Project Journal

This log contains a description of our collaboration and the progress of the project.

## 12/05-2017 - work work (report)

Started writing result chapter, based on the test from the 9/05. We discovered that the next test to be carried out should we: measure down to the millimeter instead of centimer. also the test at the bottom of the workspace and the one in the top of the workspace where target and frobit are parallel to the width, should have the same distance in the bottom as in the top, in order to show how much more error margin is obtained as the measurements are moved further away from the center of the camera.

## 10/05-2017 - Filling out report

## 9/05-2017 - Status on Report / Writing reportd1

The frobits fmExecutors has been altered in the waypoint\_list and waypoint\_navigation. We’ve changed it from loading from a file, to writing our own loading in waypoint\_list. It starts by subscribing to a topic, and then we pluck out the coordinates and append them to a list.

The waypoint\_navigation now calls this list whenever we press “a” in the terminal, and starts automode on the frobit-simulations. It then runs the waypoints from the list.

Functions in the waypoints\_navigation has also been altered.

front page - %

About authors - d1

resume - %

Preface - %

Glossary/reading guide - %

Introduction

* Background of the project - d1
* Related work %
* Problem statement - d1
* Hypothesis - d1
* Aim of the project - d1
* (Project structure and approach?)

Materials & Methods

* System overview - d1
* MarkerCapture - d1
* MarkerLocators - d1
* Feature Detection -d1
* Frobit - %
* Intercommunication - d1
* Path planning\_node - d1
  + astar -d1
  + coordtrans - d1
  + Bridge between frobit and system - %

- Some kind of finish for this chapter? Maybe easier to write when everything else looks fine

Process

* SCRUM /Agile
  + why, how etc.
* Examples/graphs so on
* Reflections

Results

* Test etc -> video / pictures
* Precision
* Discussion
* conclusion

Future work - %

Application possibilities - d1

Appendix - %

## 

## 8/05-2017 - Status on Report / Writing reportd1

* Started searching through frobomind in fmExecutors files, trying to modify the loading function from compile time to running time. The auto\_mode function must properly be rewritten as it runs sequentially and its very selective because of all its IF- statements.

front page - %

About authors - d1

resume - %

Preface - %

Glossary/reading guide - %

Introduction

* Background of the project - d1
* Related work %
* Problem statement - d1
* Hypothesis - d1
* Aim of the project - d1
* (Project structure and approach?)

Materials & Methods

* System overview - d1
* MarkerCapture - d1
* MarkerLocators - d1
* Feature Detection -d1
* Frobit - %
* Intercommunication - d1
* Path planning\_node - d1
  + astar -d1
  + coordtrans - d1
  + Bridge between frobit and system - %
* Closure / Overall system %

Process

* SCRUM /Agile
  + why, how etc.
* Examples/graphs so on
* Reflections

Results

* Test etc -> video / pictures
* Precision
* Discussion
* conclusion

Future work - %

Application possibilities - d1

Appendix - %

## 3/05-2017 - Adjusted report structure / filling report

front page

About authors

resume

Preface

Glossary/reading guide

Introduction

* Background of the project
* Related work
* Problem statement
* Hypothesis
* Aim of the project
* (Project structure and approach?)

Materials & Methods

* System overview ---------------------------------> Mark
* Drone/rasp PI --------------------------------------> Thor
* “Computing Device”
  + MarkerLocators
  + Feature Detection
* Frobit
* Intercommunication
* Closure / Overall system

Process

* SCRUM /Agile
  + why, how etc.
* Examples/graphs so on
* Reflections

Results

* Test etc -> video / pictures
* Precision
* Discussion
* conclusion

Future work

Appendix

## 1/05-2017 - Report structure outline

front page

About authors

resume

Preface

Glossary/reading guide

Introduction

* Background of the project
* Related work
* Problem statement
* Hypothesis
* Aim of the project
* (Project structure and approach?)

Materials & Methods

* System overview ---------------------------------> Mark
* Drone/rasp PI --------------------------------------> Thor
* “Computing Device”
* Frobit
* Intercommunication
* Closure / Overall system

Process

* SCRUM /Agile
  + why, how etc.
* Examples/graphs so on
* Reflections

Results

* Test etc -> video / pictures
* Precision
* Discussion
* conclusion

Appendix

## 28/04-2017 - Sprint 5 review and planning sprint 6.

Downsizing our research and start to focus on writing report. at least 1-2 pages a day, minus the corrections afterwards.

Some research are allowed as long as it doesn't interfere with the report writing.

Some guy, Mathias is located in Drone center and has a ROS-node that is capable to push waypoints into the frobit in runtime .

## 26/04-2017 - Different heights for drone.

For now we work with the assumptions that the the corners, that is used for cropping the send picture is known. This is to get the factor for scaling the picture from pixel to meter. Factor = Pixel/meters.

Test course setup. width max 243 cm, height max 277cm, distance from frobit to target 170 cm

Conversion between pixels and meters is done in the astarPP.cpp file, it’s done by knowing the width of the workspace in meters, and dividing by width of the workspace in pixels. The conversion is done as the last step in the node. Some testing has been done today, where the values are not precise but close, we think that in order to make it precise, we need to do some extensive test where the workspace is set up very accurately.

## 25/04-2017 - Raspberry PI on ProjektNet - robolab

if you cant find the raspberry, write sudo arp-scan -l interface=’wifi-controller name’ and pick one of the ip’s.

now Raspberry pi 10.126.128.46

## 12/04-2017 - Raspberry PI on network

because of bad configured DHCP’s we need to set the ROS\_HOSTNAME, ROS\_MASTER\_URI and ROS\_IP on every machine we want online.

<http://wiki.ros.org/ROS/NetworkSetup>

[http://wiki.ros.org/Robots/TurtleBot/Network%20Setup](http://wiki.ros.org/Robots/TurtleBot/Network Setup)

These two guides describes how to set up a connection. It’s very important that they all point to the same master. The ROS\_IP is machines own IP. and host name is own host\_name. all needs to be written into ~/.bashrc.

FIREWALL AND OTHER BLOCKING SOFTWARE MUST BE TURNED OFF - or VPN for the machines can be an alternative.

here are two commands for quick publishing / debugging:

machine 1 : rostopic pub -r10 /hello std\_msgs/String "hello"

machine 2: rostopic echo /hello

There is a mount point for any usb at /media/usb.

## 11/04-2017 - Transformations matrix & distributed ROS

By adding the angles of the down left markerlocator & the frobit to our custom message, mapping between the Frobits local coordinate system and the image should be easy with a transformation matrix.

to do:

* Strengthen the feature\_detection node.
* Reinstall the PI with Ubuntu mate.
* Have the PI run the markerlocator.
* Connect the markerlocater with feature\_detection node through ProjektNet.

Small changes to the feature node and the custom message. it now sends theta and does additional checking of flags AND quality from the MarkerCapture.

The PI was reinstalled due to kernel failure.

The PI runs any node given, but does NOT communicate with at PC over ProjektNet. Might be cause bad .conf file or settings made by users.

Planning to take a step back and practice communicating over network with 2 laptops thus eliminating failure from the PI.

Made a coordTrans class for the path planning\_node. This class handles the transformation from a coordinate in the picture to a coordinate in the Frobits relative coordinate system. The class is not fully utilized in the path planner node yet, but it’s primary method has been tested, and seems fairly accurate, although it was a quick test. For now the path planner node only runs once, then terminates, next step is to make it run periodically, maybe implement it like a state machine, but will first be clear after more R&D.

The launch file for run\_frobit\_waypoints (or something like that) Has been modified to have the actual diameter of the Frobit pro and not the frobit. This has made the Frobit able to run a lot better as it’s supposed to according to the waypoints, some adjustment with the ticks per (revolution?) may have to be done aswell tho.

## 10/04-2017 - Feature\_detection & Path planning nodes

defined different callback functions in the feature\_detection node depending on which markerpose we’re looking for. The ros library TimeSynchronizerand approximatetimeSyncronizer would not work with python so a custom method for making sure all the correct markerpose’ are being processed in the wanted order. The link is still weak but functional. The method consist of different global flags that are set True and False when there ready for being processed.

4 markerlocators are used to pinpoint the corners. The MarkerLocators publishes the coordinates as float, but since where using them for cropping the image the floats are rounded at converted to ints. some precession may be lost in this processes. The markerlocators are 26 pixels wide in the old calibrating launch file.

To get a full picture and account for the target being in line with 2 markerlocators 13 pixels are added to the highest crop line, and 13 pixels are subtracted from the lower crop line.

picture below describes how the nodes are implemented.

## 

The picture doesn’t show that feature\_detection also publishes 2 messages to the path planner simultaneously, but with different topics.

One publishes a Grey-scale image containing only a cropped image from the original made up of 0’s and 7’s for blank space or occupied space.

The second topic is a custom message which contains the x - and y-coordinate for the frobit and the goal/target.

Path planner node subscribes on two topics which feature detect publishes:   
 /output/image\_treated, which contains a gray scale picture, suited for the A\* algorithm.

/output/target\_frobit, which contains x and y coordinates for the frobit and the target.

The picture is used as a workspace, and the coordinates as a start and end coordinate for the path planner. The path is returned with all the pixels making up the path, or a set of waypoints (x, y coordinates) where the frobit has to move before changing direction.

## 05/04-2017 - CMakeList.txt

Worked with linking the different libraries and making the ROSnode compile with C++.

We had a trouble with having multiple header and source files that need to be compiled correctly with the catkin\_make. It need the path to every file, not just the folder.

Added custom made c++ classes in a ROS project by first adding the .cpp files as a library:



And the library is added under target\_link\_libraries.

The .h files needs to be located under project/include and under include\_directories include needs to be aswell.

## 03/04-2017 - Work day

Launch file for markerlocator with 6 markers created, 4 for coordinate system, 1 for the Frobit and 1 for the target position. A new ros package was created, in order to make feature detection subscribe on the markerlocator.

Still trying to figure out how to manipulate individual pixels in a picture in C++ with openCV.

The feature\_detection ROS node subscribes on the MarkerLocater markerpose on every corner.

the same ROS node needs to subscribe on a markerlocater markerpose pos. from the robot and goal. The plan is to send it as one big package to the path planner.

Feature\_detection also checks weather the quality is “high” enough to accept the picture from the capture node. It then crops the picture accordingly. all this to minimize the calculations needed in path planner, and to de-stress the drones by having all calculations on a laptop/computer between the drone and the frobit.

## 31/03-2017 - Workday

For now we will ONLY work on having the system runs once! Then we will add on to the functions.

Printed markerLocators for workspace corners, frobit & target position.

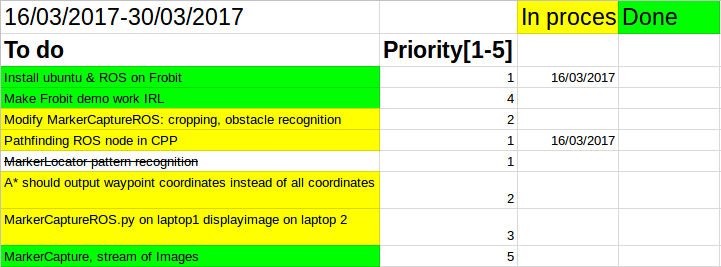
For this sprint we will only implement a simple A\*, where configuration space and roadmaps will be ignored, but possibly implemented later.

We agreed that on this sprint we will work towards testing in the end of the sprint.

## 31/03-2017 - Sprint 4 review meeting

The agenda for the meeting

* Sprint 4 review
* - We decided to transfer the jobs from Sprint 4 to Sprint 5, that wasn’t completely done.



* A\* output
* The output from the A\* will go be saved into the Waypoint mission in the frobit, in order to test the path made from drone above.
* PI cam ?
* For now, the camera on the ceiling in robolab will be our reference point. Also we will work with a fixed height for now, in order to keep reinforcing our weak points and nodes. When this is done, a dynamic function will be created to calculate height later, or at distance measuring tool will be added to the drone.
* Sprint 5 goals
* The goals from Sprint 4, which have not been completed.
* As extra, the weak links in our nodes system needs to be reinforced/finished so its possible, to have a fully functioning chain , from taking af picture, to making a path planning. When this is made, we have a project where we can add further complexity to.

## 30/03-2017 - Work

Plan for today:

* Make the frobit move, so we can identify eventual problems.
* Finish the c++ subscribe node
* Talk to Henrik Midtiby about a way to map different heights over in pixels.

Raspberry pi 3 ip - ProjektNet

10.126.128.35

Frobit - ProjektNet

10.126.128.27

The frobit.rules is placed in /etc/udev on the frobit, and its now possible to send commands from the computer, by ssh into the robot. It’s now controllable by computer with the bin/frobit\_run.

We also know how to but in coordinates for the autonomous mission in the frobit. we need to have the a\* pump out coordinates and save those into the mission planer to have a system that runs ones. The plan is to have the system be dynamic, but for starters, we just want, to run the path ones.

MarkerLocater has almost been looked through, and we are almost sure how to proceed now. We want to have all other parameters estimated by knowing the length between two points in our field.

Testing in robo-lab is about to start.

astarPP ros project running in c++ created, after problems with including opencv in CMakselists.txt and package.xml.

CV\_bridge has been implemented and the node is able to subscribe to published sensor\_msgs::image, and then converting them into opencv image format. Next is figuring out a way to send cv images for the path planning. algorithm, so we can work directly on the data the cv::image points to.

## 28/03-2017 - Normal work

still trying to install everything on frobit, and ssh into it. trying to make an scripts that automatically connects to projektNet, so its possible to turn on the frobit and instantly ssh into it.

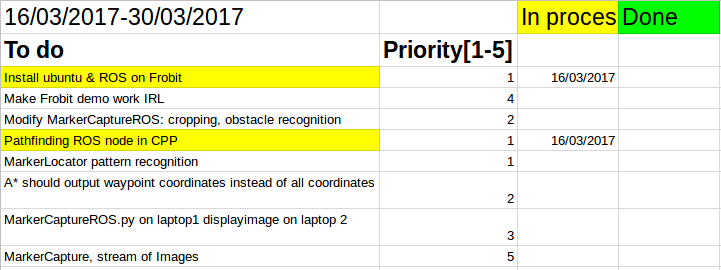
Same idea applies for the drone / rasb PI.

MarkerCapture, and MarkerLocater is still under investigation. Needs to ask Henrik Midtiby about a way to map from points to pixel with various’ heights.

Sprint ends Friday. - so far, not much development. some due to work in brobygning.

## 16/03-2017 - Start of sprint 4

A new sprint has been planned:



## 15/03-2017 - MarkerCapture

We decided to **change from text format to the Image format** in every ROS node, because Image is a type that ROS supports, and it’s fairly easy to implement rather than saving to txt files and opening said files. We suspects it is also cheaper computation wise.

We Made a ROS node used for debugging. The ROS node subscribes to a topic /display\_image, and displays whatever is on that topic.

We decided to write the subscribing node for the path planner in C++ and everyone else in python, because of the easy nature of python.

MarkerCapture works, but needs alteration to work with our project.

It’s possible to simulate our path using the autonomous mode in Yields Frobit\_demo , by writing coordinates in a text file called waypoints.txt in /frobit\_demo/waypoints/.

## 10/03-2017 - Sprint 2 - review meeting with Kjeld

* Presented work done under sprint 2.
* Agreed to switch from text file to matrix format in ROS nodes.
* Marker-locater ROSnode from frobomind for targeting the robot.
* Dynamically send pictures from camera, start working with video feed.
* Attend lecture on gazebo - the visualization software for robots/drones/environment.
* Visit mathias hoejgaard to talk about the marker-locater node.
* Frobit handed out. - needs Wifi dongle for communicating with our computer.
* **System overview changed from ‘Drone-Frobit’ to ‘Drone-Laptop-Frobit’**
* Prepare for next sprint meeting on 17/03-2017.

## 09/03-2017 - Documentating & Draw route

A ROS node to draw the route the frobit should take, has been established. The route is based on the A\*.

The hypothesis from the introduction of the report has been refined. The introduction of the report has been added on shareLatex.

## 07/03-2017 - Documentating

We wrote a couple of pages about the A\* algorithm and how the computer vision worked.

Flowcharts to explain the nature of our software has also been started.

## 6/03-2017 - ROS nodes - Version 1

The first primitive ROS nodes are done. They can each be run on the same computer and send the .txt file. and then the image is processed, saved to txt and send to the subscriber.

When the subscribing node gets any .txt file it immediately runs A\*.

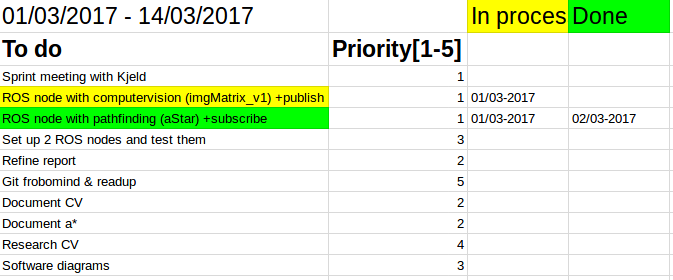
## 2/03-2017 - Linking computervision and path finding

A branch on github named ROSnodes was created. It has two .py nodes, one for subscribing and one for publishing.

* The publishing node takes a .txt file and publishes in a .msg using string.
* The subscribing nodes takes the string and puts it into a .txt file
* The subscribing node then calls demo.exe(A\* algorithm), which produces a new .txt file containing the shortest path between two points in the .txt file. For now the start and ending points are hardcoded.

Started on incorporating the computer vision part, into the publishing node.

## 1/03-2017 - Linking computervision and path finding

Planned 3rd sprint:  


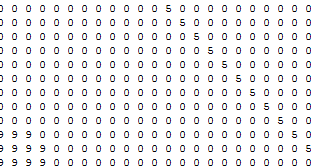
Created a document on Google docs called meeting agendas, which contains subjects we’ll like to discuss at our meetings.

Tried to publish/subscribe a compressed image, to no avail.

Looked into the ROS structure of frobomind and ran the frobit\_demo

## 27/02-2017 - Linking computervision and path finding

* Computerization auto crops after the green colors found in corners of workspace.
* Its is now possible to all the red obstacles in the map, even if they lie in the edges of the map.
* The image is saved as a greyscale - with obstacles (white = 255) and free space (black = 0) in a seperate picture.
* The grey scaled image of free space and obstacles is mapped into a matrix structure, and saved as an .txt file to make the implementation easier in C++.
* The obstacles are re-”scaled” from 255 to 9 , to make it easier to retrieve the digit from the string-data structure.



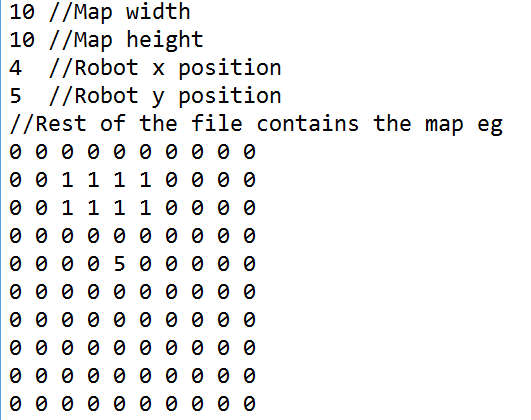
In the lower left corner of the picture a corner of an obstacle can be seen represented with nine, and the shortest path can be seen as diagonal with five in the upper right corner.

A mapHandler class has been implemented in cpp, with the purpose of writing and reading from .txt files, and passing a pointer to the map to A\* algorithm.

## 27/02-2017 - Handling the map as a .txt file

The map of the workspace will be saved in a .txt file, by making a general structure of the .txt file we can minimize overhead to the most necessary data, with the map in a .txt file specific data about the map can be read, without working with the whole map (eg. robots position).

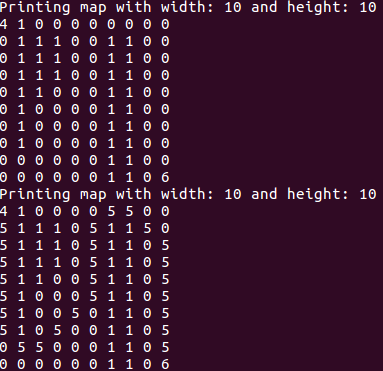
The general structure of the .txt file is as follows:



## 24/02-2017 - basic A\* algorithm

A\* algorithm is now functional with vector<vector<int>> as input along with a starting position and destination position.  
The following picture illustrates how A\* will find the shortest path and stay clear of obstacles. Meaning of the values are as follows:

|  |  |
| --- | --- |
| 0 | Free space |
| 1 | Space occupied by obstacle |
| 4 | Starting position |
| 5 | Path taken |
| 6 | Destination position |



First map is a printout of the initial map with start and end positions, where the shortest path is marked in the last printout.

The .exe file was created with: g++ -o a.exe -std=c++11 source.cpp astar.h astar.cpp

The files has been added to the git branch path planning.

## 21/02-2017 - More on computervision

when one opens a picture, in grey scale, the lowest value is 0 and coresponds to BLACK - 255 corresponds to WHITE.

## 15/02-2017 - Thors entry on Computervision

through OpenCV we can load an picture, extract the colors with HSV “protocols. Then we can extract whatever colors we want by specifying what Hue, Saturation and Value. We need upper, and lower boundaries. HSV is made of 3 array, length 0-179,0-255,0-255. The values are “computed” from the degrees hue(360 degree) to a the array, by dividing the decided color value by 2.

Red can now be located on our picture. Any color should now be possible to locate.

## 10/02-2017 - Starting on second sprint(computer vision and path library)

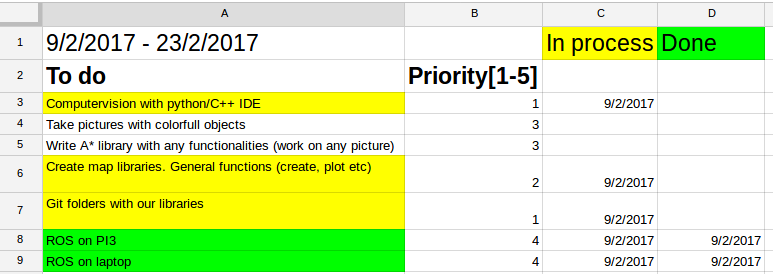
* Making a python script that handles computervision, finds different colors on the floor.
  + Starting working with HSV instead of RGB because of simplicity

explains the pro’s of having HSV instead of RGB.

<https://www.kirupa.com/design/little_about_color_hsv_rgb.htm> - 10/2

* Started to implement aStar path finding in a ros node
* We decided that the workspace map, should be kept as vector<vector<int>>, where obstacles will be marked with a integer value of 2, freespace with a value of 0 and the robots position with a value of 1.

## 9/02-2017 - Forming of second sprint

* Implementation of ROS nodes using opencv has been started, using the following guide: <http://wiki.ros.org/image_transport/Tutorials/PublishingImages>
* Meeting with Kjeld about first sprint, following points were agreed upon:
* Improve academic writing style: Use references and more quantities. Only include references to work which has been referenced, keep it short and concise.
* Identify major problems/time consumers in the project, and start with those.
* Prepare agenda for Kjeld in this document before each meeting and also update this document with answers.
* Start computer vision and map building algorithm, write it as a python library where functionality will be added as the project processes.
* Picture of second sprint:  
  

## 8/02-2017 - Preliminary planning

Milestones excel has been updated appropriately with 14 days iterations(sprints)

Four milestones was planned, and approx. 4 major task was put in the Product Backlog(PB). Each task will probably be sliced into even bigger task’s when the overhead layer/direction i confirmed with Kjeld.

The product of the bachelor has been loss-ly defined due to the shear size of the project. Some aspects as the decomposition of a crop-field for instance, has been lead out of the PB until the projects timeline and magnitude has been better established.

*“Our map does not update unless all four corners of the workspace is seen by the drone/camera. everything else is discarded.”* - Decisions-making for when the camera is mounted on the drone.

## 14/01-2017 - End of sprint 1

Kept working on the application for Kjeld.

Re planned the project and the boundaries for the project. manual drone steering.

Worked extensively in making ad hoc connection between rasp PI and PC, to run ROS nodes in network. abandoned due to poor results, and time consumering and will use a dedicated router, to route packages between the 2 systems.

**Questions for Kjeld:**

* Should we start by having a static video feed from a camera in the ceiling, and then move on the mounting the camera on the drone?
* Should we have altitude sensors on the drone? Or should we just use computer vision to estimate the drones altitude?

Project description was updated with hypothesis and aim of the project.

## 09/01-2017 - Meeting with Kjeld

The drone should not be autonomous, but still using ROS on a raspberry PI3.

Find related work: Google scholar, field robotic event.

Hypotheses: Find some measurable values we want to strive after, maybe best object avoidance, fast path finding eg.

Do research about relevant equipment (although we may end using what the university can provide).

## 29/12-2016 - First Sprint

SCRUM method was implemented.

Sprint backlog and product backlog, both i Google sheets. Also a burndown chart has been made. First sprint initiated

Bachelor project application to kjeld jensen was started.

Section with application of the idea implemented in the report.

We talked strategies for the drone, and the idea was put on paper. ROS and FroboMind tutorials is an ongoing process.