Base script for mid review presentation

* what we have done so far? (status)
  + Project stage and progress
* compare to the plan (did we behind schedule? Yes, but if we finish this then we nearly achieved the scope)

1. Introduce Drone version 1 (matek405mini)
   1. Reasons (keep it short)
2. Challenges working with that setup (short)
   1. Difficulties customizing the platform
3. Introduce Drone version 2 (don’t mention hardware – just mention goals)
4. Script: Going into our second drone with what we learned from our first one. We decided it’s best to look for a fly controller that supports further customization and better automation capability to help us achieve our goal of an automation drone and avoid previous problems

b. Content:Main goals of the second drone (**discuss later and pick or add**)

1. Better automation support
2. OS capability
3. Support sufficient number of sensors for automation
4. Automation speed
5. Alternatives (short comparison table)
   1. Pixhawk
   2. Rpi 3B + Navio 2
   3. Arduino
6. Decision made and pictures?
7. Hardware team what did you do?
   1. Hoan
   2. Thịnh
   3. Qưng
8. Current challenges:

Drone setup

Currently, we have a problem when configuring the settings of Pre-arm Safety Checks in Mission Planner (a ground control station for Plane, Copter and Rover). Some parameters, stored in ardupilot- raspberry, are not suitable when working with indoor-environment.

This leads to the failsafe triggers, and the drone does not arm even if we change the manual flight modes that do not require GPS for the purpose of indoor flight.

We are still working to solve this problem by searching this error, trying to adjust the value of parameters relating to Pre-arm Safety Checks. As a result, the Pre-arm Safety Checks can match the right conditions and the drone can arm.

1. Automation progress
   1. Chosen platform (reasons) – Ardupilot + dronekit
      1. Ardupilot

Ardupilot is open source autopilot software. Suitable for our indoor copter since it is the only autopilot software capable of controlling almost any vehicle system imaginable. It has advanced data-logging, analysis and simulation tools.

* + 1. Dronekit

DroneKit-Python (formerly DroneAPI-Python) contains the python language implementation of DroneKit an API allows developers to create Python apps that communicate with vehicles over MAVLink. A best solution to our indoor copter that uses onboard companion computers with the addition of open source library.

* 1. Connection script, simulation, basic commands (simulated)

We’ve created script using Dronekit API to do basic command from communicating with drone to simple go-to. Most of the scripts are test in a simulated environment and waiting to be prompted into hardware.

1. Software team what did you do?
   1. Í

Research compatible platform (software)

Configuration flashing port to connect the board

Drone component research

Research on drone frame and model

Fixing Transmitter can’t activate armed mode to control

Establish component into drone frame

Set up simulated environment

Parameter configuration

Basic drone movement script (connection)

Connecting APM planner to script to simulate script

Establish connection between companion computer to raspberry pi

* 1. Q

Research compatible platform (software)

Configuration detecting plugin fly control board for betaflight

Drone component research

Research on drone frame and model

Buying drone component

Config Matek f405mini firmware from betaflight

Basic drone movement script (take-off, landing, configuration, velocity setting)

Simple go-to script (Flight forward)

nonGPS navigating solutions (Manually import flight path)

Guided flight script (drone take-off, move forward and landing without using GPS)

* 1. Qưng

Test run simulator

Start SITL (Software in the Loop) drone

1. Current challenges:
   1. Ask í or Q (software)

Connect code to hardware

Guided flight code still has some problem

Integrate nonGPS to guide drone flight direction

Implement script onto drone

1. Plans:
   1. Solve hardware problems
   2. Establish connection between companion computer with copter
   3. Apply mission plan into copter
   4. Optical-flow research to increase safe flying

Task list:

Drone Version 1:

<http://www.mateksys.com/?portfolio=f405-mini#tab-id-3>

Research frames and drone schematics

Research components (propeller, ESC, FC, PDB, motor, battery)

Assemble:

* Attach propeller with motors
* Attach components to the frame

Solder drone parts:

* Wield wires to FC
* Wield ESC FC

Install firmware to drone controller

Install Flight controller software (Beta-flight) to laptop

Connect flight controller to laptop and configure drone settings

Adjust radio control signals

Calibrate ESCs and propellers

Test motors and RX signals (receiver)

Test indoor flight

Drone version 2:

Research compatible drone-hardware

Find suitable ground control

Purchase drone components

Assemble drone parts (flight controller, wires, ESC, ...)

Solder RPi combo with ESC to test motor

Install raspbian and ardupilot firmware to Rpi

Configure ardupilot in Rpi

SSH to Rpi via wifi

Install mission planner (ground control station) on laptop and config/connect to drone

Test ground connection with hardware

Calibrate ESC and compass of drone with mission planner

Adjust parameters of ardupilot in Rpi to let the drone arm