1. Outdoor experiments plans (You should wear Helmet)

[For one drone only]

    a. Definition of the channels
        i. Channel 6 will be a master channel (default modes fixed using Qgroundcontrol)-
            1. Loiter (950 - 1150)
            2. Stabilize (1900 - 2020)
        ii. Channel 5 will be a secondary channel for custom scripts
            1. Desired velocity commands in terms (950 - 1150)
                a. x_des_dot
                b. y_des_dot
                c. z_des_dot
                d. ps_des_dot
            2. Desired attitude commands and yaw rates (1400 - 1600)
                a. ph_des
                b. th_des
                c. ps_des_dot
                d. z_des_dot
    b. Initial drone testing
        i. Flying in loiter mode -> Check how the performance is by applying the external perturbation
        ii. Fly in stabilize mode -> Check how the performance is by applying the commands from the radio controller
        iii. Change the modes using the radio controller, i.e., loiter and stabilize
    c. Auto tuning
        i. Perform auto tuning
            1. Watch the entire video first
    d. Verifying the drone's state by plotting in Gazebo in outdoor settings
        i. Plots the translational data: x, y, z, x_dot, y_dot, z_dot
        ii. Plots the attitude data: ph, th, ps, ph_dot, th_dot, ps_dot
        iii. Plots the CAM device data: ph_p, th_p
    e. Verifying the drone's state by plotting in rqt_plot in outdoor settings
        i. Plots the translational data: x, y, z, x_dot, y_dot, z_dot
        ii. Plots the attitude data: ph, th, ps, ph_dot, th_dot, ps_dot
        iii. Plots the CAM device data: ph_p, th_p
    f. Task - 0: [done] Interpreting the ROS commands to drone
        i. First, check what the higher level commands sent from the MAVROS to the Pixhawk
            1. [done] Attitude commands [/set_point/attitude]
            2. Position commands [/set_point/attitude]
                a. Simple map Joystick to position
                    i. Start with [-1 1]
                b. Autonomous mode
                    i. Do hard code in python scripts
                        1. Task - 1

                a. Simple take off at 3m and land
            2. Task - 2 []
                a. Autonomous take off
                b. Follow a rectangular path of 3x3x3m
                c. Autonomous land

g. Task - 0.1
    i. Make a off_board_code on ground computer and verify the performance
h. Task - 0.2
    i. Integrate the WIFI booster
    ii. Hardcore the IP address of the Drone's Rpi

[Two drones]

i. Task - 0.3 [Flying two drones]
    i. Make entire system ready
    ii. See the quadcopters in Gazebo
    iii. Make a off_board_code on ground computer
        1. Task -1
            a. Simple take off at 3m and land
        2. Task -2
            a. Autonomous take off
            b. Follow a rectangular path of 3x3x3m
            c. Autonomous land
            d. Here, both the drones will fly from their local coordinates
        3. Task -3
            a. Check the global position of the drones
                i. Make one of the drone's initial position as the world coordinates frame
            b. Autonomous take off
            c. Follow a rectangular path of 3x3x3m
            d. Autonomous land

______ on later stage ______

j. Task - 1: [later] Manual flying of the drone velocity mode and attitude through scripts
    i. Manual fly the drone in velocity commands mode through main.py file
    ii. Manual fly the drone in attitude commands mode through main.py file
k. Task - 2: [later] Position controller in velocity mode
    i. Make a simple altitude controller
        1. Tune its gain values
    ii. Stabilize the drone at a particular point
l. Task - 3: [later] Position controller in attitude mode
    i. Make a simple altitude controller
        1. Tune its gain values
    ii. Stabilize the drone at a particular point

2. Read the CAM device data on Rpi
    a. Make a .py script to read the data at GPIO pins of the Rpi
        i. Let's do it in C++
    b. Use software serial to make a ROS node that publishes cable_states.msg
    c. This data also should be published at least 100 Hz

3. Write a custom script to fly the system
    a. [Safety protocol] Drone collision avoidance system
    b. Make sure that the two drones should maintain more than [x] meter of distance in the horizontal plane
    c. Read the two drones' data in this code and verify them by plotting or logging
        i. Do a real-time simulation on Rviz
        ii. Debug the current state of the drone
        iii. Also, do a simulation of the cable attitude
    d. Check the user's data using rqt_plot and verify by logging
    e. Make a final code Cable attitude controller and payload attitude controller
    f. Make a code to change the mode of the drones using a radio controller
    g. Make a node to change the modes/fail safes using the keyboard

4. [done] Make communication between Rpi and Cube
    a. Go to location - /opt/ros/noetic/share
    b. Check the rate of the data by changing rosrate(100)
        i. Make sure it should work without overriding the data
            1. If possible, you can check this using rqt_plot (later on)
        ii. The data rate should be transferred at least 100 Hz
        iii. If transferring all the data reduces the rate, then only transfer the necessary data

5. Connect the PC's ROS to Rpi's ROS using wifi communication
    a. Make a hotspot and create a server
    b. Username - hcrlab
    c. Password - hcrlab8318
    d. Configure the rpi to do the following once it boots
        i. Connect to the server
        ii. Publishing the drones' states by auto-running the drone's states
            1. In this file, it will be the subscription of the user's commands

6. Make custom messages in ROS for the quadcopter's state as mentioned below
    a. Name of the message "quad_states.msg"
        i. float64 quad_x
        ii. float64 quad_y
        iii. float64 quad_z
        iv. float64 quad_ph

        v. float64 quad_th
        vi. float64 quad_ps
        vii. float64 quad_x_dot
        viii. float64 quad_y_dot
        ix. float64 quad_z_dot
        x. float64 quad_ph_dot
        xi. float64 quad_th_dot
        xii. float64 quad_ps_dot
        xiii. float64 quad_x_dot_dot
        xiv. float64 quad_y_dot_dot
        xv. float64 quad_z_dot_dot

7. Make custom messages in ROS for the CAM device’s state as mentioned below
    a. Name of the message “quad_ph
    b. g”
        i. float64 cable_ph
        ii. float64 cable_th
        iii. float64 cable_ph_dot
        iv. float64 cable_th_dot

8. Make custom messages in ROS for human inputs from the joystick as mentioned below
    a. Name of the message “human_inputs.msg”
        i. int16 channel1
        ii. int16 channel2
        iii. int16 channel3
        iv. int16 channel4
        v. int16 channel5
        vi. int16 channel6
        vii. int16 channel7
        viii. int16 channel8
        ix. nt16 channel9
        x. int16 channel10
9. Make custom messages in ROS for the quadcopter commands as mentioned below
    a. Name of the message “quad_high_level_cmd.msg”
        i. float64 x_des
        ii. float64 y_des
        iii. float64 z_des
        iv. float64 x_dot_des
        v. float64 y_dot_des
        vi. float64 z_dot_des
        vii. float64 ph_des
        viii. float64 th_des
        ix. float64 ps_des

Youtube Video

1. **PixHawk Video Series - Simple initial setup, config and calibration - link**

So, there is a flight controller called Apm which has arduino based system. The more advanced version of this was launched which is Pixhawk. Everything is more advanced in pixhawk like cpu , memory , sensors etc.
First we have connected external GPS system to our pixhawk board using to pin connector, one pin at GPS slot and another magnetometer pin at i2c slot.
Then we will connect the receiver(FR sky-D4R 2)to the board through s-bus or cppm.
Then we will connect the board to the PC, downloading mission planner.
Safety switch and buzzer : will install later
Setting up frame type , acclerometere, magnetometer/compass, radio calibration anf flight modes in mission planner software.

Power to Pixhawk : Will use power module having 6 pins insertion to the board , will supply 5 volts and current sensing information through the pins.
ESC : ESC means Electronic Speed Controller and it is a device t used in drones to control the speed and direction of the motors. The ESC takes a signal from the flight controller and converts it into a voltage that controls the motor speed. By varying the voltage supplied to the motor, the ESC can control the speed and direction of the motor rotation.
Connect ESCs to the respective 1,2,3,4 slots to the pixhawk board.
Flight Modes:

- Stablize:- Not using GPS, controlled manually by the pilot using the transmitter's joystick.
- Loiter :- drone utilizes GPS to maintain its position and altitude, remains in a fixed position.
- Auto :- Fly at predetermined GPS waypints, not requires any manual control.

Smit :
16/05 : Understood the project and learned practically various components of drone(small one) and its assembly.
17/05: Set up the big drone ,learned Mission planner basics and calliarated the big drone, had a successful test flight on small drone in stablize, loiter and auto mode.
18/05 : planned and set up various missions on small drone and conducted test flight to check it. Designed and laser cut GPS mount for big drone.
20/05: Planned mission on big drone and fly it. Dismantled it to understand its behaviour. Inspected the flight logging data for the flight.
22/05: Explored Pix 4d software and completed a basic project on it. Fixed the arm issue of the big drone and started assembling it.

23/05: Fix the 3D printed battery mount on the drone and calibrated it. Flew the big drone in stablize,loiter and auto mode successfully.
24/05 : Fixed landing gear issues and installed the spring in the RC transmitter.
25/05 : Conducted missions at the terrain near research park and processed the images in pix4d to obtain 3d model of terrain.

Drone components

Chirag Research proposal:

The use of unmanned aerial vehicles (UAVs), particularly quadcopters, has gained significant attention in various industries, including surveying and mapping. Quadcopters offer a flexible and cost-effective solution for capturing high-resolution imagery with an option of autonomous flying and taking images at particular locations with the help of GPS. This research proposal aims to investigate the application of quadcopters for surveying three locations in the IIT Gandhinagar campus and then generate a 3D map of the surface of these areas, measure area and gather vegetation information using the data collected.

Our objective for this project is to fly one drone with a camera on a predetermined path and let it take images at predefined GPS coordinates of the area under survey. Then using this 2D data and some mapping tools, generate a 3D model of the survey area.

Milestones:

- Formulate safety protocols which are to be followed before flying the drone and get appropriate permissions for the survey.
- Make a small drone, fly it in loiter and stabilise modes, and then fly it autonomously on a waypoint mission.
- Fly the bigger survey drone too autonomously on a waypoint mission and integrate GoPro into it.

- Study Pix4D software and do 3D surface generation, area measurement and vegetation information gathering using the existing data set.
- Carry out the same again but by flying the survey drone in a smaller area and using its data.
- Conduct the case studies at the IIT Gandhinagar campus by carrying out surveys using drone and processing the gathered data. Locations identified fit for this survey:
    - i. Ravines near to research park area
    - ii. Vegetable farms
    - iii. Near to Sabarmati river

Chirag:
12/05/2023 : Understood what project is exactly about
15/05/2023 : Assembly of small drone, writing code for taking photos using go pro, written few safety protocols and found good references
16/05/2023 : completed writing code for taking photos using go pro, 3d cad modelled and printed attachment for go pro, started working on to find and implement ways to trigger GoPro using flight controller.
17/05/2023: Triggered GoPro using flight controller, completed setup and calibration of both small and large drone, flew small drone on loiter, stabilize, altitude hold, position hold, rtl successfully and also completed one way point mission.
18/05/2023: Learnt about data logging, viewing and it's analysis on mission planner, learnt about setting up survey missions, did multiple waypoint and survey missions on small drone and did it's data logging and analysis, updated code for node MCU, learnt and did simulation on mission planner, laser cut a acrylic mount for GPS.
19/05/2023: 3D cad modelled and printed an updated version of go pro and battery mount, started learning about pix4d and did a practice project.
20/05/2023: Updated the 3D cad model and put it for print, programmed big stone for a survey mission, tried to fly big drone (fail), analysed flight data, disassembled big drone.
22/05/2023: Updated the 3D CAD model of gopro mount and put for print, assembled the entire big drone
23/05/2023: Attached the 3D CAD model on drone. Flew the big drone in stabilize, loiter and auto, used GoPro to take pictures automatically and logged all the data.
24/05/2023: Big drone had issues with landing gear and the 3D printed battery and go pro mount, rectified that. Modified RC transmitter hardware according to our needs. Got permission to fly the big drone at higher altitudes.
25/05/2023: Flew the drone near research park ravines, did two survey missions and collect images, processed these images on pix4D to get 3D map of terrain, had obtained only half the images so found out the reason for it using flight data log.
26/05/2023: Flew the drone near research park ravines, did two survey missions and collect images, processed these images on pix4D to get 3D map of terrain.

27/05/2023: Analysed the 3d map, created animated video of the 3D map and tried to retrieve data from it. Found that images from the camera were blurry in some areas, found reason and started looking for solutions for this issue
29/05/2023: Started making presentation on work progress and also started researching on various aspects of the project (Agricultural use and markets, how pix4D works, it's alternatives, ways to improve functionality of alternatives, better camera options, scenario of surveying world and India, etc.), installed meshroom and uploaded previous survey data for 3D map formation.
30/05/2023: Continued research on above topics.
31/05/2023: Made 3d model with meshroom and found it to be less accurate than pix4D. Continued research on above topics, found uses of and indices used in multispectral photogrammetry. Searched for multispectral cameras that can be purchased in India. Put the previous data again for processing in pix4D but with additional settings and tweakings.
1/06/2023: Analysed pix4D map, calculated area and volume of ravine area, made r, b, g, grauscale index maps. Continued research on project. Updated presentation regarding all of this.

__2nd project:__ Saw tutorials on raspberry pi and pixhawk integration, also saw tutorials and did installation of ardupilot, sitl, gazebo, Qground control, mavproxy and other dependencies on ubuntu. Ran simulations on gazebo and Q ground control.

2/06/2023: Started seeing tutorials on ROS, gazebo and sitl (drone software development) and started implementing it.
3/06/2023: Continued seeing tutorials on ROS, gazebo and sitl (drone software development) and continued implementing it.
5/06/2023: Continued seeing tutorials on ROS, gazebo and sitl (drone software development) and continued implementing it.
6/06/2023: Continued seeing tutorials on ROS, gazebo and sitl (drone software development) and continued implementing it.
7/06/2023: Continued seeing tutorials on ROS, gazebo and sitl (drone software development) and continued implementing it.
8/06/2023: Continued seeing tutorials on ROS, gazebo and sitl (drone software development) and continued implementing it. Also watched ctu prague workshop video.
9/06/2023: Completed seeing tutorials on ROS, gazebo and sitl (drone software development) and implementing it. Started to do the implementation on raspberry pi.
10/06/2023-28/06/2023: Implemented ros on raspberry pi, made scripts for increasing imu data frequency, attitude control, custom ros messages and topics and launch file for all of these, attached everything on drone in a very compact form and tested drone in hand for response in attitude mode in offboard flight mode based on rc inputs.

Safety Protocols:

While programming pixhawk:

- Check that battery is disconnected and remove all the propellers

- Check that all components are firmly attached to the drone
- Make sure arming checks and failsafes are in place.

Before,during and after flying:

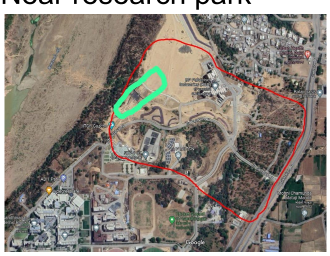
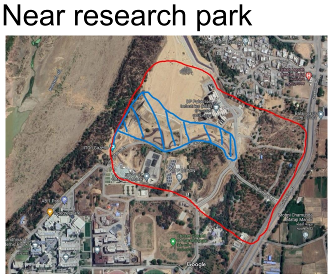
1. Check that all components are firmly attached to the drone
2. Battery puff level and voltage level.
3. Connect battery and make sure safety switch is off
4. Start on a plane surface
5. Make sure no is near the drone before arming and also make sure to have GPS lock.
6. Takeoff in stabilize or loiter mode(loiter preferred)
7. Fly the drone cautiously
8. If drone in auto mode then always keep an eye on it and if drone behaves erratically or there is an eminent collision, quickly move the drone to stabilize or loiter mode.
9. If a crash is unavoidable, disarm the drone at correct moment
10. After landing, once the propellers stop, press the safety switch till it starts blinking.
11. Disconnect battery and detach battery and go pro from the mount
12. Check battery temperature and voltage

Good videos and webpages to take reference from:

- Aerial mapping : An Overview of Aerial Mapping with Pixhawk, Pix4D, and GoPro Hero3+ - YouTube
- Pixhawk wiring: Pixhawk Wiring Quick Start — Copter documentation (ardupilot.org) Connect ESCs and Motors — Copter documentation (ardupilot.org)
- Pixhawk firmware: Determining PixHawk FMU version - ArduCopter - ArduPilot Discourse
- Go pro commands: goprowifihack/HERO6-Commands.md at master · KonradIT/goprowifihack · GitHub
- Go pro remote: DIY GoPro Wi-Fi Remote Using ESP8266 - Hackster.io
- Arduino and pixhawk: https://discuss.ardupilot.org/t/mavlink-and-arduino-step-by-step/25566
- Image acquisition: Image acquisition (pix4d.com)
- How to select/change the images geolocation and orientation:

How to select/change the images geolocation and orientation (pix4d.com)

- Some more links:
  How to verify that there is enough overlap between the images - PIX4Dmapper
  How to verify that the Image Quality is sufficient - PIX4Dmapper
    - Geotagging Images from a Telemetry Log - Informatics and GIS Program (ucanr.edu)
    - TuffWing - Geotag pictures with a Pixhawk log file and Mission Planner
    - Geotagging Images with Mission Planner — Copter documentation (ardupilot.org)
    - Geotag GoPro images with a Pixhawk Log File using Mission Planner and known coordinate system - PIX4Dmapper / PIX4Dmapper Questions/Troubleshooting - Pix4D Community
    - Tutorial: Meshroom for Beginners — Meshroom v2021.0.1 documentation (meshroom-manual.readthedocs.io)

	Date and Time	Place	Description
1	17/05 , 5:30PM	Behind Jasubhai Audiotorium	Fly small drone in all stablize, loiter and auto mode. Auto mode(a line and a circle at 10 meter altitude)
2	23/05/2023 6:00 PM to 6:30 PM	Behind Jasubhai Audiotorium	Flew big drone in stabilize, loiter and auto. 10m height
3	25/05/23 3:30 PM to 5:30PM	Near research park 	Did a survey mission on big drone. Alt 30m
4	26/05/23 5:40PM to 7:00PM	Near research park 	Did a survey mission on big drone. Alt 30m

RESEARCH:

- Camera:
    - **Camera parameters to take into consideration -**
      Shutter speed (no blurry images when on the go)(but too much shutter speed = less exposure)(min 1/1000)
      Also iso (noise)(100-800)
      24 to 36mp recommend, 12 min
      Lens (wide), 18 to 50mm
      Camera selection with all good options: How to Choose the Best Camera for Photogrammetry | Vision Aerial
      What Camera Should You Use for Photogrammetry? | by 80Level | Medium

      A shutter speed of 1/250 sec should be fast enough to freeze people walking around, whereas 1/500 sec is better if your subject is moving a bit quicker. For faster objects such as cars and birds in flight, shutter speeds of 1/2000 sec, 1/4000 sec or quicker are preferred~canon
      All you need to know about motion photography - Canon Europe (canon-europe.com)

      Current camera: GoPro Hero 7 Silver Action Camera Review - Is It Worth It? - Action Gadgets Reviews
  - Agricultural research(crop health, pesticides, etc), multi-layered camera(IR cameras and all)
    - **Drone height: How to figure out the optimal drone flying height for mapping! (geonadir.com)**
    - Use and benefits of multispectral camera their use in agriculture etc.: Multispectral Drones: Benefits and Use Cases – heliguy™
      - Multispectral Drone Imaging | Multispectral imaging India| AIRPIX
      - The 5 Best Multispectral Cameras For Drones - Aetha
      - Multispec cams in india:
        - PARROT SEQUOIA+ Multispectral Camera – Asim India
        - Multispectral Cameras at Rs 360000/piece | CCD Camera in Bengaluru | ID: 6393493948 (indiamart.com)
        - Parrot Multispectral Camera at Best Price in Kolkata, West Bengal | Rchobbytech Solutions Private Limited (tradeindia.com)
    - All about drones in agriculture: vandermerwe2020.pdf

  - Carrying out at large scale(Pirated projects)
    - Multiple flights can be conducted ensuring that ample of overlap in photos is available at junction point (did this)
    - Note: multiple drones can simultaneously carry out survey but use same camera, properly geotag all the images.

- For bigger areas usage of fixed wing better due to higher payload capacities(better camera), longer flight time (more area) and comparatively stabler flight. Fixed-wing vs multirotor drones: Which is better? (geonadir.com)

- Current survey methods and Indian scenario(cost)
  Link for commercial drones:Top 10 Advanced Drones for Aerial Photography that are Irresistible (analyticsinsight.net)
  Drones and digital photogrammetry: from classifications to continuums for monitoring river habitat and hydromorphology - Woodget - 2017 - WIREs Water - Wiley Online Library

At present, there are government guidelines in India for aerial mapping and photography.(https://static.pib.gov.in/WriteReadData/specificdocs/documents/2022/mar/doc202232932501.pdf)
There are many companies for aerial mapping and surveying in India, few are mentioned below:

1) Aerial Photo : https://aerialphoto.in/
2) Drone Nation Solutions: http://www.dronenation.in/
3) Aarav Unmanned Systems Private Limited: https://aus.co.in/
4) Survey of India: https://surveyofindia.gov.in/
5)

There are wide range of applications of aerial mapping and survey :

1) Urban Planning: City planning, traffic management , Land use patterns, topography(https://hobitech.in/urban-planning-using-drones/)
2) Agriculture : Crop health & stress analysis, irrigation monitoring and planning, field soil analysis, pesticide spraying.(https://www.equinoxsdrones.com/agriculture)
3) Environmental Monitoring : Monitor ecosystems, detect changes in vegetation cover, assess wildlife habitats, track deforestation or land degradation, ocean health(https://www.azocleantech.com/article.aspx?ArticleID=1519)
4) Infrastructure Design and Construction: 3D models help costructuion of bridges, railways, Pipelines , highways.
5) Flood Modeling and Risk Assessment: Detailed research paper (https://www.mdpi.com/2073-4441/12/6/1717)

Commercial drones uses for aerial transportation:
https://www.analyticsinsight.net/top-10-advanced-drones-for-aerial-photography-that-are-irresistible/