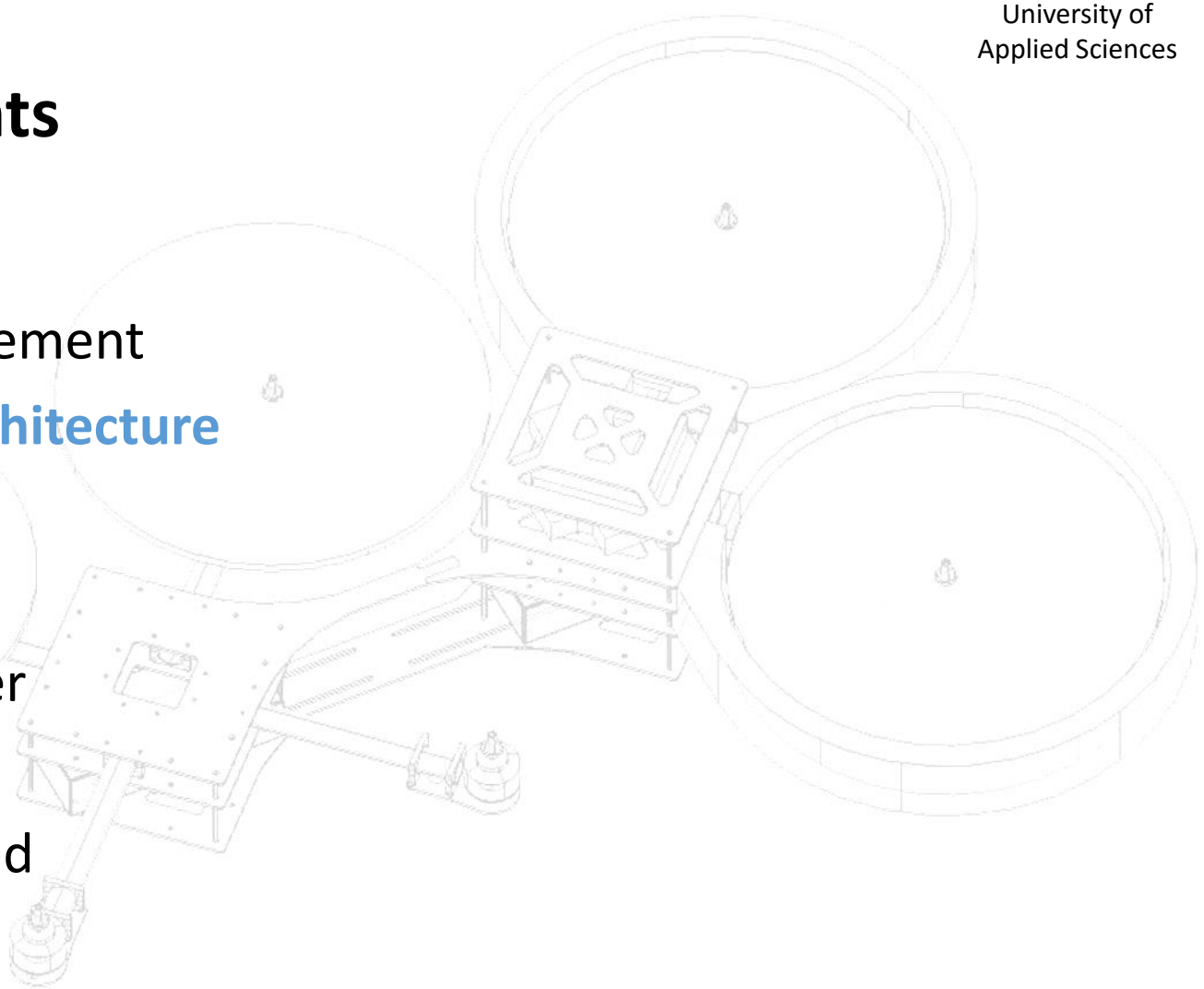
Positive	Negative
Daily Scrum	Sprint Planning
Taskboard	Sprint Review
	Sprint Retrospektive

Conclusion: Scrum only works if the team meets every day

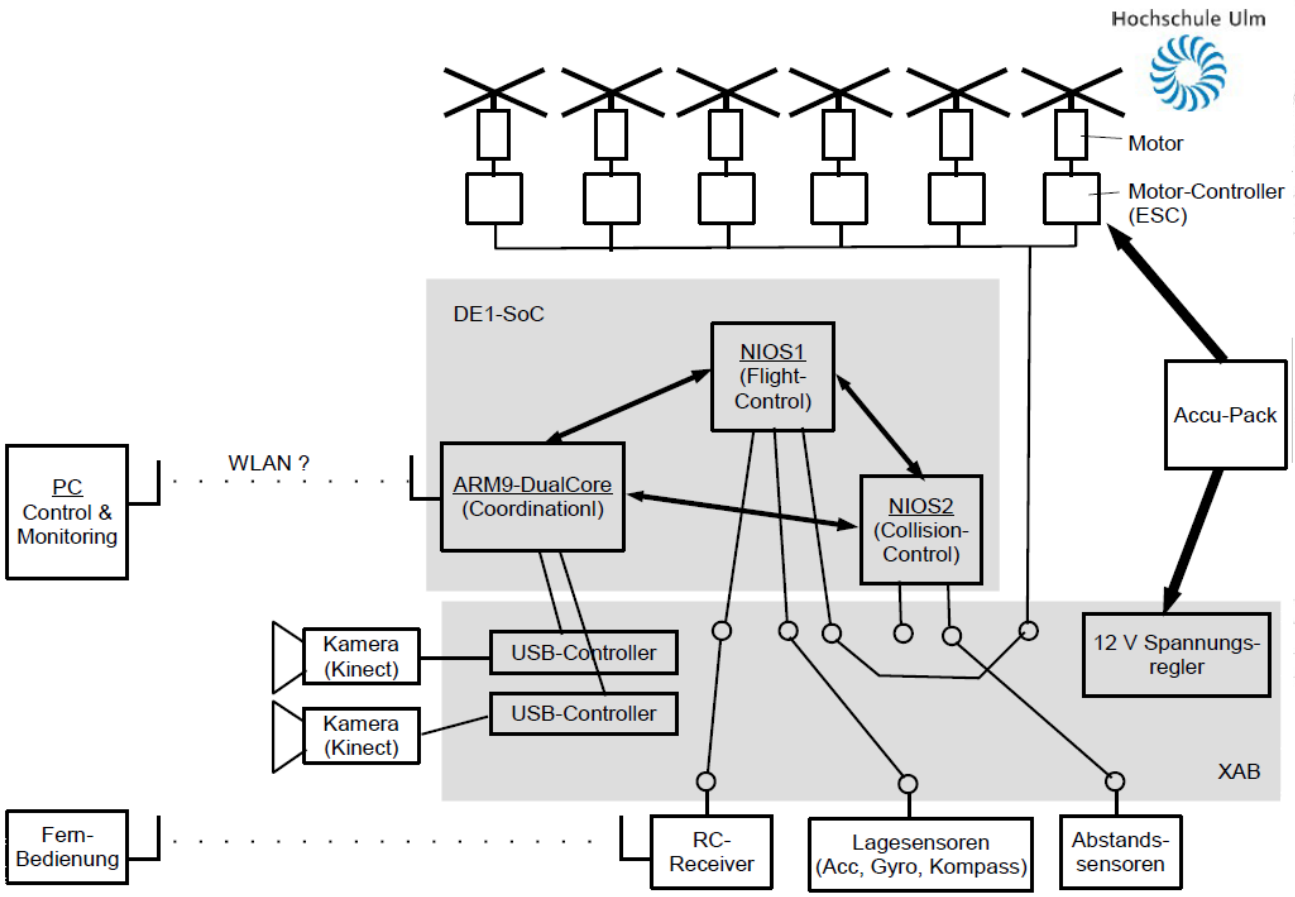
Flight test of the model:

Flight test of the model:

Table of Contents



Old System Architecture



New System Architecture

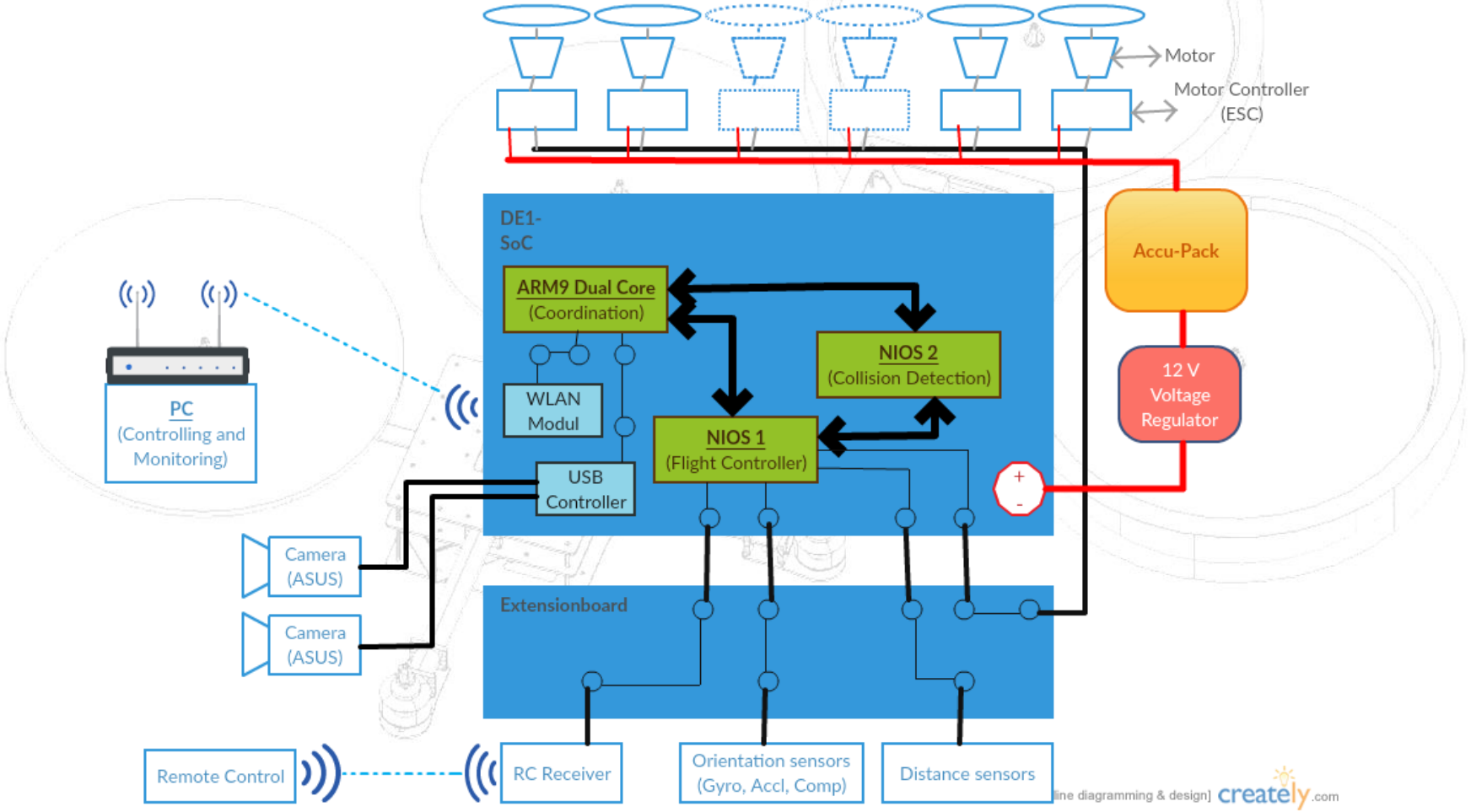
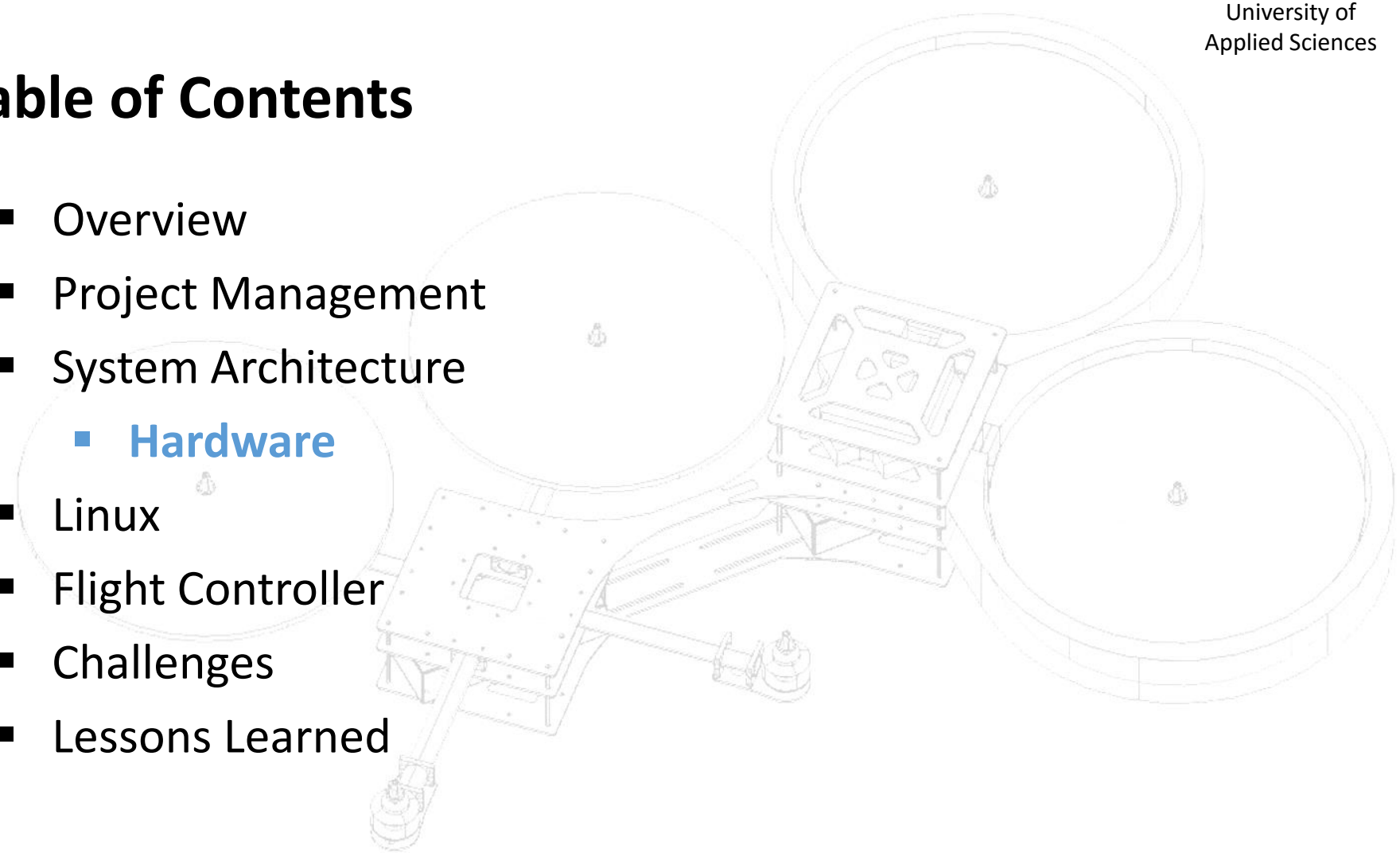


Table of Contents



Additional hardware

Commercial power supply

- 5-30 Vin
- 5-12Vout
- max 6A output

Graupner MX-16 RC transmitter

Gr-16 2.4 GHz RC receiver

RC Receiver

Graupner MX-16 HoTT Transmitter

GR-16 2.4 GHz Receiver supporting

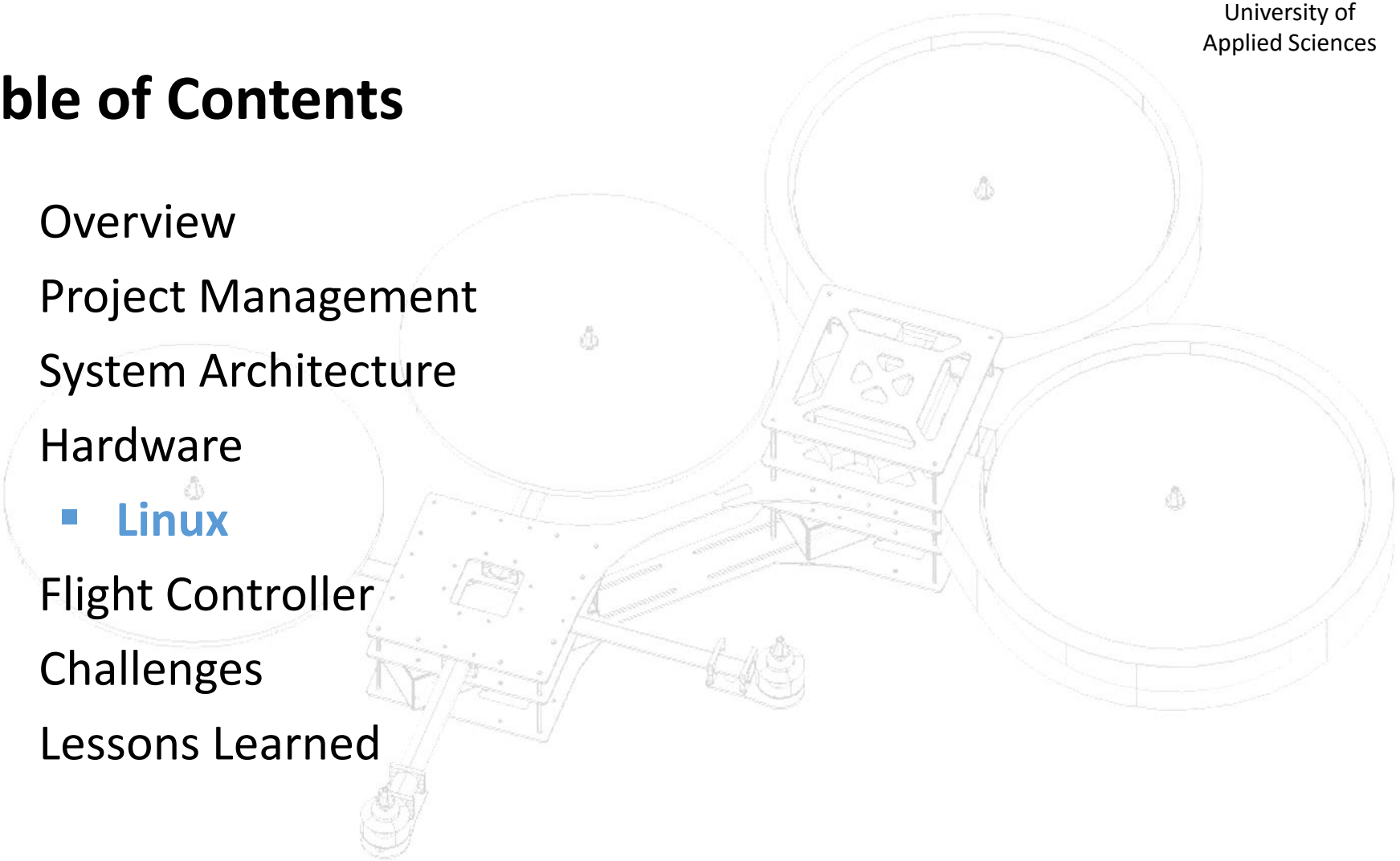
- Sum signal (PPM)
- Serial sum signal (SUMD)

Support for telemetry data

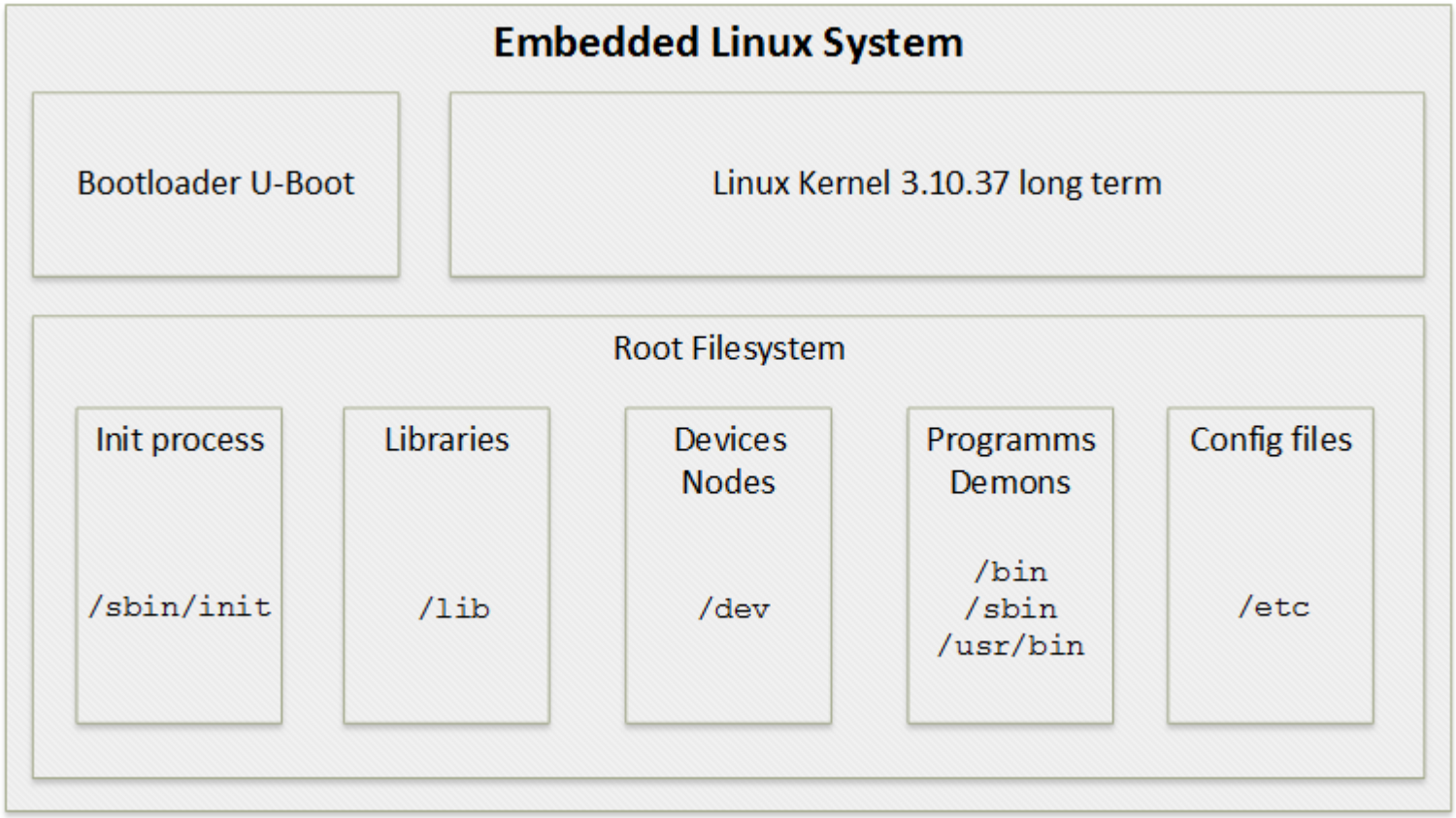
Graupner SUMD protocol

- Signal can be read by UART
- Updating every 10 ms
- 16 bit output resolution
- Fairly easy to parse
- In contrast to PPM no real-time encoding required.

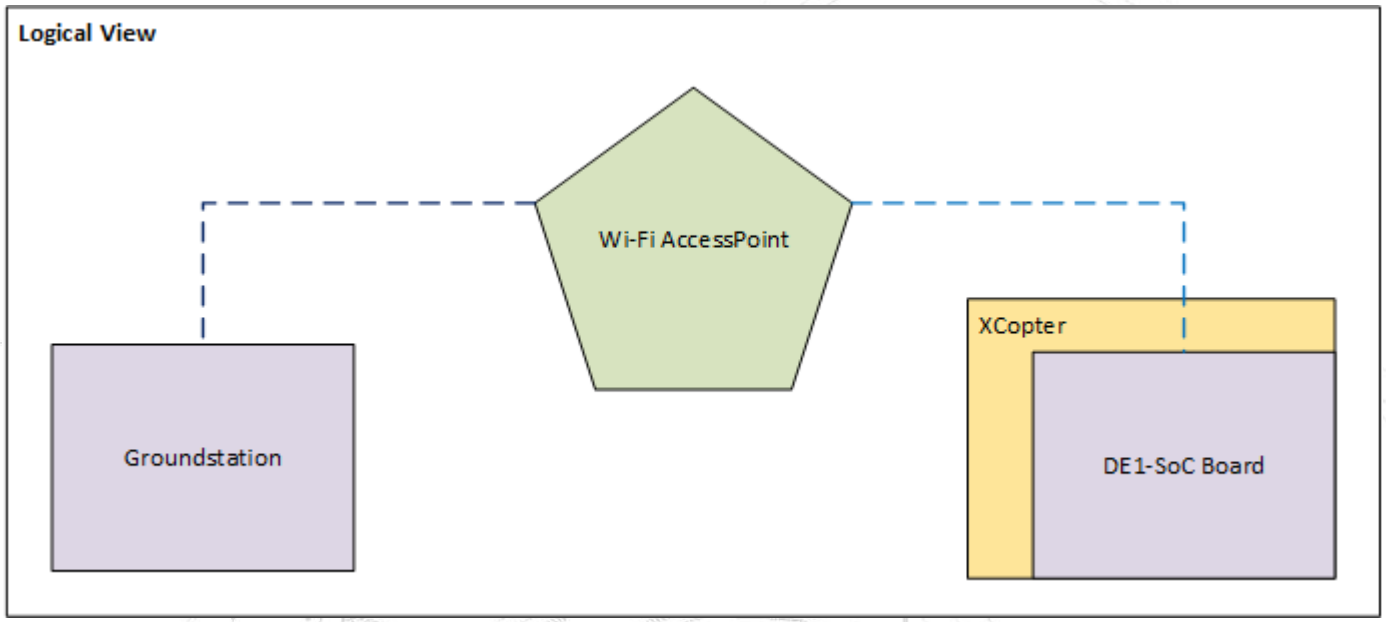
Table of Contents



Buildroot


Embedded Linux System
Bootloader U-Boot
Linux Kernel 3.10.37 long term
Root Filesystem
Init process
/sbin/init
Libraries
/lib
Devices
Nodes
/dev
Programms
Demons
/bin
/sbin
/usr/bin
Config files
/etc

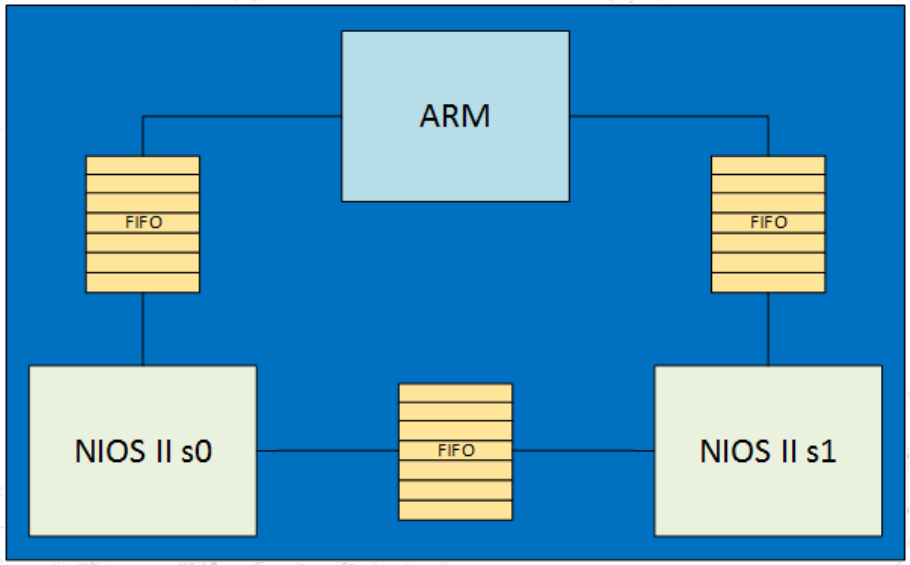
Wi-Fi

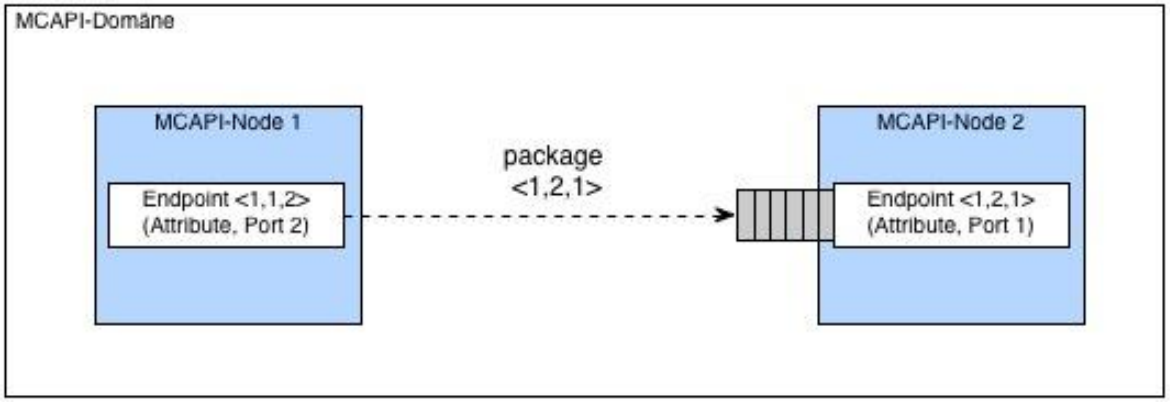


- Automatically connects at start up
- Stable link to access point
- Encrypted communication with WPA2

MCAPI

- Interprocessor communication
- FIFO based





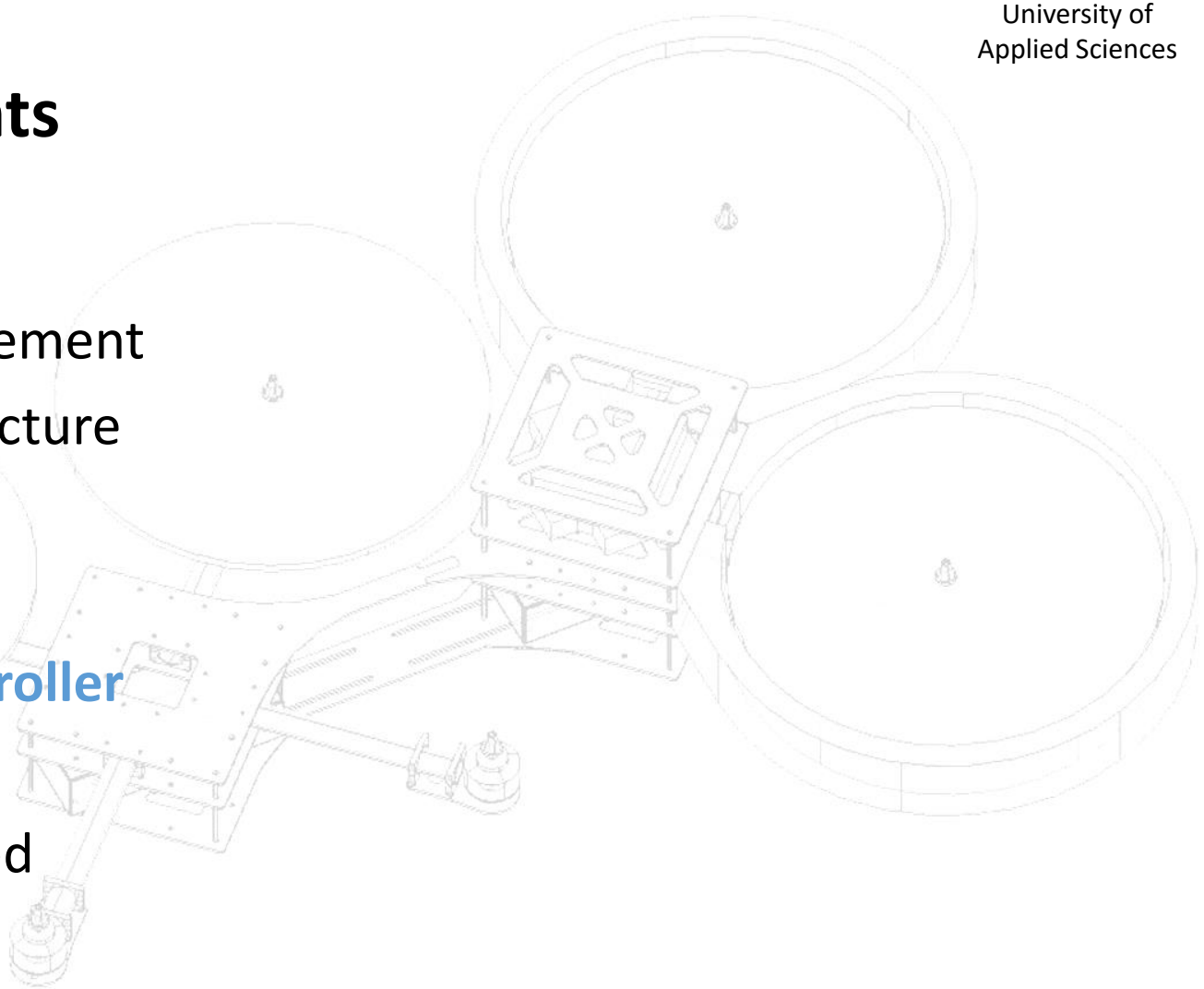
MAVLink

Micro Air Vehicle Communication Protocol

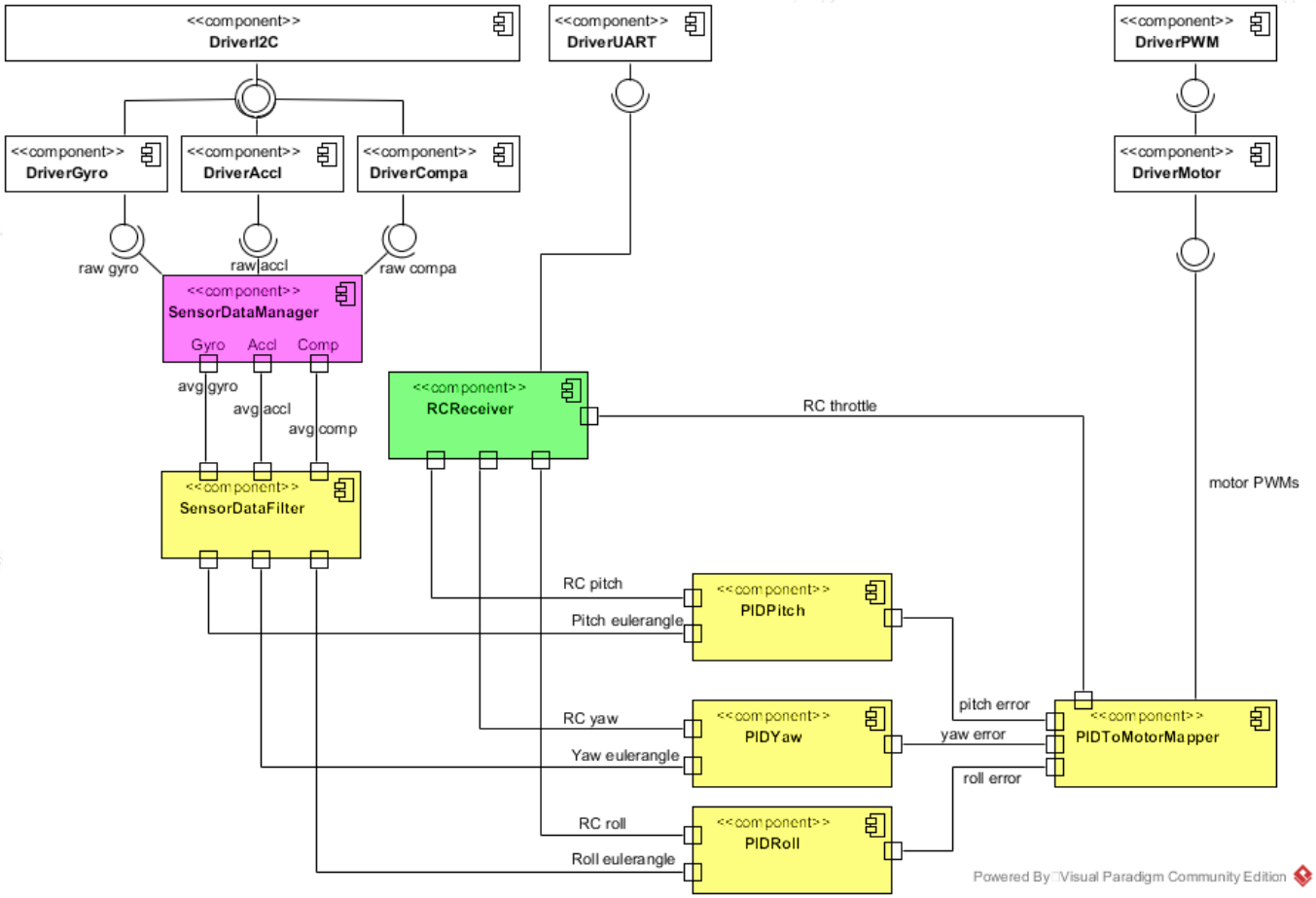
- Communication protocol between UAV and ground station
- C library for header based messaging
- QGroundControl

MAVLINK
MICRO AIR VEHICLE COMMUNICATION PROTOCOL

Table of Contents

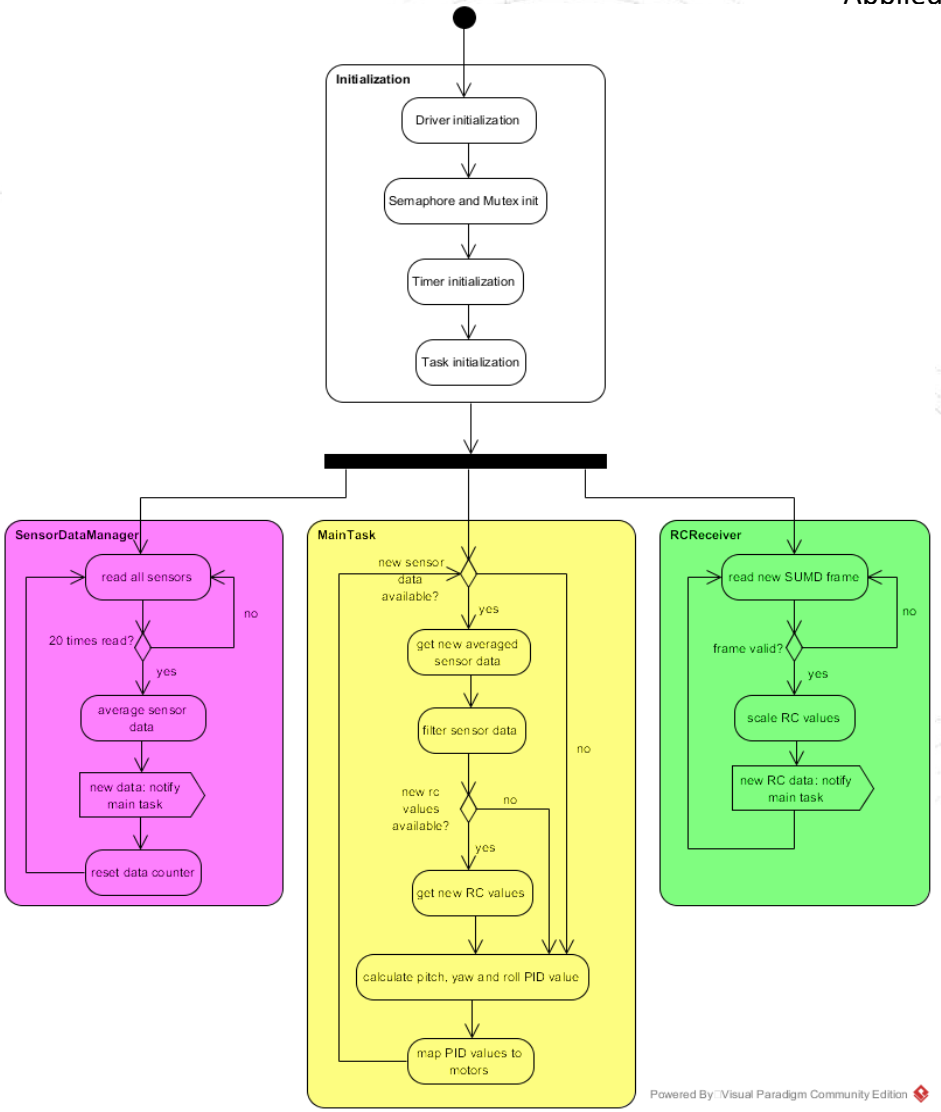


System Architecture

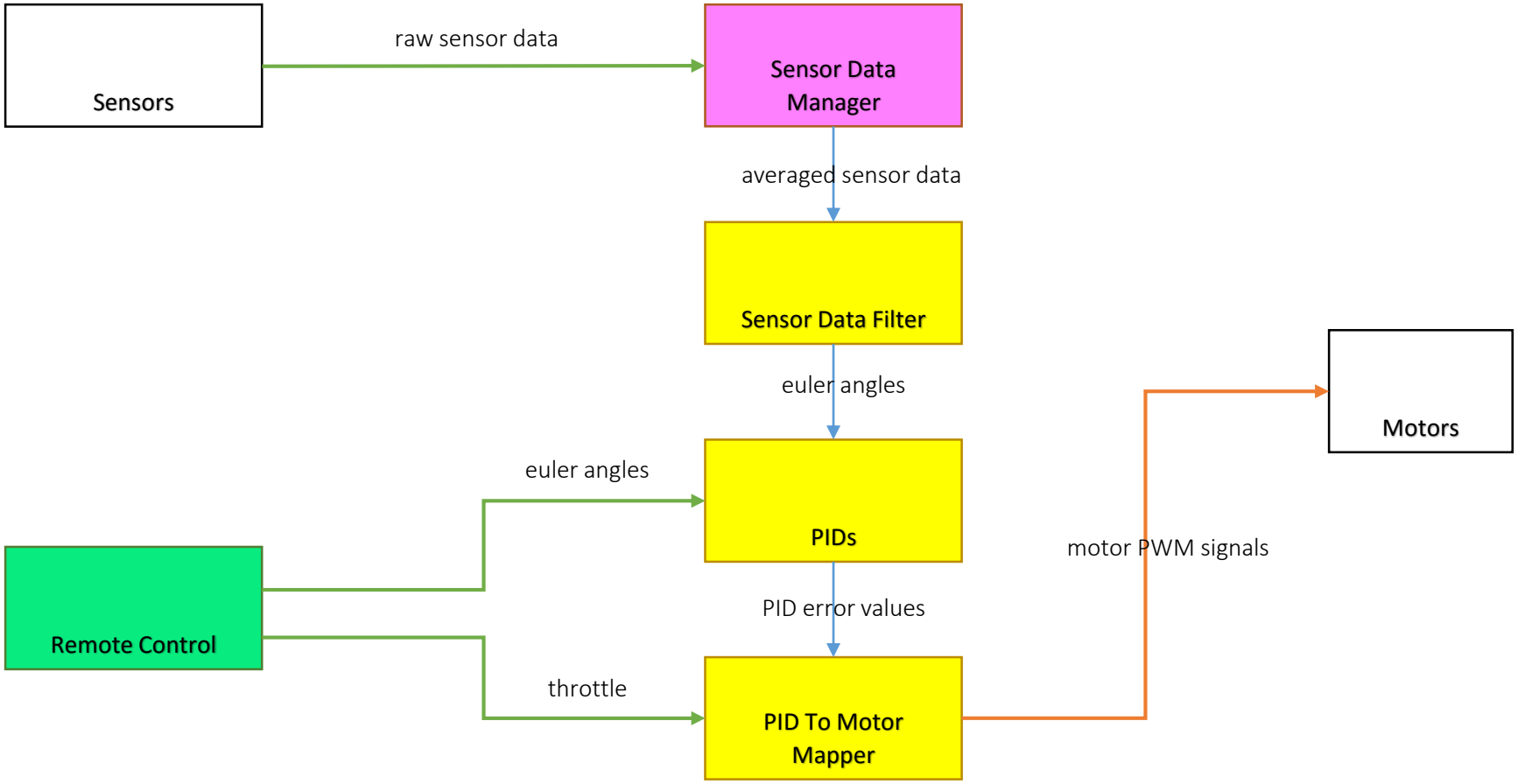


System Program Flow

1. Init state
   - Driver, Mutex, Timer & Task initialization
2. Split into three tasks
   - Sensor Data Manager Task
   - RC Receiver Task
   - Main Task



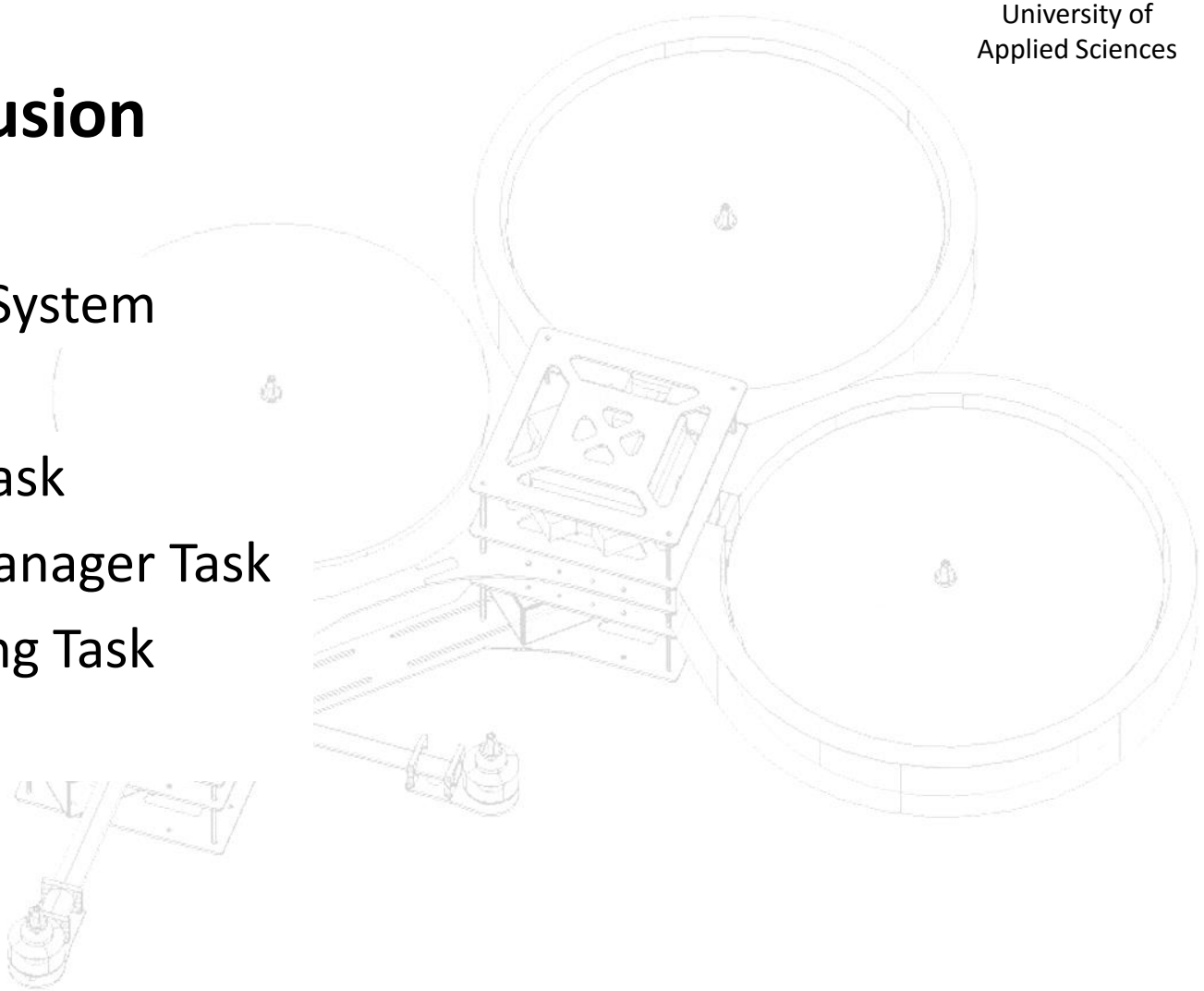
Dataflow


Sensors
raw sensor data
Sensor Data Manager
averaged sensor data
Sensor Data Filter
euler angles
Motors
euler angles
PIDs
motor PWM signals
Remote Control
PID error values
throttle
PID To Motor Mapper

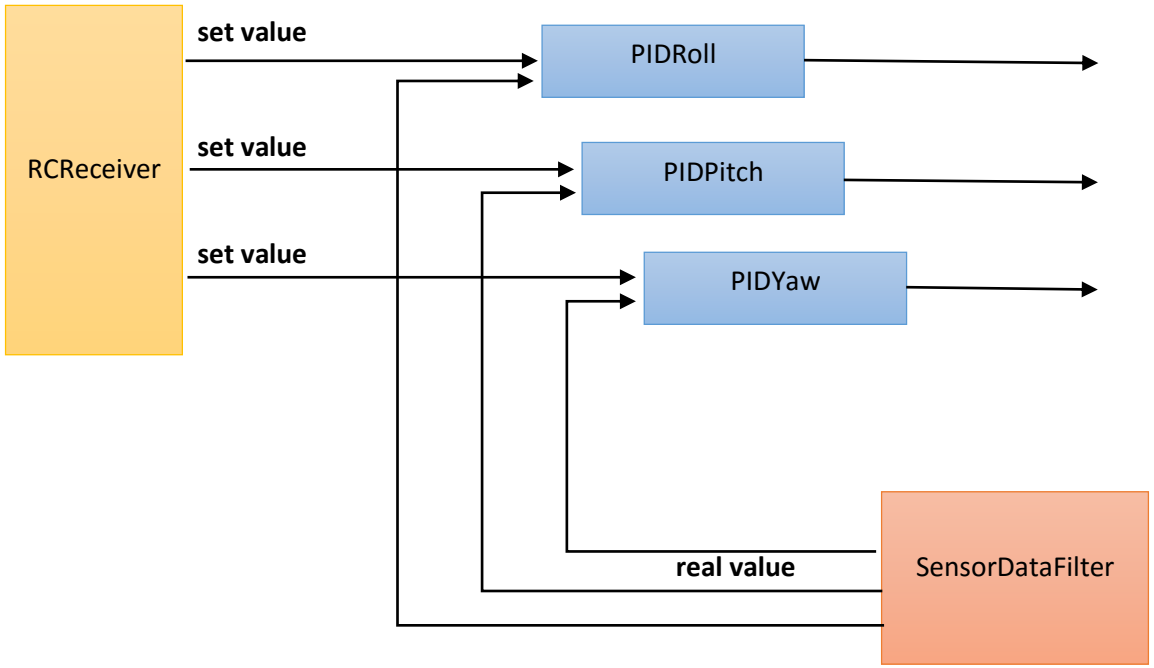
Dataflow conclusion

Four Tasks in the System

- RC controller Task
- Sensor Data Manager Task
- Main Controlling Task
- Logger Task



PID regulators



PIDToMotorMapper

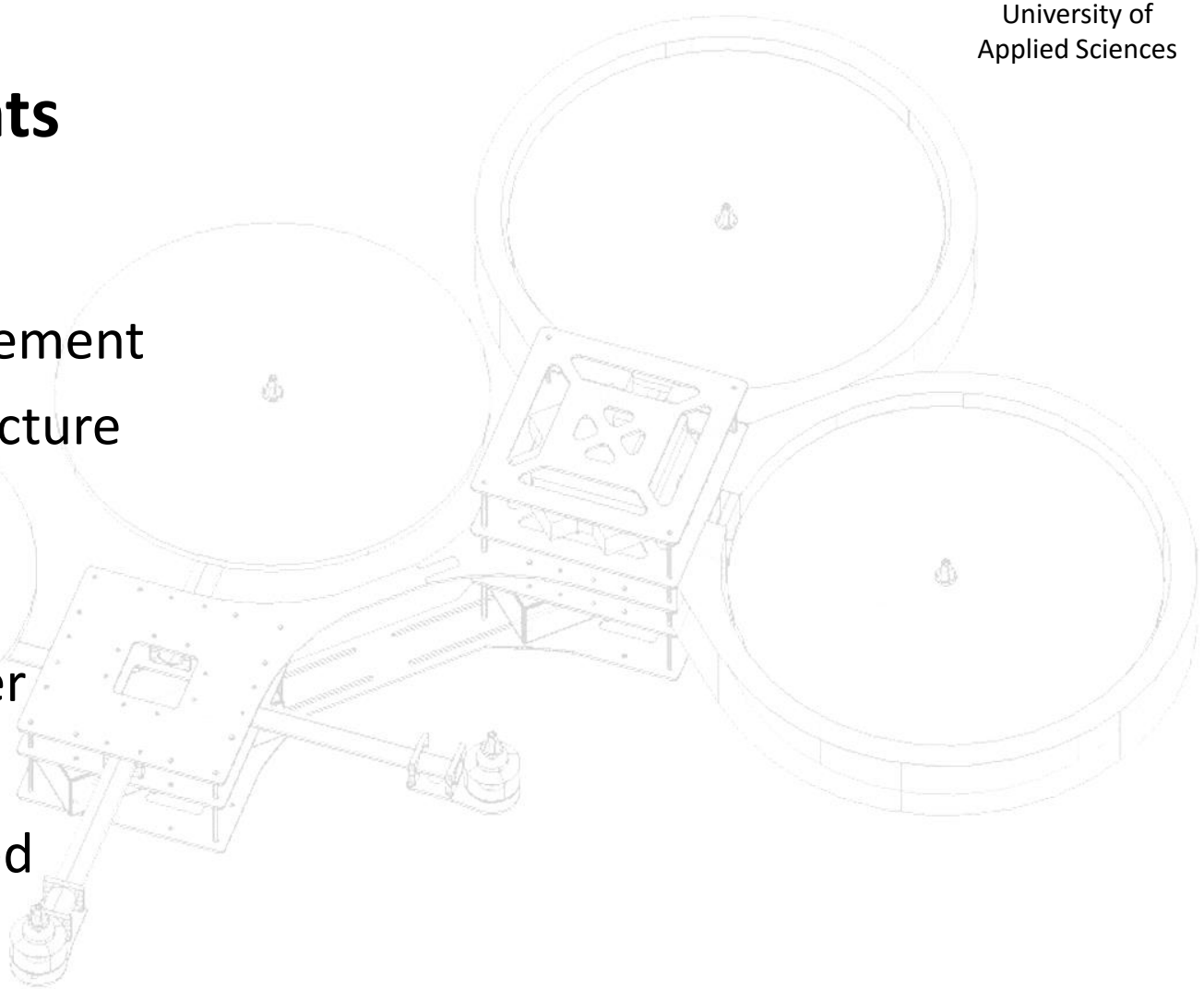
First step:

```
//Mapping table for a QUADX configuration
motorQuadx[0] = PIDMIX(-1,+1,-1, throttle,roll, pitch, yaw); //REAR_R
motorQuadx[1] = PIDMIX(-1,-1,+1, throttle,roll, pitch, yaw); //FRONT_R
motorQuadx[2] = PIDMIX(+1,+1,+1, throttle,roll, pitch, yaw); //REAR_L
motorQuadx[3] = PIDMIX(+1,-1,-1, throttle,roll, pitch, yaw); //FRONT_L
```

Second step:

$$C_{th} * throttle + C_{mix} * PID_{mix} = 100\%$$

Table of Contents



Flight controller

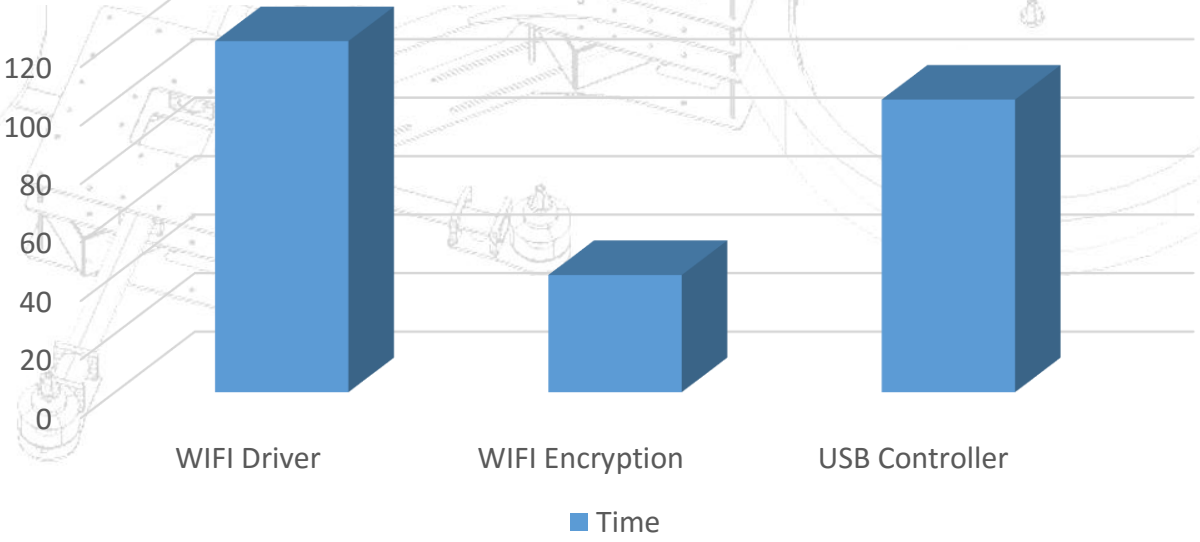
At this point in time the output values of the motor mapper module aren´t correct!

Possible errors:

- Wrong input ranges because of a wrong conversion
- Logical error in the module itself

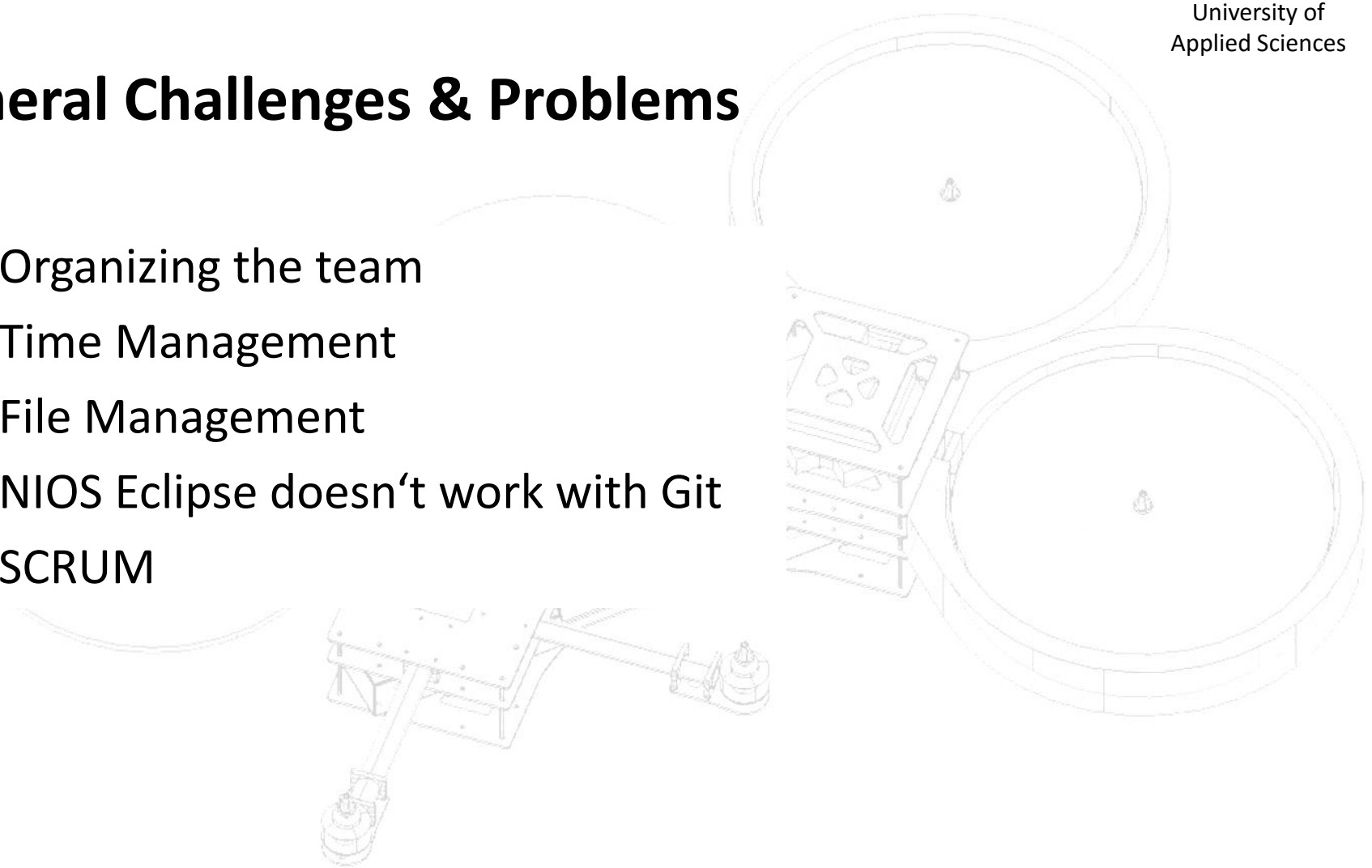
Linux

- Underestimated the complexity
- Implementation of WIFI
- USB Controller
- MAVLINK & MCAPI



General Challenges & Problems

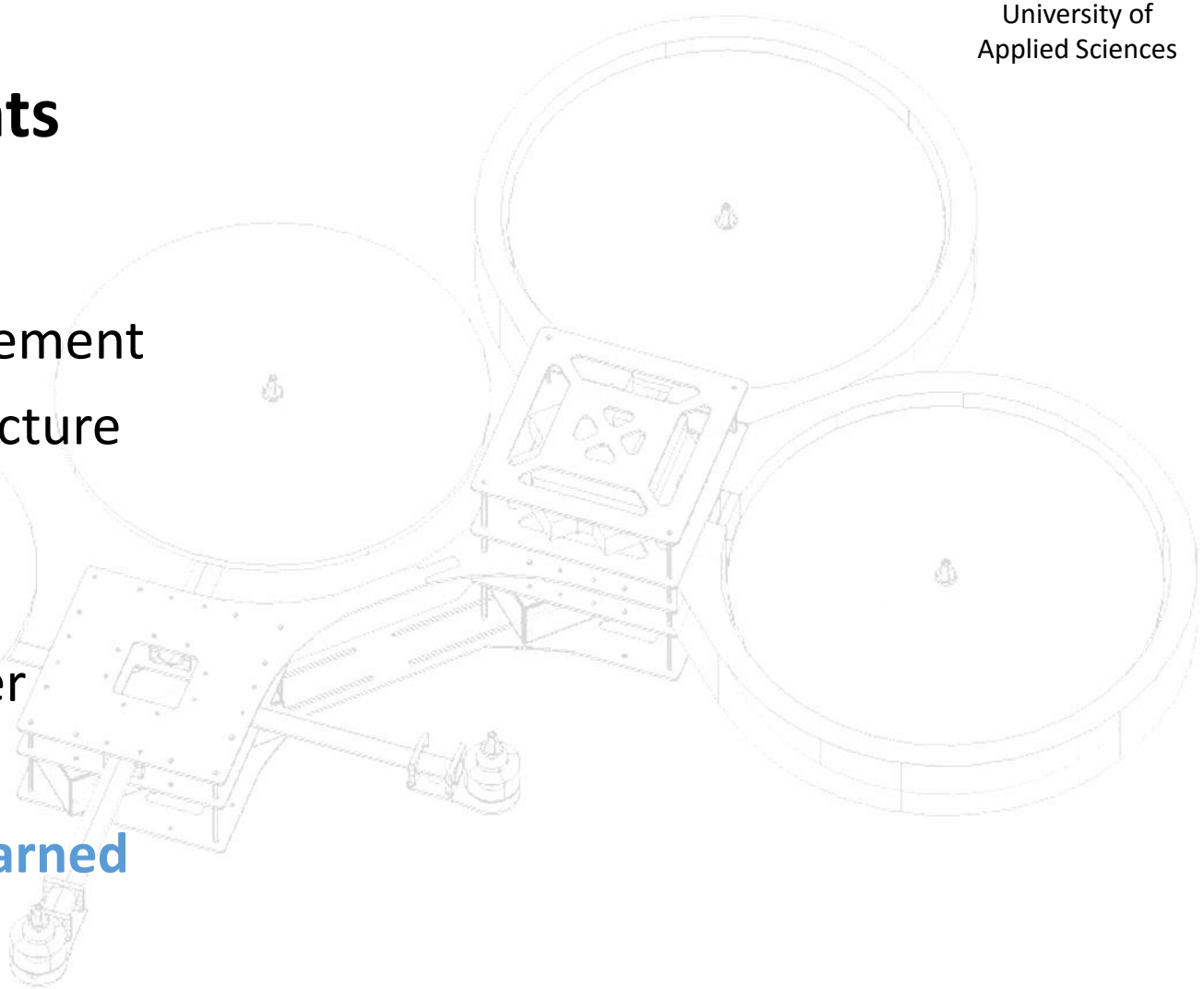
- Organizing the team
- Time Management
- File Management
- NIOS Eclipse doesn‘t work with Git
- SCRUM



Improvements

Initial state	Actual state
Wrong USB Hardware	No USB Controller
Power supply doesn‘t work	Power supply works
No test flight	Successful flight test
-	In principle connection to ground station established
-	Working concept with our own flight controller

Table of Contents



Lesson Learned

- Do not postpone things with unknown risks
- Teamwork
- Regularly team meeting
- Generous time calculations
- Scrum only in fulltime job
- Previous documentation important

Thank you for listening