Project Requirements

Introduction

(Assigned Author: Abhilash Kotha)

a. Purpose of this software project

Drones provide an excellent learning opportunity for science and technology educators, students, kids, and drone enthusiasts. They provide a hands-on learning experience about the principles of flight and electronics, among other topics. The current cost of a full drone setup, however, can quickly exceed one's budget. One potential solution to expand the availability of classroom drone projects is to replace the drone's standalone remote controller with a mobile app that allows the users to control the drones indoors or within a closed range.

The project intends to develop a mobile application that provides some of the basic functionalities of a drone's standalone remote controller.

As part of the project, we want to build an application that

- Is easy to use.
- Allows pairing between the drone's onboard receiver with the drone pilot's mobile device (iOS or Android) via Bluetooth.
- Sends requests to the drone's receiver to control the throttle, aileron, elevator, and rudder.
- Allows users to configure settings particular to their particular drones (i.e., degrees of movement for each control surface).
- Gets the current status of the drone (i.e., the current position of the control surfaces).

b. Intended Use

This application is intended to be used by anyone who owns a drone that is compatible with our mobile application and wants to use it either indoors or in a controlled range. By using this application users would be able to control the drone with respect to the below four parameters.

- Rudder: The rudder is used to control the yaw, or horizontal rotation, of the drone. When the rudder is moved left or right, the drone will turn in that direction.
- Throttle: The throttle controls the vertical motion of the drone. When the throttle is increased, the drone will ascend, and when it is decreased, the drone will descend.
- Elevator: The elevator controls the pitch, or angle of inclination, of the drone. When the elevator is moved up, the drone will pitch up and when it is moved down, the drone will pitch down.

- Aileron: The ailerons control the roll, or lateral rotation, of the drone. When the aileron on one side is moved up, the drone will roll in that direction, and when it is moved down, the drone will roll in the opposite direction.

Overall Description

(Assigned Author: Cory Gardner)

a. System stakeholders

- Student drone pilots, who should be able to increase their understanding of flight through increased accessibility to drone technology.
- Educators, such as Dr. Gururajan, who will benefit from lowered cost and increased availability of the drones for classroom use.
- The school system, which will benefit by the lowered cost of the course.
- Drone parts manufacturers/distributors, who supply the parts included in the drone kits.
- App store providers, such as the Apple App and Google Play stores, from where the app will be made available.
- Developers of the application

b. User Needs

- The user needs a less expensive drone controller to make drone kits more accessible for classroom purposes.
- The user needs an application with a GUI that mimics a typical model airplane RF controller and allows control of the in-flight movements of the drone (roll, pitch, yaw) via manipulation of the throttle and primary control surfaces (aileron, elevator, and rudder).
- The user needs to be able to uniquely pair each drone to the app, even in the presence of multiple drones.

c. Assumptions and Dependencies

Assumptions:

- The application will be for use on mobile or handheld devices.
- Users will have access to a compatible device (iOS or Android).
- The users' devices will have bluetooth capabilities.
- The application will be available via a central delivery method (i.e., the Play Store).
- The end-users will have basic knowledge of how to control and operate a drone.

- The drone boards will have firmware that is open source and provides a front-end API that can be accessed using common high-level programming languages such as python.
- The drone will have a mechanism to allow only one controller to be connected to its Bluetooth at a given time.

Dependencies:

- The application's pairing and communication will be dependent upon the protocol made available by the on-board receiver.
- The application's functionality will be dependent upon the methods made available by the on-board receiver.

System Features and Requirements

(Assigned Author: Okwudilichukwu Okafor)

a. Proposed process model

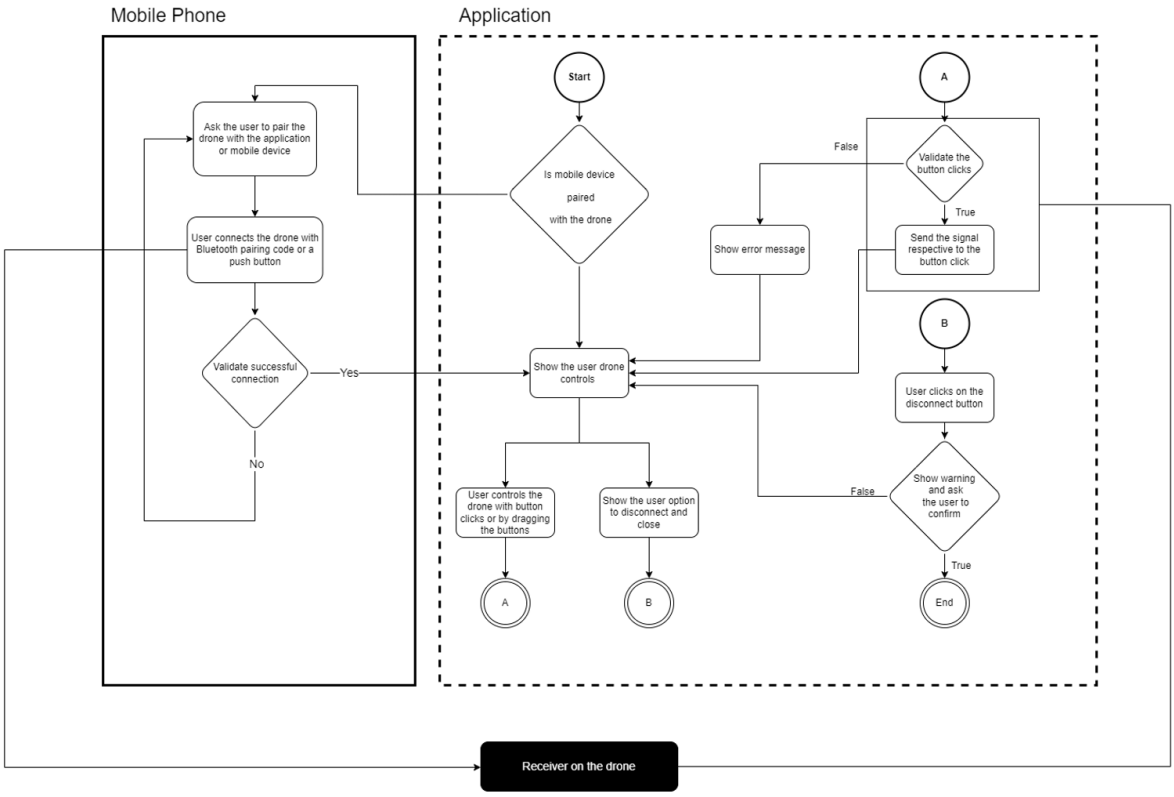
The image shown below shows how our mobile application works at the basic level. Everything outside the dotted rectangular box is external to the system that we are going to build.

The app requires the user to have their mobile device's Bluetooth and the drone's Bluetooth turned on. The first step involves pairing the mobile device with the drone's receiver. If the pairing is unsuccessful, the user is informed and prompted to retry the process until it is successful.

Once the pairing is successful, the user will be shown drone controls like throttle, aileron, rudder, and elevator along with a close button that disconnects the drone from phone

The user controls the drone using the controls displayed on the mobile device's screen. The app validates the user's commands to ensure that they are valid for the drone. If the command is valid, the app sends the respective signal to the drone's receiver, and the drone executes the command. If the command is invalid, the user receives an error message, and they can attempt to control the drone again.

Finally, If the user wants to disconnect the app from the drone, they can do so by clicking the close button displayed, and the drone will be unpaired from the mobile device and the user can exit the app.



b. Functional Requirements

FR1. A user should be able to connect the mobile application to the drone via bluetooth.

FR2. The user must be able to see controls for the Rudder, Throttle, Elevator, and Aileron.

FR3. The application should allow manipulation of the throttle, aileron, elevator, and rudder, thereby allowing the end user to control the drone's flight the same as if using an RF remote.

FR4. The user should be able to disconnect the drone from the application.

FR5. The application must be able to control the drone to not cross the Bluetooth range.

c. Non-Functional Requirements

NFR1. The size of the app should be at most 50 MB allowing a user to easily download and install the app.

NFR2. The app must use simple icons or symbols to indicate the functionality thus improving ease of use.

NFR3. The application should be compatible with major mobile operating systems such as iOS and Android.

NFR4. The launch (start) time of the app should be at most 2 seconds.

NFR5. When a user enters an input (that is, presses a button), the response from the app should be done within 0.5 seconds. These responses include arming and taking off the drone, and other movements of the drone.

NFR6. As multiple drones will be in the same area, the connection must remain unique between each phone and drone.

NFR7. The application must be able to record all the logs of button clicks and events.

d. Domain Requirements

DR1. Connection between the mobile app and the drone has to be done via Bluetooth Low Energy wireless technology.

DR2. The application must ensure that it is not sending two conflicting signals to the drone like elevation up and down at the same time

DR3 The application should not send signals with impossible deflection (example: 100 meters elevation up with each click)

Appendix

Questions for clarification:

Q1. Is the target drone hardware a quadcopter or fixed wing aircraft?
Q2. What would the user expect the drone to do when the connectivity is lost?
Q3. What mode of pairing is offered by the drone receiver? (i.e., push-button, access code, et cetera).