**To-do list**

* Make a sharelatex shared document
* Make a share folder; Google drive, dropbox etc.
* What kind of OS are we going to use for developing the ground station? (Important for possible GUI development)
  + *My preference goes to Linux (Ubuntu) since it don’t have a windows copy available*

**Architectural design**

Due to our project group only consisting out of three team members the key is to keep things simple. The goal is to reach *Goal 5* as soon as possible, at this point there is room for added complexity.

*[General image of the architectural design]*

**Ground station (PC):**

The ground station (PC) communicates with the drone (ES) via the PC link and is controlled based on the user-input given via the joystick. The state of the ground station equals that of the drone with help of the telemetry data received from the drone. This information will be visualized with help of a possible GUI.

* USB connection with the Joystick
* GUI? *Think we need to discus*

**PC link:**

The ground station and the drone communicate via the PC link, which can be wired or wireless. The priority is the wired connection, when completed the focus will be shifted towards the wireless implementation.

Communication protocol

* UART (Simple, already implemented in the sample code)
* When time schedule allows, possibly implement own protocol

**ES (Drone):**

For the software architecture the choice is made for Round-Robin with interrupts. This because its relatively simple to implement and often appropriate. The addition of interrupts is beneficial since a fixed periodic sensor reading is desired. A RTOS implementation is not an option due to the large added implementation time in combination with the smaller project team.

Main program (Round-Robin)

* Control system + signal processing (Controlling the motor speed)
* Battery sensor reading (Non-critical process, not worthwhile via interrupt)

Interrupt based

* Reading sensor (gyro, accelerometer, barometer)
* Data communication with PC