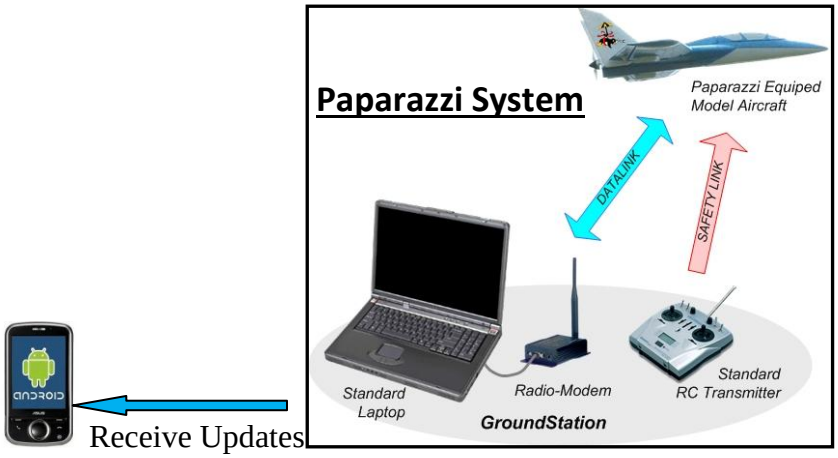
(I) Overview

Android application that communicates seamlessly with Paparazzi ground system. “Paparazzi” is open source software that simulates the autopilot system for fixed wing UAVs as well as multicopters. The Paparazzi consists of two parts: 1. Ground system 2. the hardware. It has an option of either using hardware UAV or using the simulator. The ground system communicates with the UAV i.e. sends and receives data through data channels. It can send motion commands to the UAV through these data links.

Currently the only way to control the UAV is through the input of the ground station. This application enables a user to connect to the UAVs via the ground system and control it. The app can send commands and receive data updates such as battery level, altitude, ground speed etc.

The complete system consists of three modules:

1. Android Application
2. Paparazzi System (UAV)
   a. Ground System (GS)
   b. UAV (currently using the simulation)



- *Android Application*:
  - The app is analogous to the ground system control panel. The commands that originally the GS could send the UAV can now be sent via the android app. Thus not restraining the user to a particular position in the duration of the flight and providing the ease of use.
  - The App would also have provisions to receive from the GS, updates regarding the UAV’s critical attributes such as battery level, altitude, ground speed etc.

(II) Methodology

- ***Languages/Framework***
  - Android Programming requires code in JAVA, where as the paparazzi software is in C and OCAML. The multithreaded environment would be created on the android side thus would include the use of JAVA threads.

- ***Integration of the Platforms***
  - The Paparazzi software, android app that would also include the multithreading would be integrated into one system. The app would behave as a client to the ground station of the paparazzi. The android application and the GS would be communicating with each other in order to exchange data. The updating of data would be handled in different threads so that the working of the commands would be faster.

(III) Significance

- *Paparazzi System:*
  - It consists of 2 parts: the ground station (GS) and the UAV. This is the main module and the android application will interact with the UAV via the GS. The GS controls the UAV through the commands.

- *Android Application:*
  - It will increase the mobility of the user as he will not be restricted to the ground station. It would simulate the role of a remote control for the UAV. The app would connect to the GS and signals. The application would keep a record of the current UAV environment attributes and track it in real time.

- *Multicore:*
  - To enhance the performance of the android application, I would implement the sending of commands and receiving of signals on different threads. This will enable the app to keep the most recent update for each attribute and block upon the commands that are being sent or received.