Allison Beasley

Bobby Scott

CS380 Final Write Up  
  
**Project Description:**

A User-Tracking Drone which integrates technology to create a versatile drone system capable of actively following a user's movements. Combining an Arduino uno microcontroller, GPS module, Xbee radio, and a web application this system enables the drone to interpret GPS data, adjust its flight path accordingly, and communicate with a Python program on a computer to facilitate seamless user interaction and allow for the visual presentation of GPS data on a map in a web application.

**Key Components:**

1. Arduino with GPS and Xbee Radio: The Arduino is equipped with a GPS module for precise location tracking and an Xbee radio for communication. It collects GPS data and facilitates real-time communication with the Python program running on the computer.

2. User Interactive Buttons: Two buttons on the Arduino enable user control. Button one (white) toggles between flight and landing modes, ensuring safe takeoff and landing operations. Button two (blue) switches the drone between tracking the user and a stationary state, allowing the user to reposition themselves/the drone if needed.

3. Python Program: Running on a computer, the Python program receives GPS data from the Arduino, interprets it, and generates movement instructions for the drone. It facilitates dynamic tracking, allowing the drone to adjust its position based on the user's movements.

4. Web App Integration: Goal of uploading live GPS data to a web app which displays location on an updating map.

**Functionality:**

- User Tracking: The drone actively tracks the user's movements based on GPS data, adjusting its flight path accordingly.

- Flight Control: The Arduino's buttons provide manual control for takeoff, landing, and switching between tracking and stationary modes.

- User Flexibility: The system allows the user to switch the drone into a stationary mode for repositioning, enhancing flexibility during operation.

- Data Management: All GPS and flight data are uploaded to the AWS cloud, facilitating comprehensive data review and analysis for performance optimization and future enhancements.

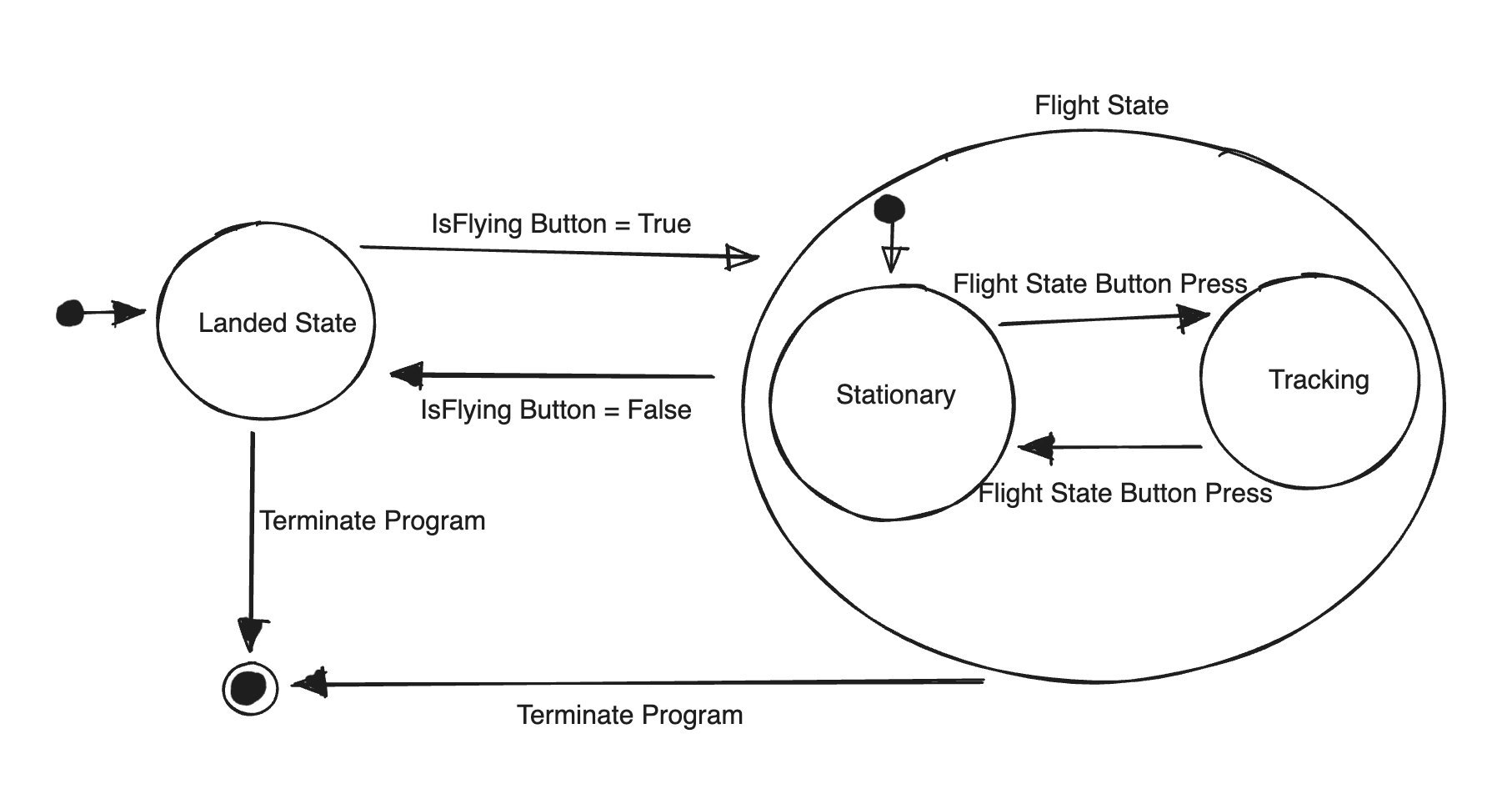
**Bugs:**

* Drone will occasionally throw error "error No valid imu" when it receives input. This can be due to the IMU Status requiring calibration and other drone-specific factors.
* GPS is not fully accurate the second it makes a satellite connection, required to wait an additional 20-60 seconds for the GPS to become reliable.
* Inconsistencies automatically updating web application with GPS data without blocking functionality for drone commands, so automatic updates for web app turned off for demo. Manual GPS updates are still enabled.

**Limitations:**

* GPS to centimeters of movement conversion has some rounding errors due to amount of longitude and latitude data produced by GPS (GPS produces 6 decimals of data).
* The drone must start pointing North since GPS and drone do not have cardinality capability
* The drone is lightweight and can be easily influenced by wind.
* The drone tends to drift even when in calm conditions influencing the flight path.
* Difficulty obtaining GPS signal indoors, so project must mainly be used outside

**Finite State Machine Diagram:**



**Circuit Diagram:**