

Final Project Report

For

APM Rover System [APM Rabbit]



Table of Contents

Table of Contents	
Group Members	
1. Final Project Summary.....	
1.1. Program Interface.....	
1.2. Program Structure	
1.2.1 Android Code	
1.2.2 Project Website	
1.2.3 GoPro.....	
1.2.4 UML Diagram	
2. Security	
3. Employability	
4. Improvements and Extensions.....	
5. Difficulties Encountered	
6. Conclusion	

GROUP MEMBERS:

Name	Role	Responsibilities
Kenny Hunter	Project Manager & Database	Oversee project activities, supervise solution integration into existing project build up. Design and install database
Himangi Topiwala	Algorithms & Project Documentation	Build the projects algorithms and supervise the project navigation as per algorithms designed. Made the user manual(Documentation) of the project
Sanvir Kaur	User Interface & Final Presentation	Build user interface and make recommendations for the App's design and interface. Made final presentation on the project

1. Project Summary

APMRabbit was created by students as a part of the final computer systems project. APMRabbit software was created for the APMRover Project inspired by the rover project located (<http://rover.ardupilot.com/>). The APM: Rover Unmanned Ground Vehicle(UGV) is an autonomous rover, using the APM board that can run missions following a list of recorded GPS waypoints.

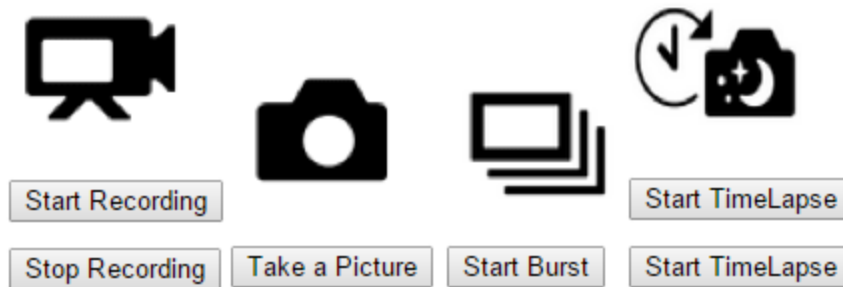
APMRabbit has two main applications, a viewing application that uses the GoPro Hero 3+ camera to stream, shoot video and take pictures and the APM communicator to control the rover in any direction it wants i.e. APM Rover App.

1.1 Program Interface

This section explains how the user communicates with the project. Download the APM rover App from the Google Store. Start the APM rover with the power button. Then switch on the GoPro camera as well. Then from the App user can select the options to do with rover i.e. change the rover mode from steering, manual, automatic, learning. All the hardware parts needed by the user to build up the Rover are listed on the project website under "Project Parts". The wooden base of the rover is made using the laser cutting tool and the design is created with Corel draw software.

The App can be downloaded from the Google App Store with the name APMRabbit.

APMRabbit GoPro Software: This project also includes the streaming website of the GoPro. If users does not have an Android OS they can still view their GoPro live streaming through the Website. In stream section, one need to enter password to connect with GoPro Camera. Then one can select basic camera modes:



Then from the settings section, user can modify from more advanced settings of the GoPro. This includes:

- Video resolution: Here user can change the resolution settings to their choice.
- Auto Low Light: Allows user to turn auto low light feature on or off
- Color Settings: Allows user to choose between flat and go pro color
- Sharpness: Include options to change sharpness from Low-Medium-High
- Exposure Compensation: Allows user to change exposure compensation in the range -2.0 to +2.0 with the interval of 0.5
- Video Protune resolutions: Allows user to on protune resolutions and also change it to other value.
- White Balance: Allows user to change the white balance of the video.
- Orientation: Allows user to change the orientation into vertical or horizontal

APMRabbit App:

1.2 Program Structure

This is the “technical manual” of the program. This section discuss how the backend programming was developed for this software.

Android Code: This section will lay out all the android classes behind the app.

- **APMRabbit.java**: This is the main activity of the app. This class extends `FragmentActivity` which implements tab listener. This class includes the handler to choose between the tabs (Rover, GoPro,History).
- **GoPro.java**: This activity is set as the default page of the app. In short this is the first tab of the app. This activity extends `Fragment` which implements `view.OnClickListener`, and `sharedPreferencesListener`. Theres a video view that handles the live streaming captured from the GoPro. Also there are Image Buttons for video, photo mode, burst mode, time lapse mode. To control the video streaming, the activity has buttons for start and top recording. These options are stored locally on a device as a shared preference file, so when the user clicks on

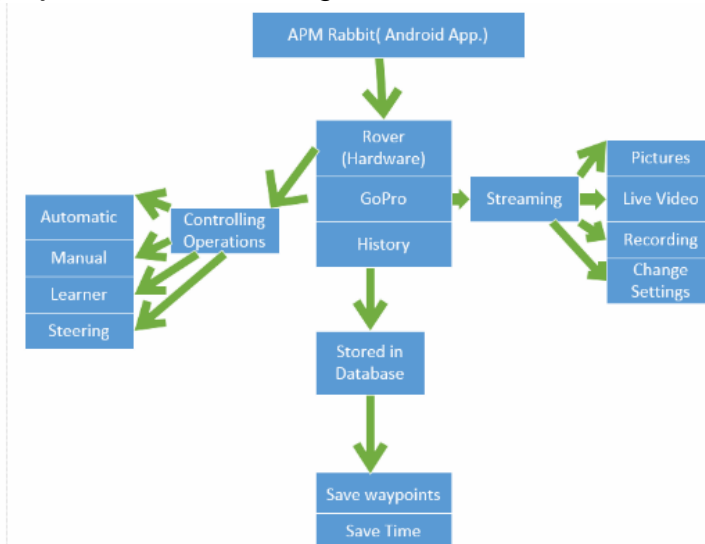
one of the image its corresponding id stored in the file is picked up and does the work accordingly.

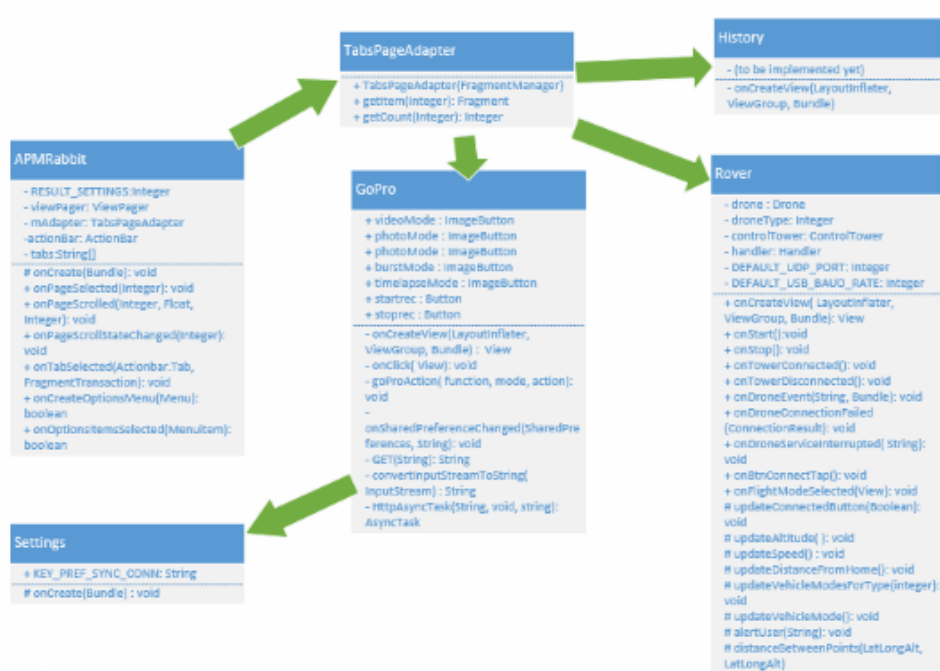
The goProAction() method of this class validates for the password of the GoPro. If the password provided is correct then the user is able to communicate with the GoPro settings.

- **Rover.java**: This activity extends Fragment which implements DroneListener and TowerListener. In this activity, the first thing that is done is to connect to the rover. If the connection is successful then user can start the rover with selected mode type. This activity also shows the current home location and the distance of the rover from home. User can also see the speed of the rover updated.
- **Settings.java**: This activity takes the user to the advanced GoPro settings mode in the app.

Project Website: The project website is structured in the way where a user can get access to all of the resources for the project. This website includes a home page: to get to know more about the software. A 30-sec video: that shows how the app is interacting with the rover. The final budget report of the software. The information about the purchase of the parts needed. The source code section redirects the user to our GitHub account where they can download any source code i.e. HTML code of the GoPro Website or APMRabbit App.

UML Diagram: As a part of the algorithm of the software this section provides the UML design of the project which helps the user to have a better understanding of the project in terms of navigation and work flow.





2. Security

Anytime a new server or device is introduced to a network it represents a new potential vulnerability and access point for malicious activity if not properly safeguarded against threats. The APMRabbit software is integrated in such a way that it uses password validation to connect with the go pro which is a level 1 security check.

Secondly, GoPro's own wifi environment is encrypted and secured from other networks, hence users connected to this wifi only can have access to all gopro features. This software has been made taking into account our privacy policy i.e. The Rover must be operated in a fashion that does not needlessly compromise the privacy of non-public figures. If at all possible, record only images of activities in public spaces, and censor images of private individuals in private spaces that occur beyond the scope of investigation

3. Employability

The scope of this project was to develop a portable software application for the autonomous rover was designed to meet several initial strict requirements. The project had to be developed at a minimal cost, and meet the course requirement of Humber College. The selection of software and hardware products was carefully chosen to conform to the necessary design requirements. Eclipse Application was selected as low cost android solution that could run on hardware, in this case an Android phone.

The database and web interfaces were developed with team members, providing future support for database changes, upgrades or the addition of new features or capabilities should the need arise.

The final APMRover project produced by the team successfully meets all the pre-determined design requirements and operational criteria. The project makes use of all possible network infrastructure in place, and requires minimal additional components to be purchased and no outside contracting to complete the software development.

4. Improvements and Extensions

In this section, the parts of the program that need improvement and some possible future extensions are discussed. In terms of future direction our team has decided to make the project as open source. We have thought for this because we are welcoming equal opportunities who can take our project and can improve any aspect and can extend its liability and performance. This project is open to have feedbacks that are based upon changing or improving something valuable to the software or the hardware aspect.

5. Difficulties Encountered

The major difficulty that our team faced during the course of development was not being able to run the rover in automatic mode. Because of this difficulty we could not develop testing's that would need the rover to run automatically.

6. Conclusion

As a conclusion, APMRover project has been built taking into account all the course requirements. Our team has strived to provide all the possible details that can be given in order to reproduce the project in future.