SoftwareControlledDrone Project

Projects website: https://github.com/Libs1/SoftwareControlledDrone.github.io

Declaration of Authorship

Approved Proposal

Proposal

Proposal for the Software Controlled Drone Project

Prepared by Denis Stepanov and Kevin Libdan
Computer Engineering Technology Student
github.com/libs1/softwarecontrolleddrone.github.io

Executive Summary

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with sensors and actuators for controlling a drone. The database will store information on flight control. The mobile device functionality will include software support for controlling the flight of the drone and will be further detailed in the mobile application proposal. I will be collaborating with the following company/department Future Humber Quadcopter Swarm. In the winter semester I plan to form a group with Kevin Libdan, who is also building similar hardware this term and working on the mobile application with me. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 2 or 3 student group.

Background

The problem solved by project is to control a drone remotely with an application built for android devices. The problem is to recognize the radio signal used to communicate with the drone and make the android application to send proper flight instructions to the drone.

Drone technology is fairly new and it's becoming popular day by day. It has become popular enough so that an organization like Amazon is testing product delivery by drone. But controlling manual or automated flight path of drones is difficult and requires dedicated controllers. Controllers can sometimes be more expensive or non-intuitive to use. In this situation widely popular can make the task more fun, intuitive and widely accessible. Android applications are easily obtainable and can have very intuitive user interface. In addition android devices are ubiquitous. So having proper android application for controlling the drones has very high potential to make drones more popular and user friendly.

I have searched for prior art via Humber's IEEE subscription selecting "My Subscribed Content"[1] and have found and read [2] which provides insight into similar efforts.

The first article discusses how UAVS have been getting a lot of attention due to its low cost of implementation and how an AR drone is being controlled by a motion capture system to follow a moving target. (Gomes, Leal, Oliveira, Cunha, & Revoredo, 2016)

The second article discusses how to collect input information for the controller used for the AR drone. (Barták, Hraško, & Obdržálek, 2014)

The third article discusses estimating UAV systems total ownership cost including hardware components, software design, and operations. (Malone, Apgar, Stukes, & Sterk, 2013)

In the Computer Engineering Technology program we have learned about the following topics from the respective relevant courses:

- Java Docs from CENG 212 Programming Techniques In Java,
- Construction of circuits from CENG 215 Digital And Interfacing Systems,
- Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
- Micro computing from CENG 252 Embedded Systems,
- SQL from CENG 254 Database With Java,
- Web access of databases from CENG 256 Internet Scripting; and,
- Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:
Phase 1 Hardware build.
Phase 2 System integration.
Phase 3 Demonstration to future employers.

Phase 1 Hardware build

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

Phase 2 System integration

The system integration will be completed in the fall term.

Phase 3 Demonstration to future employers

This project will showcase the knowledge and skills that I have learned to potential employers.

The tables below provide rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

Labour Estimates	**Hrs**	**Notes**
Phase 1		
Writing proposal.	9	Tech identification quiz.
Creating project schedule. Initial project team meeting.	9	Proposal due.
Creating budget. Status Meeting.	9	Project Schedule due.
Acquiring components and writing progress report.	9	Budget due.
Mechanical assembly and writing progress report. Status Meeting.	9	Progress Report due (components acquired milestone).
PCB fabrication.	9	Progress Report due (Mechanical Assembly milestone).
Interface wiring, Placard design, Status Meeting.	9	PCB Due (power up milestone).
Preparing for demonstration.	9	Placard due.
Writing progress report and demonstrating project.	9	Progress Report due (Demonstrations at Open House Saturday, November 7, 2015 from 10 a.m. - 2 p.m.).
Editing build video.	9	Peer grading of demonstrations due.

Incorporation of feedback from demonstration and writing progress report. Status Meeting.	9	30 second build video due.
Practice presentations	9	Progress Report due.
1st round of Presentations, Collaborators present.	9	Presentation PowerPoint file due.
2nd round of Presentations	9	Build instructions up due.
Project videos, Status Meeting.	9	30 second script due.
Phase 1 Total	**135**	
Phase 2		
Meet with collaborators	9	Status Meeting
Initial integration.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Meet with collaborators	9	Status Meeting
Incorporation of feedback.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Prepare for demonstration.	9	Progress Report
Complete presentation.	9	Demonstration at Open House Saturday, April 9, 2016 10 a.m. to 2 p.m.
Complete final report. 1st round of Presentations.	9	Presentation PowerPoint file due.
Write video script. 2nd round of Presentations, delivery of project.	9	Final written report including final budget and record of expenditures, covering both this semester and the previous semester.
Project videos.	9	Video script due
Phase 2 Total	**135**	
Phase 3		
Interviews	TBD	
Phase 3 Total	**TBD**	
Material Estimates	**Cost**	**Notes**
Phase 1		
A microcomputer composed of a quad-core Windows 10 IoT core compatible Broadcom BCM2836 SoC with a 900MHz Application ARM Cortex-A7 32 bit RISC v7-A processor core stacked under 1GB of 450MHz SDRAM, 10/100 Mbit/s Ethernet, GPIO, UART, I²C bus, SPI bus, 8 GB of Secure Digital storage, a power supply, and a USB Wi-Fi adaptor.	>$80.00	An example of a retailer: [3].
Peripherals with cables		
Sensors		
Actuators		
Hardware, etc.		
Phase 1 Total	**>$200.00**	
Phase 2		
Materials to improve functionality, fit, and finish of project.		
Phase 2 Total	**TBD**	
Phase 3		
Off campus colocation	<$100.00	An example: [4].
Shipping	*TBD*	

Tax	*TBD*
Duty	*TBD*
Phase 3 Total	**TBD**

Concluding remarks

This proposal presents a plan for providing an IoT solution for *Future Humber Quadcopter Swarm*. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects such as the initiative described by [3]. I request approval of this project.

Table of Contents

Illustrations or Diagrams

1. Introduction

2. Software Requirements Specifications

2.1 Overall Description

2.1.1 Database

The MYSQL database is responsible for storing the users account information and as well as the drone's activity information. The database uses PHP in order for it to connect with the android application and as well, the database will be running on hostinger which is a free hosting website. There are two tables within the database; the first table is the DroneMembers table which contains the user's information and the second table which is the DroneInfo table which contains information of the drone's activity. The DroneMembers table contains four fields which are first name, last name, username and password. The first name and last name fields contain the user's first name and last name. The username field contains the user's username which is used for logging in. The password field contains the user's password which is used when logging in. The DroneInfo field contains two fields which are date and flight duration. The date field is contains the date of when the drone has been flown and the flight duration field contains the user's total flight time. (Developed by Kevin Libdan)

2.1.2 Web Interface

Depending the how long the binding process takes we will develop the web interface. The web interface will be developed to allow users to register an account. The website will also be capable of allowing the users to view their drone's information such the date of when the drone has been flown and as well as the flight duration. The website will handle basic functionality such as viewing the user's drone runtime information. (Developed by Denis Stepanov and Kevin Libdan)

2.1.3 Hardware

The hardware built is used to control the Syma X12s drone. The hardware to control the Syma X12s is an Arduino, a NRF24L01+ transceiver and two analog joysticks. As for now, we are still having trouble trying to bind the Syma X12s drone with our hardware. One of the main resources to helping us find a solution to this problem is Goebish. Goebish is responsible for creating source code for multiple protocols that support the NRF24L01+ After contacting Goebish, we have tried to bind our drone with his source code by modifying it to force select the Syma X12s protocol and as well as modifying the code to bind the drone with the two analog sticks. With multiple attempts to bind the drone with our transceiver and back and forth communication with Goebish, we have concluded that there was not anything wrong with the code itself but the hardware. The protocol we were trying to bind our drone with was the Syma X12 protocol (compatible with X5C-1, X11, X11C) when we had a Syma X12s drone. We were informed that Syma X12s may be using a different protocol than the Syma X12 protocol while the source code does not support the Syma X12s. We were suggested to get a different drone, Eachine H8 mini which is compatible with the code as it has the supported protocol. (This Hardware Section is due to change as soon as we have the drone binding successfully). (Developed by Denis Stepanov and Kevin Libdan)

2.1.4 Application

Software Drone Android Application is the software used to bind with the quadcopter, maneuver it and save data in MYSQL database using the readings from the drone's flight, implemented in one of the classes. Android Application consists of 10 different java classes which support multiple functionalities. Main classes, which would be visible to users, such as 'ControllerActivity','FlightsActivity','LoginActivity','MenuActivity', 'RegisterActivity' and 'PopActivity' use specific layouts assigned to them. Each one of those activities' layouts support phone, tablet, and large tablet (>10") size of the screen. All of visible activities have the function to adjust to both portrait and landscape mode (except 'ControllerActivity', which is always set on landscape to create more user-friendly interface). All of the main classes use custom 'drawables' to make the application more presentable. There are also 5 other classes which are used for better functionality of the application. All of those 5 classes have communication in some sort with database. For example, 'TableData' class is consists of the information about the drone flight (date and flight duration).Moreover, the application adjusts to different resolutions of images and icons for different devices automatically using Android Asset. There is also a support for 4 different languages (English, French, Spanish, and Russian). (Developed by Denis Stepanov)

3. Conclusion

4. Recommendations

5. Bibliography

Barták, R., Hraško, A., & Obdržálek, D. (2014). A controller for autonomous landing of aR.Drone. In *The 26th chinese control and decision conference (2014 cCDC)* (pp. 329–334). https://doi.org/10.1109/CCDC.2014.6852167

Gomes, L. L., Leal, L. P., Oliveira, T. R., Cunha, J. P. V. S. da, & Revoredo, T. C. (2016). Unmanned quadcopter control using a motion capture system. *IEEE Latin America Transactions*, *14*(8), 3606–3613. https://doi.org/10.1109/TLA.2016.7786340

Malone, P., Apgar, H., Stukes, S., & Sterk, S. (2013). Unmanned aerial vehicles unique cost estimating requirements. In *2013 iEEE aerospace conference* (pp. 1–8). https://doi.org/10.1109/AERO.2013.6496852