



# Final Project

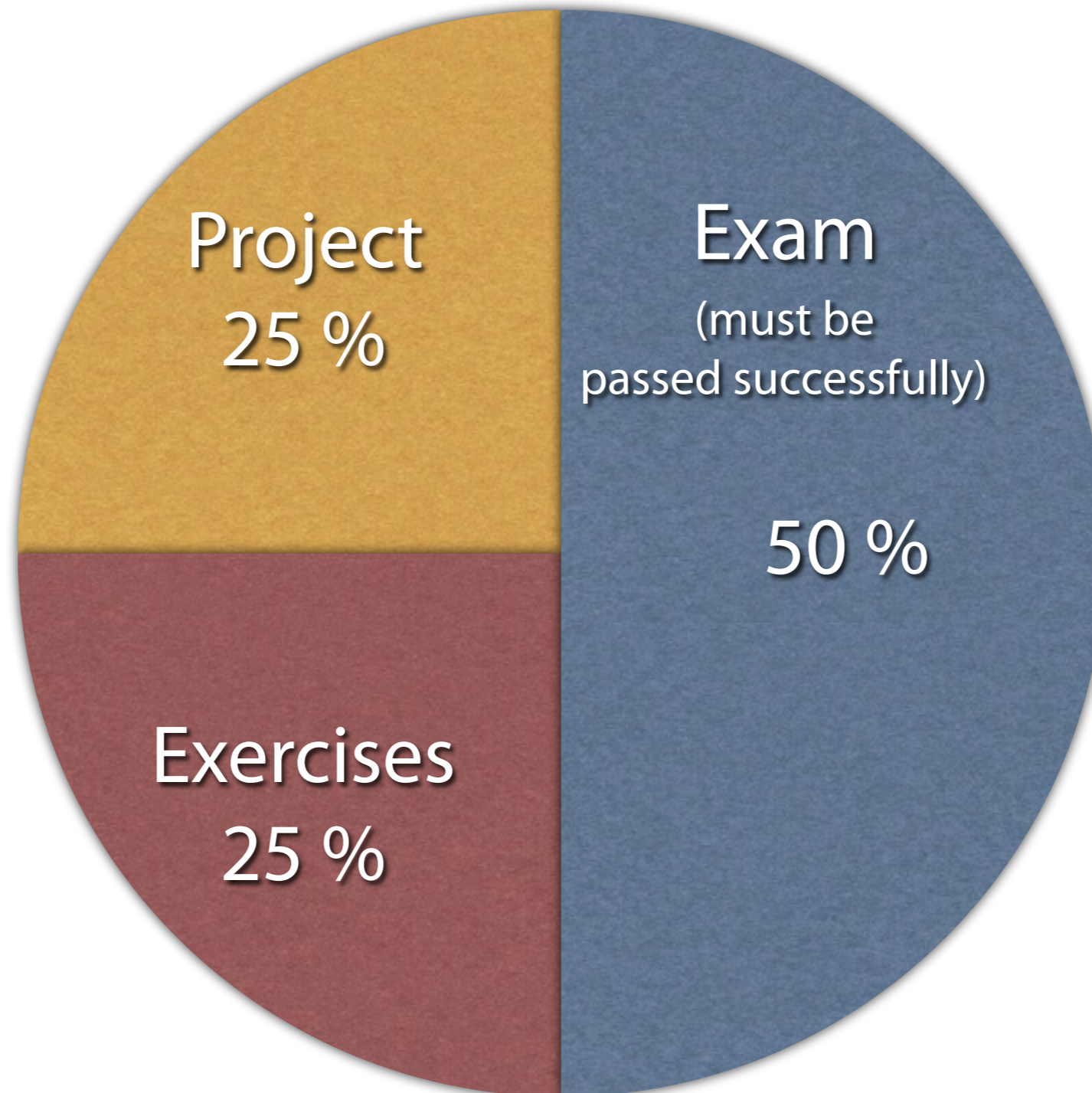
Programming for Engineers  
Winter 2015

Andreas Rau, Saarland University

# The Final Project

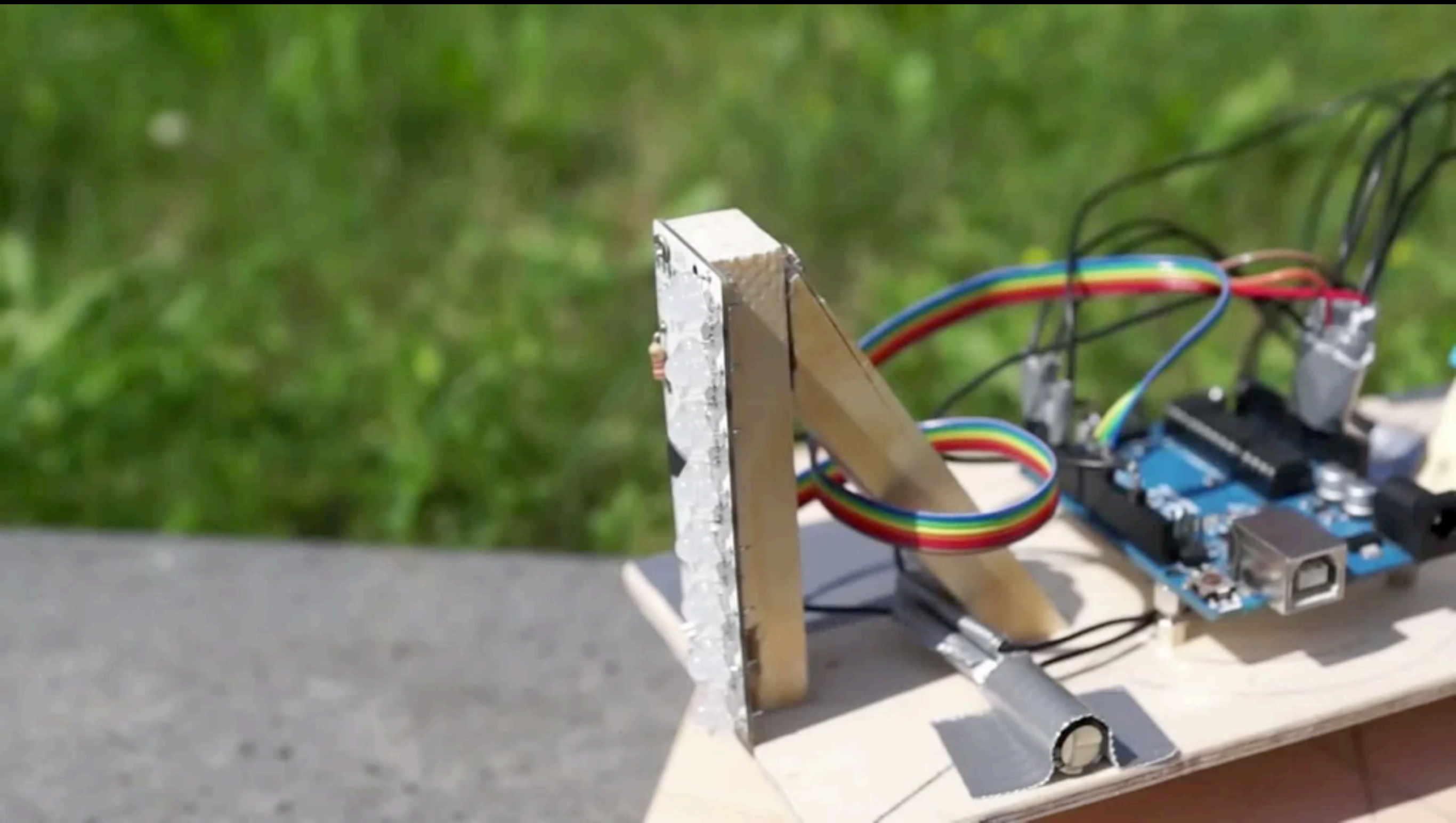
- Groups of 2 Students
- Design & Implement a Complete Arduino Project (Hard- & Software)
- Deadline: Tuesday, March 1st
- Grading based on
  - ★ originality
  - ★ complexity
  - ★ functionality

# Overall Grading



# Licht-Uhr

Murat Güner, Maximilian Junk, Pierre Kehl und Thomas Kreis





# What we expect...

- Proposal (2-3 Pages)
- Overview, Motivation
- Hardware Requirements

## 2 Hardware Requirements/List

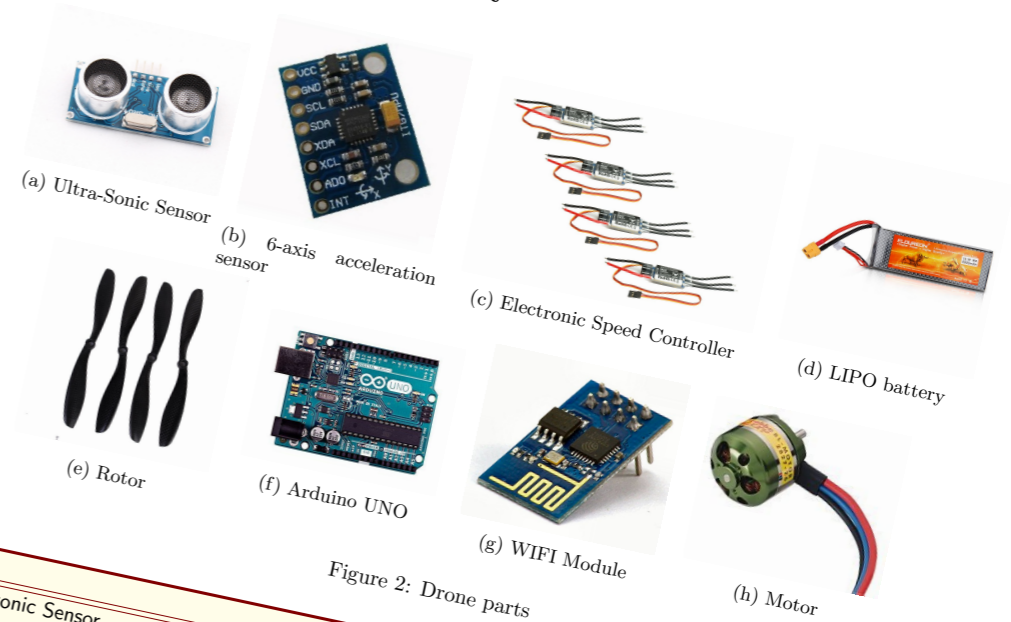


Figure 2: Drone parts

Fig	Item	Description	Price	Pieces
2a	Ultrasonic Sensor	Used for distance measures while indoor flying	0.00	0 <sup>a</sup>
2b	6 Axis acceleration sensor	Automatic level and acceleration measurement	7.73 <sup>b</sup>	1
2c	Electronic Speed Controller (ESC)	Used for distance measures while indoor flying	22.23	4
2d	LIPO battery	Power Supply while flying	34.00	1
...	...	...	...	...
SUM			130.96	

<sup>a</sup>base package - no order required  
<sup>b</sup><http://www.amazon.de/gp/product/B00D1K5Z5Q>

# What we expect...

- Proposal (2-3 Pages)
- Overview, Motivation
- Hardware Requirements
- Use cases

## 3 Contracts

### 3.1 Use case scenario

Drones have multiple purposes and can carry a lot of equipment and sensors on board for a fairly short amount of time. Using it to measure temperature condition in a remote destination is possible without setting up a central weather station. Thus we can overlook a large amount of places ...  
Our drone allows it to enter a set of GPS coordinates and the drone automatically reaches its destination, takes a picture and records the current weather conditions.

### 3.2 Must-Haves

- The drone can fly
- Webinterface allows setting of a course on google maps coordinates
- Photographs are possible and can be transmitted wirelessly to a central storage
- I can hover without changing the position

### 3.3 May-Haves

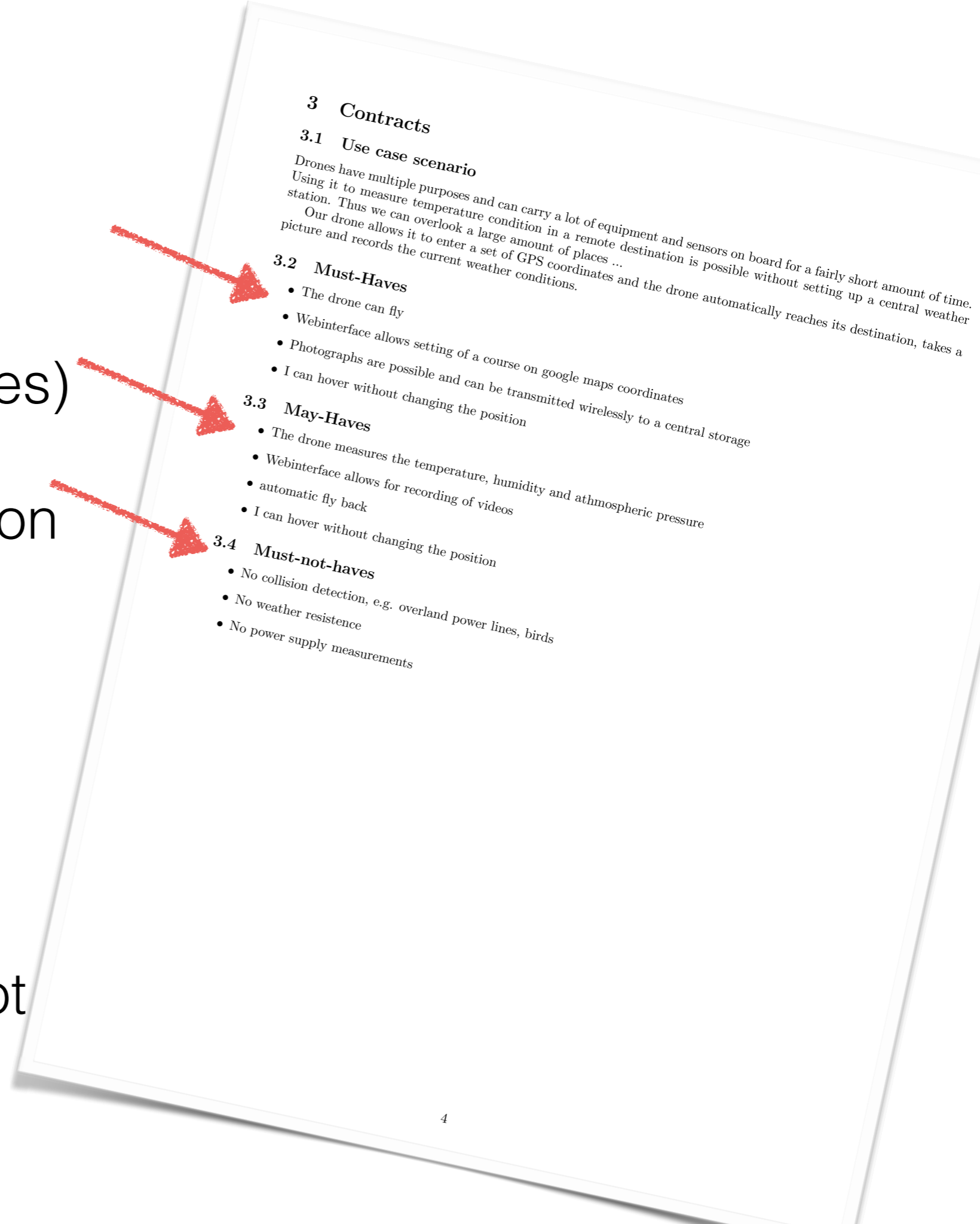
- The drone measures the temperature, humidity and atmospheric pressure
- Webinterface allows for recording of videos
- automatic fly back
- I can hover without changing the position

### 3.4 Must-not-haves

- No collision detection, e.g. overland power lines, birds
- No weather resistance
- No power supply measurements

# What we expect...

- Proposal (2-3 Pages)
- Overview, Motivation
- Hardware Requirements
- Use cases
- Must/May/Must-Not have





# Project

- Originality: ★
- Complexity: ★★★★★
- Functionality:

Checked Against Contracts



# Project

- Originality:



Checked Against Contracts



# What can I use?

- Chair can buy hardware for **10 Euro** per group (no reimbursement)
- all hardware from previous projects
- wood/metal constructions
- be creative



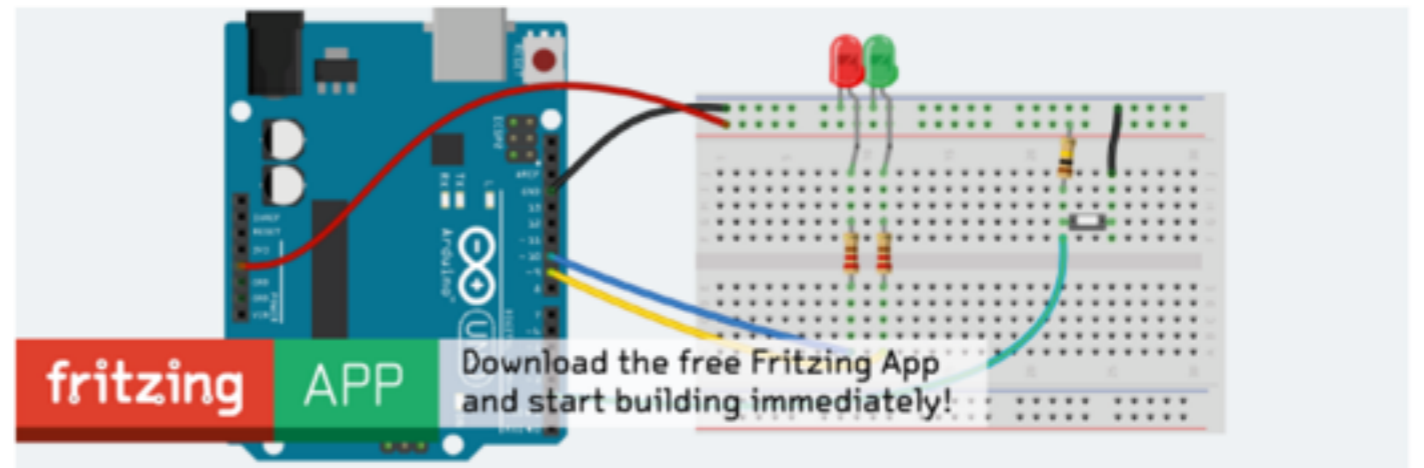
# Delivery

- **Deadline: 1st of March**
- Video
- Group Presentation



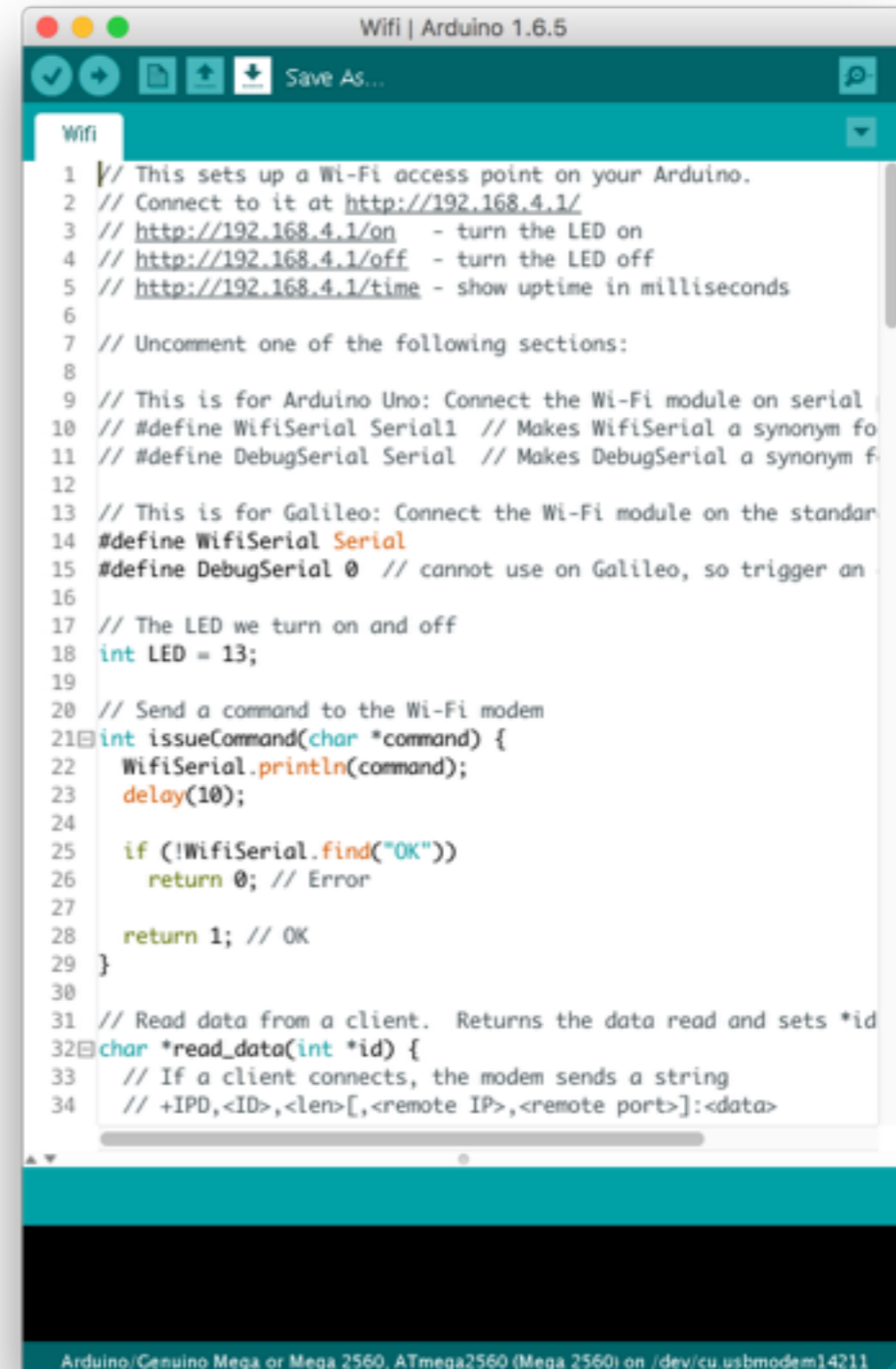
# Delivery

- **Deadline: 1st of March**
- Video
- Group Presentation
- Circuit Diagram/Design



# Delivery

- **Deadline: 1st of March**
- Video
- Group Presentation
- Circuit Diagram/Design
- Well Documented Code
- Final Document



```
Wifi | Arduino 1.6.5
Save As...
Wifi
1 // This sets up a Wi-Fi access point on your Arduino.
2 // Connect to it at http://192.168.4.1/
3 // http://192.168.4.1/on - turn the LED on
4 // http://192.168.4.1/off - turn the LED off
5 // http://192.168.4.1/time - show uptime in milliseconds
6
7 // Uncomment one of the following sections:
8
9 // This is for Arduino Uno: Connect the Wi-Fi module on serial
10 // #define WifiSerial Serial1 // Makes WifiSerial a synonym fo
11 // #define DebugSerial Serial // Makes DebugSerial a synonym f
12
13 // This is for Galileo: Connect the Wi-Fi module on the standar
14 #define WifiSerial Serial
15 #define DebugSerial 0 // cannot use on Galileo, so trigger an
16
17 // The LED we turn on and off
18 int LED = 13;
19
20 // Send a command to the Wi-Fi modem
21 int issueCommand(char *command) {
22     WifiSerial.println(command);
23     delay(10);
24
25     if (!WifiSerial.find("OK"))
26         return 0; // Error
27
28     return 1; // OK
29 }
30
31 // Read data from a client. Returns the data read and sets *id
32 char *read_data(int *id) {
33     // If a client connects, the modem sends a string
34     // +IPD,<ID>,<len>[,<remote IP>,<remote port>]:<data>
```

Arduino/Genuino Mega or Mega 2560. ATmega2560 (Mega 2560) on /dev/cu.usbmodem14211

# Final Document

- Extend Your Proposal
- Discuss Design Decisions
- Occuring Problems
- Present Your Circuit/Code



# Demo

