Quadcopter Cameraman

Project Plan

Team Number: SDMAY19-42
Client: Mir Aamid Ahbab (Descarga Latin Dance Club)
Advisers: Zhengdao Wang

Luke A. Rohl — Scribe, Drone Developer
Mir Aamid Ahbab— Chief Engineer, Hardware Engineer
Isaac Holtkamp — Software Developer, Software Tester
Nate Allen - Software Developer, Data Analyst, Software Tester
Alexander Nicklaus — Embedded System Developer, Real Time Systems Engineer
Team Email: sdmay19-42@iastate.edu
Team Website: https://sdmay19-42.sd.ece.iastate.edu
Revised: October 26th 2018 / Version 3.0

Table of Contents

List of Figures

List of Tables

List of Definitions

Quadcopter - A robotic unit that utilizes four motorized propellers to move in 3 dimensional space

Drone - Synonymous with 'Quadcopter'

Quadcopter Cameraman - The name of the project and the generalized name of the robot being developed

Module - An electronic hardware device such as GPS, Barometer, Accelerometer, ect. Installed to the Quadcopter platform

Target - In context related to the project, a target is a human being recognized by the Quadcopter and image recognition software. Typically, the target is one or both of the performing dancers

User - A person intended to interact with the Quadcopter Cameraman deliverables including the performing dancer(s), android app handlers, or quadcopter technicians

Track - Used as a verb in context with the project. Means to know one or both dancers' location(s) within the frame of the Raspberry Pi camera on board the Quadcopter Cameraman

Frame - The Quadcopter Cameraman will record video of a dance performance. A frame refers to a still image which is one of many still images which compose the full video

GPS - Acronym for Global Positioning System

Barometer - A hardware module which calculates air pressure. This is used to determine the altitude at which the air pressure reading was taken

Accelerometer - A hardware module which calculates the module's acceleration in the x,y, and z axis

On-platform - The physical quadcopter

Off-platform - Not the physical quadcopter

Onboard - synonymous with 'On-platform'

Facial identification - The ability to identify the presence of a face in a picture frame

Facial recognition - The ability to identify the presence of a particular face with a certain amount of confidence rating.

Arm the drone - Allows the motors to turn

Disarm the drone - Disables the motors from turning

Lighthouse - Will check the surrounding area by turning in place

1. Introduction

1.1. Acknowledgement

The Quadcopter Cameraman team would like to kindly thank Iowa State University and the College of Electrical, Computer, and Software Engineering for promoting student professional experience and sanctioning this cross-disciplinary project. As students, the team appreciates the university for prioritizing outstanding issues with the current small equipment checkout system from the Electronics and Technology Group (ETG). Quadcopter Cameraman team would also like to thank ETG for their mentorship in developing the team's professional skills, for allocating human and financial resources, and for sharing their workspace with a handful of engineering students.

1.2. Project Statement

Problem Statement

The problem that we are trying to solve is that the Descarga Latin Dance Club on campus is having difficulties recording themselves and other members during performances. The main issue is that a camera man can be obtrusive on a dance floor and get in the way of the dancers themselves or other dancers that may or may not be on the floor at the same time. We also wanted to provide a more dynamic approach to recording dancers. Most dances involve moving across the dance floor and will make a stationary tripod less optimal and result in a poorer quality video. To solve this issue our client has hired us to design, build, and program an autonomous quadcopter. This quadcopter will be able to identify the target dancers and follow them around at a preset distance.

Scope

The objective is to build a camera drone capable of maneuvering and keeping multiple people within the frame. To minimize externalities, we have set the project in the context of a dance performance. Target tracking and following are our primary goals. Performance and stamina come secondary. The drone should be able to follow the lead dancer at all times. The second dancer should be in frame whenever possible. As for performance, the drone should react quickly enough to create a seamless and effective recording of the dance. In terms of endurance, the drone should be able to maintain flight for the extent of the dance, a maximum of 5 minutes. The types of dances include swing, west coast, salsa, and bachata.

1.3. Operating Environment

The expected operating environment will be indoors, because of this the drone is not expected to account for changes in wind or at level obstructions.

1.4. Intended Users

The intended audience for this project will be dancers with little to knowledge of the inner works of the drone. Therefore it will be required that the drone be user friendly and easily manageable through the android app user interface.

1.5. Assumptions

Assumptions

- The dance is less than 5 minutes
- The dance has no audience
- Paired dancers are always easily distinguishable from one another
- The dance floor is level
- The auditorium has no obstacles
- The walls will not be so close as to pose a hazard to the drone
- All people involved in the use case will be easily distinguishable from one another (no twins or look alikes that could confuse facial recognition)

1.6. Expected End Product and Deliverables

At the end of the project, our team will deliver one functional autonomous drone capable of tracking a pair of dancers for the purposes of recording them. We will also provide an android app to our client so that interaction with the drone is possible. Lastly technical documentation will be delivered so that our client is able to utilize the drone. The technical documentation will also provide information in the event the client must repair or replace components in the event of damage. Our overall project success will be measured through an accuracy goal for video tracking of both dancers being in 80 percent of frames. We will present our deliverables to our client in the first week of May.

2. Proposed Approach and Statement of Work

2.1. Functional Requirements

Flight Control

The drone needs to be able to control each motor individually. The input for control will be sensors and processed image recognition which will output commands to the flight control which will in turn control the engines.

Image Recognition and Tracking

The drone will need to track the dancers while they move. To accomplish this the camera will stream camera information to the onboard pi. The pi will process the image and tracking data which it will further process to create flight commands that it will send to the flight controller.

PID Controllers

To control the drone the image processor will take frames from the video stream and output data which the PID controller can use as its current point, where it is in relation to the dancers. The PID controller set point, where it should be in relation from the dancers, should remain static i.e. we want the drone five feet from the dancers. Another PID controller will also be fed data from either a barometer on the flight controller or an additional sonar for the purposes of determining and maintaining a constant altitude. The Flight controller may be able to maintain altitude on its own depending on the hardware we purchase.

Video Quality

The drone shall be able to record video at a minimum quality of 720p. The video footage acquired will have both dancers present in 80 percent of frames.

2.2. Constraints and Considerations

2.1.1. Non-Functional Requirements

Security

The project has no reliance on a database. There are no user accounts, passwords, and data to be stored. However, the raspberry pi will act as a server for the android app client device to connect to. This device will send commands to the raspberry pi which will control many aspects of the quadcopter’s behavior. This behavior can range from autonomous protocol activation/deactivation, altitude adjustments, movement forward and backward as well as side to side, platform rotation, motors turn on/off. Thus it is of high importance that the raspberry pi can only be connected to be the intended device. All device communication should not be susceptible to Replay Attacks.

Responsiveness

The purpose of this drone is to track dancers and capture video of their performance. The drone’s movement can range from a stationary, hovering position to wild unpredictable movements depending on the dance. As the dancers move about the drone will have to move to keep them in center frame. It is of high importance that we minimizing the lag between dancers movement and drone movement in response.

Reliability

Given the extent of control which the android application has over the Quadcopter, it is imperative that the raspberry pi maintains connection with the android application. Restrictions should be placed on the quadcopter's freedom of movement to keep it within range of the off-platform device which is connected. Furthermore, when connection is lost, the raspberry pi should halt autonomous protocol and attempt to re-establish connection with the off-platform device.

Useability

The setup and usage of the product must be simple enough for any user to complete a all of the defined use cases with a minimal strain on the user's comfort.

2.1.2. Standards

We have chosen JSON as our command object standard format. The reason we have chose this standard for our command objects is because the JSON language integrates with all object oriented languages. JSON is so universal between coding languages because it processes complex objects into strings, which are easy to communicate over sockets.

A JSON object is enclosed by curly brackets '{}'. A JSON array is enclosed by square brackets '[]'. JSON Variables are denoted by the name of the variable and the value of the variable separated by a colon ':'and multiple variables are separated by commas. JSON objects cannot contain methods. JSON objects are able to be nested inside one another but that is outside the need of this project and outside the scope of this explanation.

An **example** JSON object might look something like this:

```
{
Command: “arm drone”,
DronePassKey: “Ab12C1123Dd”
}
```

Before a JSON object can be sent over the socket it will be turned into a byte array. This is because a byte is a byte. This will also ensure that no data will be lost, nor misunderstood based on UTF-8 or conversions of plaintext characters.

Once the JSON object has been serialized into a byte array, the length of the byte array will be the first thing sent over the socket. This will allow the receiver to know the length of the byte array and to know when to quit attempting to read from the socket and to start processing the byte array into a JSON object.

2.3. Technology Considerations

A quadcopter is the prefered technology for recording a dance performance thanks to its ability to move to new positions without having to pivot or overcome obstacles on the ground. The quadcopter uses the same four motors (for the propellers) for its full range of movement. i.e. the altering the proportionate speeds of each motor can cause the quadcopter to raise or lower altitude as well as move forwards, backwards, or to side ways. However, a grounded vehicle would need seperate motors for driving and turning as well as raising or angling the camera.

We chose a raspberry pi 3m for our processing unit since it has built in wifi and bluetooth capabilities. We want both since each contains its pros and cons across different grading criterion.

We have also chosen a few modules for the drone to help determine its state. These include GPS to know its distance from the dancer (who will have his/her own GPS), Barometer to determine altitude, Accelerometer to assist in calculating movement vectors and distance traveled.

2.4. Safety Considerations

The raspberry pi's security is paramount for the safety of the quadcopter. The raspberry pi's server socket will be configured to require a password before connecting. Furthermore, the Quadcopter Cameraman's on-platform software will store a list of approved IP addresses upon compilation. The raspberry pi will immediately reject any connections not associated with one of the stored IP addresses.

Testing the on-platform software require the use of all hardware modules. Inevitably, the software will function in a way we do not intend and the quadcopter could execute movement which could lead to physical damage of the quadcopter hardware. To prevent such catastrophes, the drone's propellers will be removed at all times. The propellers are only to be reinstalled when the following requirements are met.

- 3 or more group members are present
- The off-platform application can connect and disable the autonomous protocol
- The off-platform application can send commands to the on-platform software
- The on-platform software can receive commands and execute them successfully
- All present members agree to take the clear and present risk of reinstalling the propellers

At all times, all group members acknowledge that there is always a risk of damaging the quadcopter hardware.

There are many different online tutorials and lessons that go over every piece of the drone in as much detail as one is willing/able to consume. The massive amount of documentation surrounding quadcopters is one of the benefits with using them. All of our research was done online in conjunction with researching parts to build the drone; to build a drone one must first understand the drone. See references for a complete list of sites we looked at.

2.5. Possible Risks and Risk Management

General Risk

As with all quadcopters there is considerable risk to both the user and quadcopter. These risks will be mitigated by allowing the user to cancel the drone's current tracking target and to land quickly. While also programming in a collision avoidance system to help the quadcopter evaluate when it may collide with something. This prediction will hopefully allow the quadcopter to avoid the object it might have collided with.

If the drone ever loses wireless communication with the android app, it will immediately stabilize it's flight and attempt to re-establish communication with the app. If the drone is unable to re-establish communication then it will proceed to check it's surrounding and attempt a controlled landing.

Component Risk

The overall quality of our quadcopter lies within the individual components being utilized. As such risk should be considered when handling and assembling the individual components. There are risks of fire should the battery receive damage as well as through improper wiring. Our team will also need to ensure that each component is also selected to be compatible with the others especially in terms of voltage and current. Too high of voltage or current input can damage the components and potentially start a fire. Lastly, components should not be directly in contact with each other as the conducting pieces in equipment could touch and create problems.

Damage reduction to the components is essential to the longevity of the quadcopter. While in flight, the components need to be mounted properly so as to not move around. Software should take landings into considerations and work to reduce the chances of hard landings where internal components could receive damage from shock.

When not in operation, the quadcopter needs to be stored safely and securely with the propellers removed. When individual components are removed to be worked on, they also need to be stored securely and safely. This entails means of shock absorption such as bubble wrap being utilized to prevent damage during falls. The battery should also have a charger that can regulate the voltage in each cell so as to not damage the battery while charging.

Potential Challenges

While direct damage and harm to users need to be considered, we also need to anticipate challenges that could hurt the project timeline. In the event a component is rendered inoperable, we are forced to wait weeks for replacements to arrive. We anticipate that our propellers are one of the most vulnerable components on board. As such, we intend to order propellers in bulk so that they can be replaced quickly. There is also the possibility that the order will be delayed or lost. In which case, we should have a separate provider for each component so that we can order from a different company. To ensure ideal delivery time, we plan to order the components we know will be necessary right away, as well as the components that we believe will take a longer shipping time like Hong Kong based distributors.

Feasibility

Based on our research we believe our solution is feasible for our customer's needs. Drone technology is well understood and has plenty of well documented open source technology easily accessible to the public. Construction of a drone to our specifications incredibly feasible. Our solution to make the drone autonomous will integrate easily into the drone's existing design; we will put a Raspberry Pi where the drone normally has a transceiver to receive wireless commands from a controller. The Pi will take in data from various sensors on the drone, calculate commands, and generate a signal that it sends to the reset of the hardware. Our drone should be able to maneuver to keep up with and record the dancers have enough energy to stay aloft through a dance while providing the necessary power to lift the whole package. See 2.6 Proposed Design for detailed information into the design. Open Source Facial and body recognition software is publicly available and not hard to implement. This makes the physical tracking of our project feasible as well.

The difficult part of our design will be flying the drone or rather getting the drone to fly itself. Getting the data from the sensors is not too hard and issuing commands is not difficult either; figuring out what commands to issue based on input from several sensors is difficult. At first we are going to have the drone fly off one sensor likely the sonar. Get to this altitude and remain there. Then we can move on to using the on board accelerometer to determine orientation and create a feedback loop between the sensor and the command. Finally, we'll integrate camera tracking into the loop. Between these three sensors and data regarding how the drone reacts to commands we can build an autonomous flight controller.

To better prepare us for the challenges designing a real time physical control system the group has researched domain relevant material and is planning to leverage recently acquired skills for project success. To build a drone we had to first understand drones. As we researched parts to build the drone to our customer's specifications we became accustomed with a great deal of aeronautical concepts and terms and how they related to drone performance. This research will prove invaluable when tuning the control software. A group member purchased a small quadcopter to observe and better understand how quads handle and the physics surrounding them which has already lead to some useful insight. Furthermore the group has experience with Real Time Systems, Software Development, working with and developing on Raspberry Pis, Embedded Systems, and Electrical Engineering and has oriented the solution around that experience rather than planning to use technology or methods they are unfamiliar with.

2.6. Proposed Design

Hardware

Our proposed design uses an autonomous quadcopter equipped with image recognition software to track dancers during a performance that interfaces with the user through a phone app. Computation will be done on a Pi mounted on the drone which will feed information to a Flight Controller. The Flight Controller is a component normally found on drones it interprets commands from a wireless transceiver, which receives commands from the user/remote control, and translates them into a signal it sends into the ESCs (Electronic Speed Controllers). The ESCs control the amount of power sent to the motors from the battery. Each motor has its own ESC and each ESC has its own connection to the flight controller; our design has four motors thus four ESCs. By having seperate connections the flight controller can control each motor speed individually allowing the drone to execute complex maneuvers such as banks, turns, and flips. In between the battery and the ESCs is a power distribution board which acts as a node to all four ESCs. The drone will use a camera, sonar, and flight controller's on board accelerometer as sensors for flight control. The Pi, camera, and sonar sensors have a seperate power source. **Figure 2.6.1** below shows a conceptual sketch with wiring.

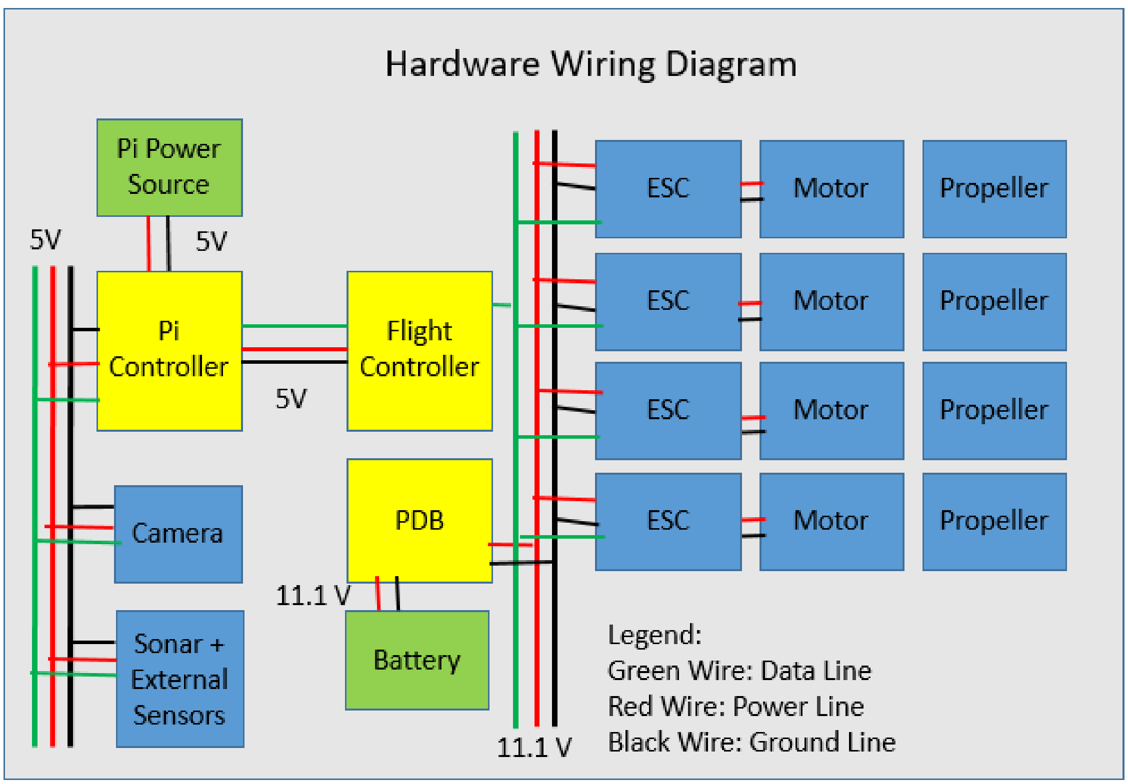

Figure 2.6.1: Hardware Wiring Diagram.

The above diagram shows how each component is linked to each other

Research and Alternatives

We considered using an off-the-shelf drone and programming it to track the dancers. However, it would prose a series of reverse engineering hurdles that made it a worse solution than building our own from scratch with open source components and software. First a drone that could do the work are already pretty expensive without the onboard AI, starting around $500 for the functionality we need. The software and possibly even the hardware would be proprietary and likely not have available documentation unless we could get source code from the company which is unlikely. Interfacing with the existing hardware might not be possible and ripping out the existing boards just to use the shell, power source, engines and camera is cost inefficient at best and still might not fit our needs.

Drone design is already a well understood and documented process with a plethora of both information resources from hobbyist and individual parts for purchase. Diving into the technical side of building a drone was more intensive and time consuming than using a prebuilt drone would have been but the process of researching and understanding the physical system we will be controlling. Going through the research also introduced us to concepts such as lift, torque, ducting the rotors to gain additional lift, how quads handle and maneuver, the difference between rotors and how those differences affect performance. The value of this research can not be understated. By researching and understanding the physical domain we are better able to code autonomous controls aware of the implications the physics have rather than illicting those implications during coding which likely would have been a long and arduous task not likely to produce a good final design. The price for our custom build is $393.72 which is cheaper than the off-the-shelf drone that would still require modifications. Additionally, we're able to customize the physical drone to the final goal rather than working around the physical design of a prebuilt. With all this in mind building a custom drone from scratch is the better solution for this project.

Physical System

For our custom build we decided on a 450mm frame with 10" rotors and 920 kV motors. The larger frame and rotors provide the real estate and power to carry heavier payloads, a smoother platform for camera, and efficiency for longer flights. Research into drone parts indicates that a heavier drone is less likely to jitter during flight from interference and moves much smoother than a lighter one because of the weight. A larger drone has the space for more payload which our solution requires as we will be adding on additional hardware. The 10 inch rotors and 920 kV motors are necessary to lift the whole contraption. The motors will generate 3200 grams of maximum thrust or 800 grams per motor. Research indicates that an ideal thrust to weight ratio for a quad is 2 : 1 meaning that the quad can hover at half throttle/power. Our weight estimates put us at just over 1600 grams.

Software Hub

We'll use a Raspberry Pi as a central hub for sensor, communication with the user from a phone app over bluetooth, and communication with flight controller. We decided on a Pi because its physically light, cheap, and easy to get all of which ideal for use on a drone which flies and might crash. The Pi is also open source and has a great deal of open source solutions available in online repositories.

User Interface

We are using an Android App for communication between the user and the drone. The app works on any Android phone communicates over Bluetooth to the Pi. We are using both personal devices and a phone provided by the school. See 2.6 Proposed Design Software: Drone Communication for more details on application.

The Quadcopter Software required for the Drone will have four hardware systems: Command Systems, Motor Systems, Video Systems, and External Systems. Components are set into their systems based on functionality and compatibility with other required components. The drone will be classified as a 450mm quadcopter meaning the 4 motors are spaced evenly 225mm from the center of the drone.

Command systems will be the brains of the quadcopter and the central hub from which commands are given and processed. The two current components are a raspberry pi and a pi powersource.

Motor Systems are the components that allow the drone to fly. The components involved are the battery, flight controller, power distribution board, electronic speed controllers, motors, and propellers. The flight controller will contain a gyroscope, accelerometer, and altimeter to feed position data to the pi in the command system. The battery will provide 5 minutes of flight time.

Video Systems are what captures and records the video. As of now, we are utilizing a cheap Pi camera for testing purposes. The quality of the camera will be constrained by costs and weight. The quadcopter frame is also a part of this system, and is at a size of 450mm across.

External Systems are all components that are not mounted within the quadcopter. The only equipment in this system are charges for the quadcopter lipo battery. The charger must be a balanced charger meaning it checks the voltages in each individual cell.

Software

It must interface with all of the hardware modules, capture video, process images for target recognition, and compute an adaptable course of action through artificially intelligent algorithms. This will require a number of independent threads to monitor each of the separate theatres. The AI will rely on real time systems to maintain the threads. This will help prevent confusion to the AI and prevent a backlog of data which would potentially lead to a failure. Additionally, all threads will be synchronized to prevent complications.

The AI could potentially be confused if a thread which was created before another thread, finishes after a thread created later. This means that all threads must be synchronized to prevent such complications.

The proposed software component diagram (See figure below) contains the main components for our android and drone software.

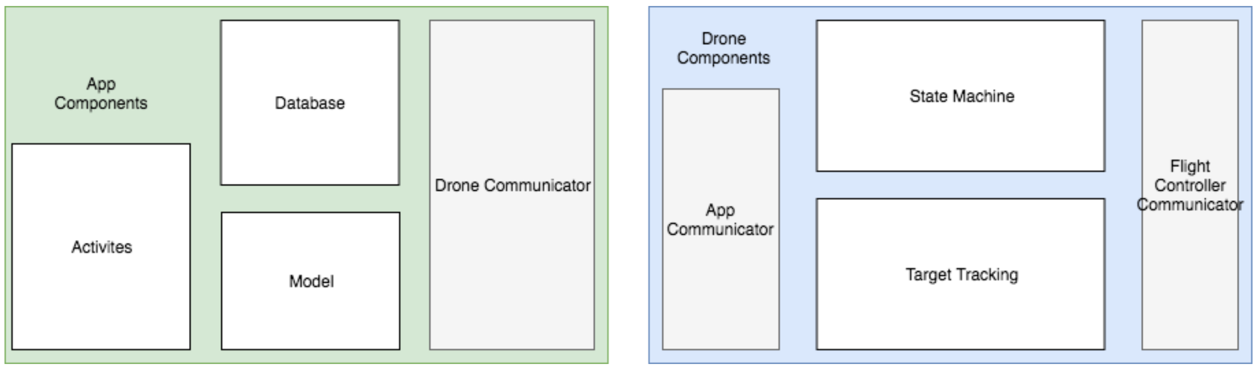

Figure 2.6.2. Software Component Diagram

This is an overview of the proposed software component diagram. The App components consists of the user Activities (UI), the models, databases, and drone communicator.

The software is currently in its early stages. So far it is capable of recognizing a human face (though not yet capable of distinguishing between a set of faces - ie recognizing or identifying a particular individual). The idea of multithreading is also being explored and there currently exists two threads. One is the main thread which captures video and stores it, the second is the processing thread which finds human faces in a given frame of the video. Even in its early stages, the method is working out quite well. The video stream maintains a steady fps and image processing can find a face roughly every ⅕ of a second. Further testing is required to determine if image processing is fast enough to fulfill the responsiveness requirement.

App

Drone Communicator

The software module that will manage wireless communication with the Android App. This communication will handle incoming and outgoing commands with the android application. Once a command has been received it will be decrypted, and parsed for the relevant information. Example of commands that the drone can receive are: Select Target, Arm Drone, Disarm Drone and Upload New Target.

Activities

The Activities component act as the user interface. These activities allow the user to interface the the drone by performing actions such as: Arming the drone, Disarming the drone, Adding new targets to the app database, selecting a target for the drone to follow and the possibility to tell the drone to emergency land.

Model

The models represent the targets and are used to store the information into the database.

Database

The database store the target information. The type of information stored in the database are the targets name, the path to the image, and the hash of the image. The target name is stored so that the user can identify the target. The path to the image is stored in the database because the image itself isn’t stored in the database. The hash of the image is stored in the database because a hash of the image can be used to identify when the user changes the image. This is useful because we only update the targets on the drone when a new target has been uploaded or a current target has had their image changed.

Drone

App Communicator

The software module that will handle communication from the app. Once communication has been received it will then pass the information onto the State Machine component.

Target Tracking

This part of the software will track the target and attempt to predict where the target may appear in the next frame that is processed by the software. These predictions will be used if the target is ever lost from frame or wasn’t able to be identified from one frame to the next.

Target Recognition

The software module that will determine the target that the drone is currently programmed to track. The user will select the desired target from a preloaded selection of targets, which the user is able to upload more targets to.

State Machine

This state machine component handles the various modes of operation. The various states of the drone are: Disarmed, Armed, Tracking, and Landing.

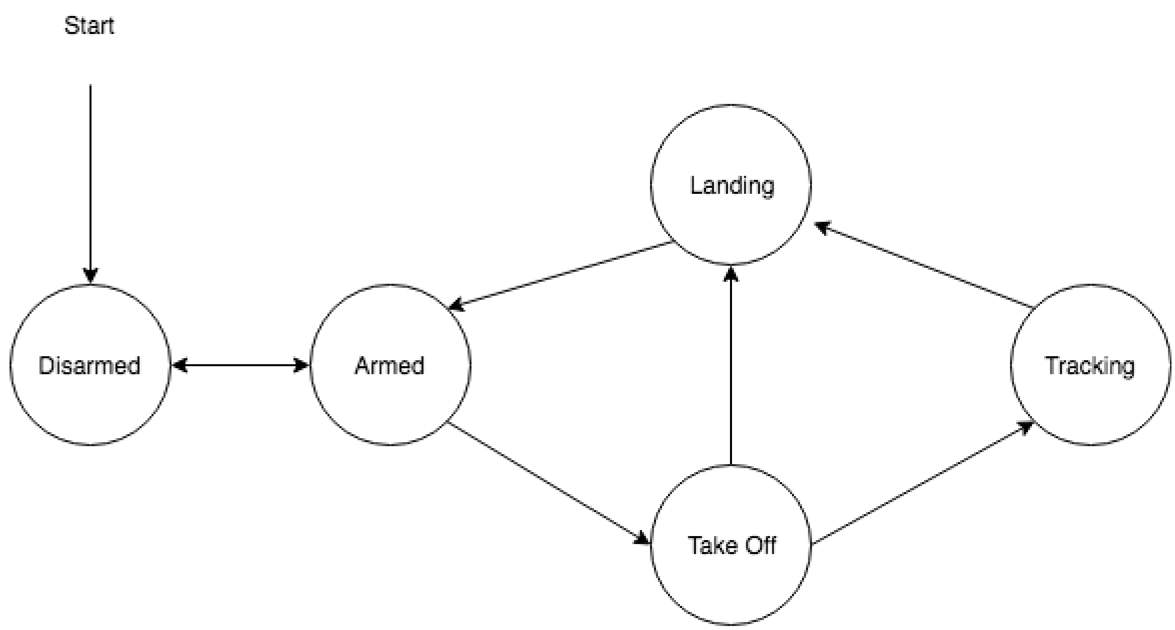

Figure 2.6.3. State Machine Diagram

The flow diagram of the proposed state machine diagram.

The **disarmed** state of the drone describes the state when the drone is sitting on the ground, either before or after a flight. The drone is unable to supply power to the propellers. This state is primarily used for pre-flight preparation and post-flight disassembly. This state can only be entered by the user pressing the “disarm” button on the android app user interface, or when the drone is initially powered on.

The **armed** state of the drone describes the state when the user has finished pre-flight preparation and is ready to select a target for the drone to follow. This state also includes the situation when the drone has recently landed and a new target has been selected to follow. The armed state of the drone can be entered by the user pressing the “arm” button on the android app user interface, or after landing.

The **take off** state of the drone describes the state where the drone already has a target selected and is in the process of lifting itself into the air. The take off state is only entered after the user has selected a target and pressed the take off button on the android app user interface.

The **tracking** state of the drone describes the state where the drone is or has already located it’s target and it attempt to stay within the allotted distance of the target. The tracking state of the drone is only entered after a successful take off has been performed.

The **landing** state of the drone describes the state where the drone is attempting to land. This state can be entered from the tracking state (normal) or the take off state (abnormal). The drone can enter the landing state abnormally for a few reasons, some reasons are: the user pressed the emergency land button, the drone detected an abnormality during take off or flight, the drone detected that the battery is below the allotted threshold.

Flight Controller Communicator

This software module will interact with the hardware flight controller and will be used to tell the hardware flight controller what direction and rotation to apply to the drone. This will cause the hardware flight controller to apply the required electrical output to the motors that are powering the propellers.

2.7. Assessment of Proposed Design

Benefits

The general problem is following a person with a camera. Our solution is to use an autonomous drone that uses image recognition to identify the individual(s), track where they are in relation to the drone, and adjust accordingly. A benefit of our proposed solution is that drone technology is well understood and documented so we will have plenty of sources to draw from throughout the different stages of the project. A drone can have equal or better response time than a human leading to smoother and more reactive video. Our solution allows the user to get dynamic shots without needing another person or in other words they can move around a great deal more than they would with traditional solo filming methods such as a tripod shot (static, camera does not move) or a dolly shot (dynamic but limited and expensive). An autonomous drone offers the user a cost-effective solution to get shots they couldn’t normally get without expensive, specialized equipment and trained staff; it’s a cheap jack-of-all trades solution. For example, a crane rig for which is used for overhead shots cost around $4,700 on the low end without a camera. Stable panning shots require a dolly system or steady cam which runs around $2,400 and $1000 respectively.

Weaknesses

Using a drone does have drawbacks. Flight time will be limited to battery life not making it an ideal solution for repetitive, long takes or long shoots but this can be mitigated with multiple battery packs. Implementing this solution will be difficult. Controlling a drone even in an indoor environment is complicated. There are many different physical factors that need to be considered and accounted for in code, on the platform, or both. For example, before flying the rotors need to be balanced as even a slight imbalance can cause additional vibrations throughout the platform which is bad for filming. An example of using code to compensate for physics is how drones bank and turn. To bank right the drone reduces power the two props on the side it wants to bank towards. This reduces the downward thrust in two ways: first there is that there is less overall lift and second the drone's orientation shifts so some of the downward thrust is used to move horizontally so the drone loses altitude as it banks. Turning along the yaw or rotating in place has a less dramatic effect on altitude as it lowers power to two props diagonally opposite each other by having these props reduce their speed this reduces the compensation they produce for another two props which causes the entire drone to rotate. Like when banking the overall thrust will go down but the drone does not suffer the additional effect of downward thrust lost to horizontal thrust. Since we do not want the drone's altitude to fluctuate while maneuvering we have to increase overall thrust in both cases. But as we end the move the drone must smoothly decrease back to the original thrust to return to a static hover. The code will also have to compensate for the momentum built up during the bank or turn. In short getting the drone to move around in a responsive but smooth fashion will be a significant technical challenge. Keeping the drone stable for the purposes of filming will require a mix between coding nuanced commands and changes to the physical rig to dampen the motion on the camera.

Trade-offs

It's a jack of all trades solution you can get any shot, but it's not specifically built for any one shot. You'll get a better overhead shot with a crane rig and a better pan with a dolly rig. But you can get all of these shots with one rig without hiring additional staff. The current solution focuses exclusively on tracking shots or rather tracking one object throughout the scene which is preferable for dancers but limits the scope of applicability for other fields. Since the job of tracking of the dancers has been automated the user will save on not hiring a cameraman but lose out on the experience and flexibility a cameraman brings. There is also the (ethical) issue of automating a profession that people may grapple with.

Comparison of Similar Systems

Drones currently on the market either use GPS or recognition technology to achieve autonomous tracking. DJI happen to be leaders in this field with products like their Mavic series, Phantom series and Spark series [4]. The Spark Drones use can use either a smartphone or a hand gesture to set operating modes. The Phantom 4's active track is able to even track nonhuman moving objects like cars or trains. The drones can use a combination of gps and active track to follow their targets as well. The biggest drawback to this is the cost of the drones. The cheapest in the line is the Spark at $400 while the Mavic and Phantom are $1500 for the base drone [2]. While the Spark is feasible, the gesture controls create a liability for dancers, as their movement may also trigger a different operating mode than intended. While the packages that these platforms offer are great, a custom built drone will give us much more control to meet our client's needs than a repurposed drone.

2.8. Project Proposed Milestones and Evaluation Criteria

Drone is Built

At this stage, all modules have been installed and wired correctly. On board processing is fully functional and quadcopter is capable of flight with direct human supervision/control.

Base App

The Android app capable of allowing a user to perform the following actions:

- Communicate with drone
- Set drone target tracking
- Initiate drone flight
- Halt drone flight

Module Libraries

A library exist for every module on board the drone. The library contains functions to receive readings from the module and to send commands to it where necessary. These libraries should be extensive enough that the drone can complete a set of instructions provided upon compiling the software. (These instructions are only to be run while blades are removed)

Drone Remote Control

The app and quadcopter can successfully communicate. Successful communication meaning that receive a set of instructions from the base app and complete them.

Single Target Recognition

The quadcopter is able to receive stream input from camera, identify that faces are present in the frame, and determine if any of those faces match the face it is supposed to be following with a high degree of accuracy.

Single Target Following

Drone can recognize a face and human and determine its distance from the target. Drone then repositions itself in the space maintain the preferred distance from the person. Drone also rotates itself to keep the dancer in frame

Multi target following

Drone has completed Milestone "Single Target Following" and is able to perform this functionality on two dancers with one of the two being the primary target used for tracking. Drone will make calculated decisions to allow it to keep both dancers in frame as much as possible. This may require the drone to choose a new preferred distance or altitude.

2.9. Project tracking procedures

The team will present weekly status reports from every member with detailed descriptions of what each individual worked on during that seven day period. All team members will have assigned tasks that come with a delivery date and time. Deadlines are not always met in industry and in academia. In most cases, issues arise and new factors must be taken into consideration. Therefore, task deadlines will be tentative, but task completion is still a priority. Team members will hold one another accountable for their contributions to ensure that one person is not completing a large majority of the workload.

Progress Tracking Measures

Weekly meetings with the client - The team will conduct weekly status meetings with the client and bi-weekly with the adviser. At these meetings, individual team members will come prepared with clarification questions, assumptions, or challenges to present to the client / adviser. The adviser will do their best to provide new insight for the team to overcome any current issues. These meetings provide a foundation for client communication.

Internal Peer Review - At the end of every month (until May), individual team members will spend 10 minutes reviewing their individual progress with the project manager, Zhengdao Wang, to discuss room for improvement. This will provide a foundation for accountability and personal improvement amongst individual team members which will translate to team improvement and project progress.

2.10. Objective of the Task

The Objective of the Task is to produce an additional camera to record dance performances. The Quadcopter Cameraman will assist the existing human cameraman by providing a second and more responsive angle of the performance. Having the second angle be captured by a quadcopter also allows for high angle shots which can be difficult and more expensive for other methods

2.11. Task Approach

Our team will be taking the incremental approach - a series of baby steps, each building off of the steps before it, to keep our progress simple and avoid making huge jumps which require encumbersome amounts of refactoring and engineering.

We will first create versatile libraries extensive enough for us to accomplish every functional requirement. These libraries will largely interface with hardware components and allow communications between a user application and the drone. When these libraries are finished, moving the drone in 3-dimensional space or sending information between user and drone will take nothing more than a quick call to the API. This API will be versatile enough to allow for entire architectural reworkings within higher levels of the software system.

The later phases will involve incrementally working toward our final goals for the actual target tracking/following software for the drone. We will start with simple tasks such as being able to recognize a human within a frame. Later, we will recognize one of the in-frame humans as our target. Then we will create tracking algorithms which allow us to deduce where our target is when identification fails. Once the algorithms can consistently and reliably keep track of our target, we can begin using knowledge of the target's location to move the drone toward, away from, or with the target.

2.12. Expected Results and Validation

We intend to deliver to our client one autonomous drone and an android app to allow our user to interface with the drone.

Drone Expectations

The drone will be a quadcopter capable of flight. This flight capability will be able to automatically handle stabilization, directional flight, taking off and landing. If the drone is able to maintain flight and maintain all previously mentioned capabilities for five minutes or longer, it will be declared a success. Additionally, if for any reason the drone hits a wall, person, item or the floor in a non-graceful and non-programmed manner, it will have failed and improvements will have to be made. While in use the drone will attempt to follow the specified target using these built in flight systems. To test this, we will perform multiple dances in which the drone will have to follow the selected user. If it is unable to and losses the target for too long it will have failed. We will also test its target finding ability by removing the target from view and having the drone run its protocol for finding the user. If it is able to it will be considered a success. Product will be delivered by 05/01/2019

Android App

The android app will be a downloadable software able to be loaded onto android devices only. The android app will perform multiple tasks; allow the user to upload multiple images of the target for maintaining video quality, start/stop the drone, assign the mode of recording and tracking, assign tracking target, command the drone to perform an emergency landing, and receive the recorded video for user viewing. To insure the success and validity of the app, we will test it at distances ranging from 5 feet to 7 meters, well outside expected distances. If the drone maintains proper flight while within assigned range and communications are still active, we will consider the communications a success. If the app crashes for any reason and communication with the drone is lost, the app will be considered a failure and improvements will have to be made. Product will be delivered by 05/01/2019.

2.13. Test Plan

Our minimum flight time is 5 minutes of continuous flight. To test that we can have a successful flight, we need to check each individual component and test the design as a whole. (If all tests are successful, the drone as a whole will be considered successful)

Hardware

Command System tests:

1. Connect the pi to the power supply and have it run for 5 minutes.

Motor Systems tests:

1. Power to all components
2. Run motors individually
    a. Change current to motor to increase speed (Observable change in thrust)
3. Run motors and maintain a hover for 5 minutes
    a. Motors should not increase in thrust unless a command is run making it
4. Hover drone at 2 meters and increase height by 1 meter (altimeter test, accelerometer test)
5. Move drone left and right (speed controllers, accelerometer test)

Video Systems tests:

1. Run camera with drone for 5 minutes

External Systems tests:

1. Batteries run for 5 minutes without using 80% of batteries.
    a. Run drone 6 minutes and 15 seconds successfully without diminishing results.

Software

Pi Software tests:

1. Drone arming and disarming tests
    a. Drone armed
        i. Blades/motors start but does not take off
        ii. All systems operable
    b. Drone disarmed
        i. Checks that drone is landed
            1. If not landed, land
        ii. Blades/motors stop
        iii. All systems non-operable
2. Height change tests
    a. Rise 2 meters (from ground) command
    b. Drop 2 meters command
    c. Hover command
3. Turn tests
    a. Rotate drone 45, 90, 180, 360 degrees both clockwise and counter-clockwise

    b. Rotate drone 360 degrees command and stop command (before reaching 360 degrees)
4. Moving tests
    a. Move drone left and right 1 meter
    b. Move drone left/right a certain amount and stop (before moving set amount)
5. Autonomous flight
    a. Each mode will be tested for desired results
    b. General tests
        i. Will maintain full flight (runs for 5 minutes)
        ii. Successfully follows target (does not lose and cannot re-find target)
            1. Rotation only
            2. Bi-directional only
            3. Omnidirectional
6. Camera face/body recognition tests
    a. Pi receives video images from camera
    b. Find target before dance
    c. Maintain tracking of target throughout flight
        i. If target is lost find target within 10 seconds
        ii. If target is lost send warning to user that target was lost

Android App Tests:

1. Communication stream
    a. App sends command to drone
        i. Arm/Disarm
        ii. Rotate
        iii. Move
        iv. Rise/Fall
    b. Receive acknowledgement from drone
    c. Send mode type to Drone
2. Button tests
    a. Each button correctly sends command to drone
        i. See Communication Stream for expected outcomes for certain button pushes
3. User recognition
    a. App receives information from pi about selected user
    b. App finds user in database

3. Estimated Resources and Project Timeline

3.1. Personnel Effort Requirements

An estimate that physical construction of the drone will take about three hours to six hours depending on how many pitfalls we hit during the build.

A great deal of time will be spent working to make sure the different pieces of hardware communicate properly. Depending on what hardware is talking to what we can end up running anywhere from six to twelve hours per communication link. The five current links are phone to pi, pi to flight controller, flight controller to pi, sonar to flight controller, and camera to pi. With the current estimate of six to twelve hours per link, it will take from 30 to 60 man hours to establish the links.

Code needs to be written to make use to the data being passed through the links. Once written the code needs to be tested first with fed values, then values fed over the link, and then in real life.

3.2. Other Resource Requirements

We will need a number of programs to support our endeavours. We will use Github to backup code as well as do version control and iteration design. Google drive and its corresponding services: Sheets, Docs, Slides, etc. are already in use. We'll be using the labs in Coover and their associated programs and hardware to work on the project.

3.3. Financial Requirements

For the hardware team, there is a financial responsibility to make the client aware of what the expected cost of the system will be. This is an ethical code that, as engineers, the team will follow in being truthful so the client will know if the funds will be available to complete the project asked of.

The overall budget was determined to be $550 to develop the drone. This budget was determined by taking the amount of money each group member deposited as a part of the Senior Design Course registration for both semesters and adding the money together. As such, each member has contributed $55 per semester, and with a team of five, that comes to a total budget of $550 over two semesters. We will also be spending more of the money upfront in the first semester to assemble the drone while the remaining funds serve to either upgrade any necessary systems like our camera or to replace damaged components like our rotor. $400 will be budgeted for first semester, and any funds left unspent will be saved for the following semester with a budget of $150.

We believe that this is a feasible budget for our needs. Kits on Amazon.com show the price for the basic components being under $200. These kits include radios, motors, and frames. We can conclude that we can make a better model that meets our clients needs within our allocated budget of $550. We then conducted research on various components and found the parts that best fit in our function requirements with the costs presented in Table 3.3.1.

A big concern for our team is preventing damage to our quadcopter components. Damage in some components like the battery could easily damage other linked components like the motors or flight controller. These components are also pricier and there is only a certain amount of funds available for replacements. Our team will be ensuring proper safekeeping and handling of each component to prevent damage when not in use. During testing, and flight, we will be implementing software safety features such as an emergency stop mode to ensure that the drone does not damage itself. We also assume that that the rotors will inevitably get damaged and so will purchase them in bulk to have spares available.

Internal Components	Model	Price
Processor	Rspbry Pi 3 Model B	$34.99
PI power	KMASHI External Battery	$10.99
Motor System		
Battery O-2	Gens Ace	$56.05
Flight Controller	ZnDiy-BRY CRIUS All in One Pro	$53.36
Power Distribution Board	Lumenier Mini 4	$11.99
Motors	Gartt4 x 2212	$135.72
ESC	Hobbywing Skywalker	$0.00
Props	Ray Corp Gemfan	$13.99
Video System		
Frame	JRLEC	$16.90
Camera	Fosa USB Camera	$8.99
Gymbal	None	
MicroSD	Kingston 16GB	$5.75
Total Weight		
External Components		
Battery Charger	Passport P1 Mini (DYNC3015)	$44.99
Total Cost		$393.72

Table 3.3.1. - Component Cost Breakdown Table

Indicates every component that was purchased and the cost for that item.

The resources used to find these approximate costs of each item came from all different companies. The process the team went about searching for items started with looking at the Iowa State University surplus website. From there, the focus was placed on looking for low cost items that will be sufficient for the system.

Some items' costs are estimates due to market variations. The hardware components' prices will vary depending on necessary wire lengths and vendors' required minimum order quantity. The total estimated cost for Quadcopter Cameraman is $393.72.

3.4. Project Timeline

Figure 1 is a representation of the ideal timeline of our project. We are currently in the 3rd week of November and we are on track for the BaseApp, and Image Recognition tasks. We are currently behind in the Build Drone task and the Communications task.

We are behind in the Build Drone task because we haven't received our flight controller hardware component, and we didn't anticipate having to build an additional shelf on the drone to support all the hardware we are installing on it. We are currently behind on the Communication tasks because we have been having issues with the android app and git. This roadblock caused us fall behind in this area because it was more important for us to fix this issue than to continue working on the communications.

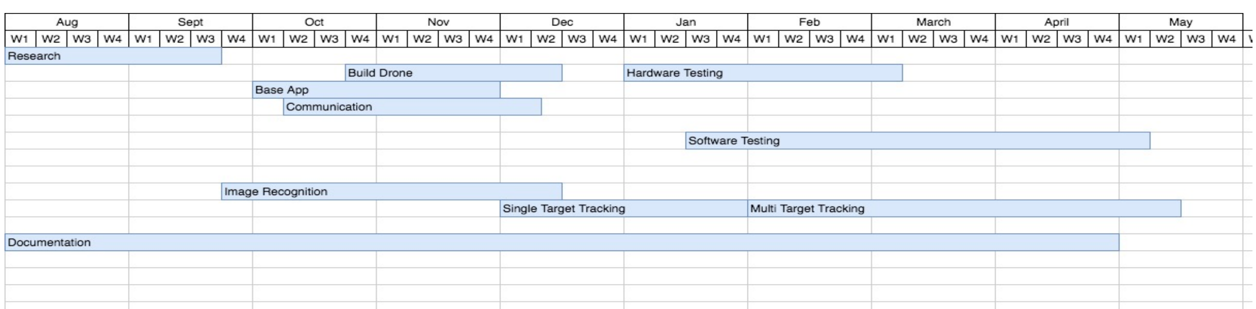

Figure 3.4.1. Project Timeline Estimation

Block-style scheduling table shows what tasks are planned and where they fall on a timeline for the project. The columns are broken down into months, and each month is broken down further into weeks. The blue boxes represent tasks.

The first semester primary goal is assembly of hardware, software modules, and the integration of the two. The secondary goal is to test the software modules and to calibrate the hardware components for our drone.

The second semester's primary goal is to test the hardware and software integration and to test our risk situations to ensure our product is safe for flight.

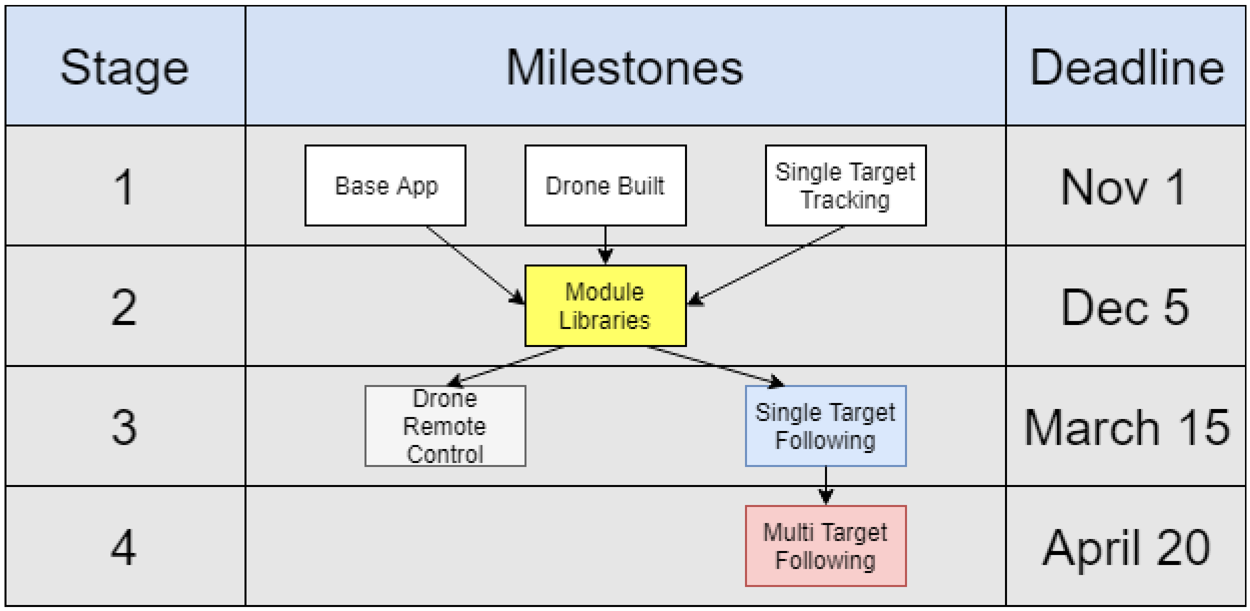

Figure 3.4.2. High Level Project Schedule

The table shows a high level schedule based on milestones' dependencies. Each colored rectangle is a milestone with an arrow showing the milestone which requires its completion before being started. The schedule is derived from that dependency graph.

Stage 1

Stage one is the assembly phase. The foundation for all of the major focal points of are project are done at this stage. The drone, the app, and the technology for tracking targets have no dependencies on each other, so they can be done at the same time.

Stage 2

The module libraries is the only milestone which will be completed at this time, since the next milestones are both dependant on this milestone. This stage will be all about the creation of the module libraries. These libraries will provide a layer of abstraction for the drone and the hardware. Since we have about 10 different hardware modules there will be a heavy workload fitting for an entire team to work on. Each of these hardware modules will need to initialize its respective hardware. It will also need to handle serialized communication with the hardware, in some cases a waveform generator will be needed to communicate with hardware devices.

Stage 3

This is the stage where the android app is finished. Now that the drone can actually move, we can have it send the information to the app as it completes its requested tasks. This requires functionality of both the basic app and the drone's movement. Also dependant on the drone's movement, is the drones ability to track a target and move itself within 3 dimensional space to follow the target

Stage 4

The final stage is only a small adjustment on the 3rd. By the end of the 3rd stage, we should be able to track and follow a single target. Also, most of our use cases will be finished in terms of app-drone communication. Now we only extend that functionality to follow multiple targets rather than a set single target.

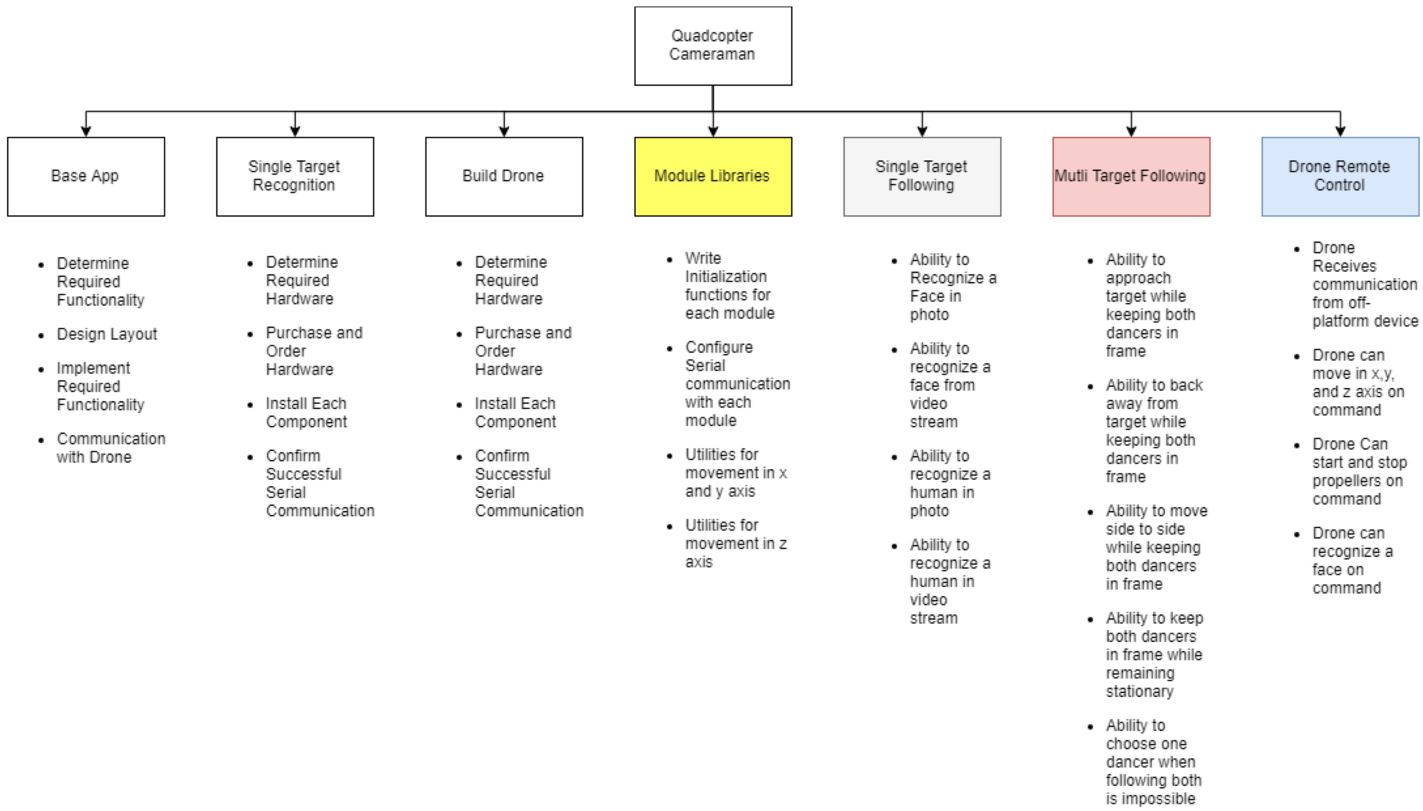

Figure 3.4.3. Work Breakdown Structure

The figure shows a tree-structured break down of the tasks required to be completed and deliver the product with all of its requirements by the end of the year. The rectangles are each of the milestones we identified earlier, the bullet points below detail the minor tasks within the corresponding milestone.

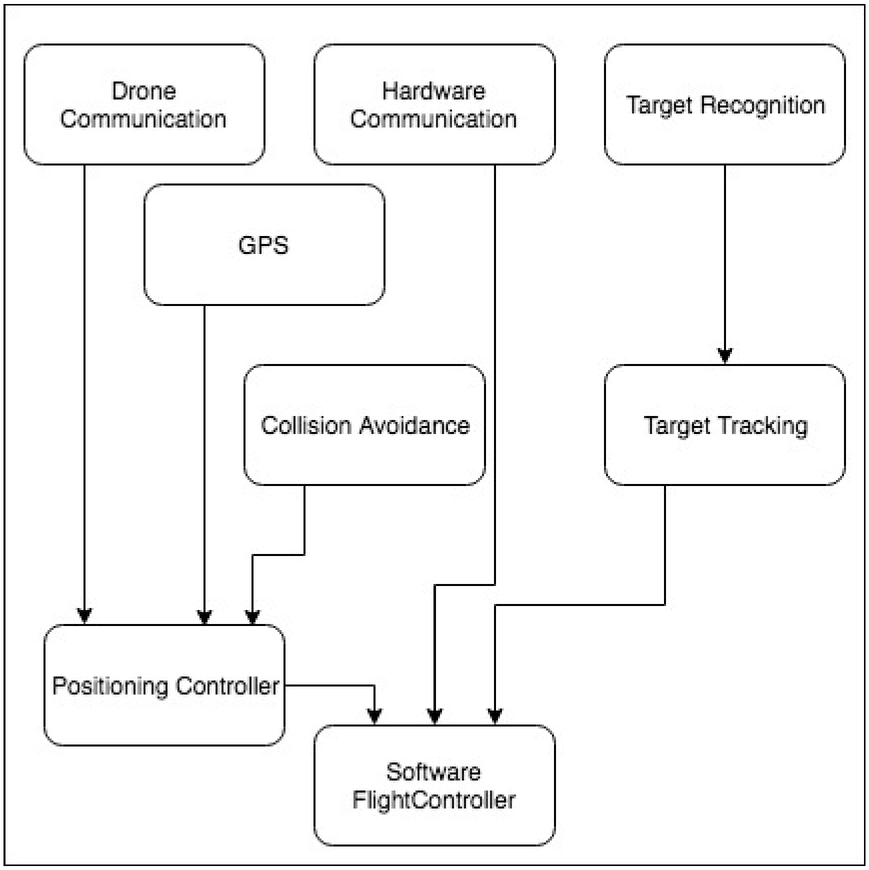

Figure 3.4.4. Software Component Dependency Graph

Is showing the dependency graph for the software components. The Software Flight Controller code is dependant upon the positioning Controller, Target Tracking and the Hardware communication. The Positioning controller is dependant on Drone communication and Collision Avoidance. The Target Tracking is dependant upon Target Recognition. Target Recognition, Drone Communication, Collision Avoidance and Hardware Communication are no dependant upon other software modules.

4. Conclusion

4.1. Conclusion

Our goal for this project is to have a interesting and comprehensive way to record videos of dancers at dance competitions and performances. We would like to improve video taking and quality with a drone that would take out or reduce human error in recording. Our design will include multiple modes of recording and allow for a range or dances that can be recorded including swing, west coast, salsa, and bachata. With facial and body recognition, we will have the ability track dancers and follow seamlessly while recording the performance. We will have many protocols for maintaining and restoring user-drone communications via bluetooth to guarantee safety of the drone and people in the area. In the end, the product's tracking and navigation capabilities will deliver a state-of-the-art and easy to use piece of aerial recording equipment.

4.2. References

Batteries For UAV. [Online]. Available: http://dronesarefun.com/BatteriesForUAV.html. [Accessed: 27-Oct-2018].

"4 X GARTT ML 2212 920KV 230W Brushless Motor with Self Locking Adapter Quadcopter F450 Multirotor Drones-in Parts & Accessories from Toys & Hobbies on Aliexpress.com | Alibaba Group," *aliexpress.com*. [Online]. Available: https://www.aliexpress.com/item/4PCS-GARTT-ML-2212-920KV-230W-Brushless-Motor-For-DJI-Phantom-QuadCopter-F450X525-Multirotor-Drone/32561949556.html. [Accessed: 27-Oct-2018].

DJI Official. (2018). DJI - All Products. [online] Available: https://www.dji.com/products#drones-nav [Accessed 27 Oct. 2018].

F. Corrigan, "12 Best Follow Me Drones And Follow You Technology Reviewed," *DroneZon*, 06-Oct-2018. [Online]. Available: https://www.dronezon.com/drone-reviews/best-follow-me-gps-mode-drone-technology-reviewed/. [Accessed 27 Oct. 2018]

F. Corrigan, "How A Quadcopter Works With Propellers And Motors Explained," *DroneZon*, 06-Oct-2018. [Online]. Available: https://www.dronezon.com/learn-about-drones-quadcopters/how-a-quadcopter-works-with-propellers-and-motors-direction-design-explained/. [Accessed: 27-Oct-2018].

F. Corrigan, “Drone Waypoint GPS Navigation Technology And Uses Explained,” *DroneZon*, 11-Sep-2018. [Online]. Available: https://www.dronezon.com/learn-about-drones-quadcopters/drone-waypoint-gps-navigation-technology-explained/. [Accessed: 27-Oct-2018].

Painless360, “How to select the right motor for your multi-rotor (all types - Tri, Quad, Hex etc.),” *YouTube*, 15-Sep-2013. [Online]. Available: https://www.youtube.com/watch?v=HSQGI6u2DIM. [Accessed: 27-Oct-2018].

“Multirotor Motor Guide,” *RotorDrone*, 30-Oct-2017. [Online]. Available: https://www.rotordronemag.com/guide-multirotor-motors/. [Accessed: 27-Oct-2018].

R. Riot, “Propellers,” *YouTube*, 18-Apr-2016. [Online]. Available: https://www.youtube.com/watch?v=TbAkcKPHqt8&feature=youtu.be. [Accessed: 27-Oct-2018].

“Understanding Kv Ratings,” *RotorDrone*, 27-Jan-2017. [Online]. Available: https://www.rotordronemag.com/understanding-kv-ratings/. [Accessed: 27-Oct-2018].