Software Controlled Drone Project

Team Members: Kevin Libdan & Denis Stepanov

Project's website: https://github.com/Libs1/SoftwareControlledDrone.github.io

Discipline: Computer Engineering Technology

Date of Submission: March 28, 2017

Declaration of Joint Authorship

The authorship of this project is evenly divided between Denis Stepanov and Kevin Libdan. Denis Stepanov is responsible for handling the android application which will be used to control the drone from an android device. He is also responsible for creating the design aspect of the android application (this includes all the layouts, animations, languages e.t.c). Kevin Libdan is responsible for creating the database which will be used to communicate with the android application and store information such as user information and drone information (flight duration, date and location). Denis Stepanov and Kevin Libdan will both be responsible for the hardware aspect. The hardware will allow the Eachine H8 drone be controlled by two analog joysticks connected to an Arduino and an android application which will have two virtual joysticks. Denis Stepanov and Kevin Libdan will also be responsible for the web interface. The web interface will mimic the same functionalities as the android application which will allow the users to register, login and view their drone's flight information.

Approved 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 students in the Computer Engineering Technology program, We will be integrating the knowledge and skills we 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. We will be collaborating with the following company/department Future Humber Quadcopter Swarm. 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 current term in CENG 355 Computer Systems Project as a member of a 2 student group.

Background

The problem solved by this 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" and have found and read three articles 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 winter 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		
Arduino Uno R3	>$30.95	(Arduino) Amazon
SparkFun Transceiver Breakout - nRF24L01	>$29.37	CanadaRobotix
Eachine H8 Mini Quadcopter	>$28.99	(EachineDirect) Amazon
Lithium AA Batteries	>$5.64	CanadianTire (1 pack comes with 4 AA batteries)
Analog Joysticks	>$32.02	Brainy-Bits (2 analog sticks)
Breadboard	>$8.90	(Elenco) Amazon or Parts kit
Pin Header (8-pin) Male	>$0.50	Sparkfun
Jumper wires	>$3.13	(Sodial) Amazon
Phase 1 Total	**>$139.50**	
Phase 2		
Materials to improve functionality, fit, and finish of project.		
Phase 2 Total	**TBD**	
Phase 3		
Off campus colocation	<$100.00	
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. I request approval of this project.

Abstract

This project is aimed to manipulate the same binding signal that the drone used with the stock controller and then being able to control the drone remotely with an application built for android devices. The android application would also allow users to register if they do not have an account, sign in and view their drone's flight information. We started this project by building a piece of hardware that would allow us to communicate with the drone and also being able to fly it. The major components of our hardware are an Arduino Uno R3, an NRF24l01 wireless transceiver, and two analog joysticks. Developing the application, database, and web interface allowed us to utilize with our hardware.

Table of Contents

Illustrations or Diagrams

NRF24L01 Solder Diagram

`![](https://github.com/Libs1/SoftwareControlledDrone.github.io/blob/master/Images/nrf24l01%20solder.png)`

This diagram is a representation of where to solder the pin header (8-pin) into the NRF24L01 transceiver.

Further instructions to do so can be found at 2.2.4 Mechanical Assembly

NRF24L01 To Arduino Hookup

`![](https://github.com/Libs1/SoftwareControlledDrone.github.io/blob/master/Images/nrf24l01%20hookup.png)`

This diagram displays the wiring between the Arduino Uno R3 and the NRF24L01 transceiver.

Further instructions to do so can be found at 2.2.4 Mechanical Assembly

Joysticks To Arduino Hookup

`![](https://github.com/Libs1/SoftwareControlledDrone.github.io/blob/master/Images/joystick%20hookup.png)`

This diagram displays the wiring between the Arduino Uno R3 and the two analog joysticks.

Further instructions to do so can be found at 2.2.4 Mechanical Assembly

1. Introduction

1.1 Purpose

Drone technology is fairly new and it's becoming popular day by day. It has become popular enough that organizations like Amazon are testing product delivery by drones, 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. Android applications are easily obtainable and can have very intuitive user interface and in addition android devices are ubiquitous. For this reason, having a proper android application that would be able to control drones would increase the popularity for drones. The Software Controlled Drone project is designed to manipulate the binding sequence and as well as controlling the Eachine H8 mini quadcopter drone. This project is focused on creating hardware and an android application that will be capable of controlling the drone. This Software Controlled Drone (SCD) project will have a hardware component, an android application, a database and a web interface.

1.2 Scope

The Software Controlled Drone project will be able to control the Eachine H8 mini quadcopter drone. The android application will have the capability of controlling drone remotely by using two virtual analog sticks in the controller activity and allowing the users to view their drone's flight information such as the duration of the flight and as well as the date of which the drone has been flown. The database will contain the user's information such as their first and last name, username and password. The database will also store information of the drone's flight information that will be displayed to the user. The web interface will allow users to review their drone's flight information.

1.3 Targeted Audience Group

Our targeted audience group would be anyone who is interested on flying drones. As drones are increasing in popularity, this project would give users a different feel of control, whether they would want to fly the drone with the physical or virtual joysticks and allow users to keep track of their runtime information.

2. Project Description

2.1 External Interface Requirements

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. Denis Stepanov is responsible for the HTML, CSS and the overall design aspect of the website. Kevin Libdan will be responsible for creating the PHP scripts that will be used to connect to the database when a user registers an account, sign in and view their drone's information. (Developed by Denis Stepanov and Kevin Libdan)

2.1.3 Hardware

The hardware built is used to control the Eachine H8 drone. The hardware to control the Eachine H8 drone is an Arduino, a NRF24L01+ transceiver and two analog joysticks. The hardware is capable of communicating with the drone and being able to control it with the two analog sticks. An OTG cable, micro male USB to Female USB, is used between the Arduino and an android application in order to establish a serial communication. This will allow the Arduino to receive data from the android application which will then be sent to the drone. Kevin Libdan is responsible for acquiring the materials and as well as the assembly of the materials. Denis Stepanov is responsible for creating the Arduino sketch that would be compatible with the hardware.(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)

2.2 Build Instructions

2.2.1 Introduction

In this section of the technical report, there will be information on how to recreate the hardware to control the Eachine H8 Mini Quadcopter with an Arduino and an Android device, which will use an NRF24L01 transceiver to communicate with the drone.

2.2.2 Build of Materials/Budget

Item	Quantity	Total Price(With Tax + Shipping)
Arduino Uno R3	1	$30.95
SparkFun Transceiver Breakout - nRF24L01	1	$29.37
Eachine H8 Mini Quadcopter	1	$28.99
AA Batteries	1 pack(containing 4 batteries)	$5.64
BreadBoard	1	$8.90
Pin Header (8-pin) Male	1	$0.50
Jumper Wires	40 pack	$3.13
Analog Joysticks	2	$36.02

2.2.3 Time Commitment

Task	Time To Complete The Task
Ordering Parts	3 Days(Shipping)
Soldering the pin header to the NRF24L01 Transceiver	20 minutes
Wiring the NRF24L01 transceiver to the Arduino	5 minutes
Wiring the joysticks to the Arduino	5 minutes
Downloading the Arduino IDE	10 minutes(Depending on computer Performance)
Importing the NRF24L01_Multipro library to the Arduino	2 minutes
Running the code	5 minutes

2.2.4 Mechanical Assembly

Step 1: Purchase the required parts

Step 2: Preparing the NRF24L01

Solder the pin header (8-pin) into the pins indicated with a green dot as shown on Figure 1.1. The pin headers will act as legs for the transceiver when you place the transceiver on the breadboard. Soldering the pin headers to the pins of the transceiver may be challenging due to the size of the pins and as well for individuals who are unexperienced in soldering.

Figure 1.1

Step 3: Hooking up the NRF24L01 to the Arduino

%20solder.png” %20solder.png.pdf” %20solder.png.PDF” %20solder.png.eps” %20solder.png.EPS” %20solder.png.mps” %20solder.png.MPS” %20solder.png.ps” %20solder.png.PS” %20solder.png.png” %20solder.png.PNG” %20solder.png.jpg” %20solder.png.JPG” %20solder.png.jpeg” %20solder.png.JPEG” %20solder.png.jp2” %20solder.png.JP2” %20solder.png.jpf” %20solder.png.JPF” %20solder.png.bmp” %20solder.png.BMP” %20solder.png.pict” %20solder.png.PICT” %20solder.png.psd” %20solder.png.PSD” %20solder.png.mac” %20solder.png.MAC” %20solder.png.TGA” %20solder.png.tga” %20solder.png.gif” %20solder.png.GIF” %20solder.png.tif” %20solder.png.TIF” %20solder.png.tiff” %20solder.png.TIFF” %20solder.png” %20solder.png” %20solder.bb” %20solder.png”

Figure 1: nrf24l01 solder.png

Before we make any wiring connections between the Arduino and the NRF24L01 transceiver, make sure the transceiver is sitting on the breadboard properly. When the transceiver is in place, make the following connections between the Arduino and the NRF24L01 transceiver. Figure 2.1 shows how the connections should look when all connections are made.

This is the pin set up for the Arduino to the NRF24L01 Transceiver:

5v -> VCC

GND -> GND

MOSI -> 3

SCK -> 4

CE -> 5

MISO -> A0

CSN -> A1

Figure 2.1

%20hookup.png” %20hookup.png.pdf” %20hookup.png.PDF” %20hookup.png.eps” %20hookup.png.EPS” %20hookup.png.mps” %20hookup.png.MPS” %20hookup.png.ps” %20hookup.png.PS” %20hookup.png.png” %20hookup.png.PNG” %20hookup.png.jpg” %20hookup.png.JPG” %20hookup.png.jpeg” %20hookup.png.JPEG” %20hookup.png.jp2” %20hookup.png.JP2” %20hookup.png.jpf” %20hookup.png.JPF” %20hookup.png.bmp” %20hookup.png.BMP” %20hookup.png.pict” %20hookup.png.PICT” %20hookup.png.psd” %20hookup.png.PSD” %20hookup.png.mac” %20hookup.png.MAC” %20hookup.png.TGA” %20hookup.png.tga” %20hookup.png.gif” %20hookup.png.GIF” %20hookup.png.tif” %20hookup.png.TIF” %20hookup.png.tiff” %20hookup.png.TIFF” %20hookup.png” %20hookup.png” %20hookup.bb” %20hookup.png”

Figure 2: nrf24l01 hookup.png

Step 4: Hooking up the joysticks to the Arduino

Make the following connections to hook up both analog sticks to the Arduino. Figure 3.1 shows the connections made between the two analog joysticks and the Arduino. Whichever joystick you set up to A2 and A3 on the Arduino will be your left joystick as it will control the throttle (up and down movement of the drone) and rudder (left rotate and right rotate of the drone). The other joystick, which will be your right joystick, will control the Aileron (leftward and rightward movement of the drone) and elevator (forward and backward movement of the drone).

Left Joystick to Arduino:

X -> A2

Y-> A3

VCC -> 5v

GND -> GND

Right Joystick to Arduino:

X -> A4

Y -> A5

VCC -> 5v

GND -> GND

Figure 3.1

%20hookup.png" %20hookup.png.pdf" %20hookup.png.PDF" %20hookup.png.eps" %20hookup.png.EPS" %20hookup.png.mps" %20hookup.png.MPS" %20hookup.png.ps" %20hookup.png.PS" %20hookup.png.png" %20hookup.png.PNG" %20hookup.png.jpg" %20hookup.png.JPG" %20hookup.png.jpeg" %20hookup.png.JPEG" %20hookup.png.jp2" %20hookup.png.JP2" %20hookup.png.jpf" %20hookup.png.JPF" %20hookup.png.bmp" %20hookup.png.BMP" %20hookup.png.pict" %20hookup.png.PICT" %20hookup.png.psd" %20hookup.png.PSD" %20hookup.png.mac" %20hookup.png.MAC" %20hookup.png.TGA" %20hookup.png.tga" %20hookup.png.gif" %20hookup.png.GIF" %20hookup.png.tif" %20hookup.png.TIF" %20hookup.png.tiff" %20hookup.png.TIFF" %20hookup.png" %20hookup.png" %20hookup.bb" %20hookup.png"

Figure 3: joystick hookup.png

2.2.5 Power Up

Make sure that all connections are in the appropriate pins and that there are not any loose connections. Once the circuit has been built, plug in the Arduino into the PC and open the Arduino IDE. Plug in the power to the Eachine H8 drone and the LED lights should be blinking in a steady pace. Now that we have the drone powered on, you can download the zip file which contains the Arduino code that will communicate with the drone. You can download it from here. The original code can be found on Goebish's github which is found here. Our code has been modified to be compatible with our two analog joysticks. Go back to your Arduino IDE and open up the "nrf24l01_multipro.ino" file and then upload the sketch to the Arduino. Once the code is uploaded, the drone's LEDs should be blinking rapidly, this indicates that the drone is ready to bind. At this point, moving your left joystick down should bind the drone and the drone's LEDs should be steady.

If you are using the **android application**, make sure to follow these steps in order:

· Ensure that the "nrf24l01_multipro.ino" file is uploaded to the Arduino as the Arduino will execute the most recent uploaded sketch.

· Connect the power to the drone. The LEDs should be blinking at a steady pace.

· Connect the OTG cable between the android device and the Arduino. The LEDS on the drone should now be blinking rapidly.

· Open the SCD app, login and proceed to the controller and then click on start

2.2.6 Unit Testing

Ensure that the drone's LEDs would be solid as it would indicate a successful binding sequence when the left joystick is moved. If the binding sequence was not successful, it would most likely be a hardware problem. It is important to check that all of the connections made between the NRF24L01, joysticks and the Arduino are made properly and secured. Another test is to ensure that the joysticks are controlling the drone as it is supposed to. The throttle (vertical movement) of the left joystick would move the drone up and down; the rudder (horizontal movement) of the left joystick would right and left rotate the drone, the elevator (vertical movement) of the right joystick would move the drone forward and backwards and the Aileron (horizontal movement) of the right joystick would move the drone leftward and rightward.

If you are flying the Eachine H8 drone with the android application, ensure that the drone's LEDS are solid after following the steps in *2.2.5 Power Up*. The left virtual joystick would control the throttle and rudder movements, and the right virtual joystick would control the aileron and elevator movements.

2.2.7 Production Testing

At this final stage of testing, the hardware is capable of controlling the drone independently with the two physical joysticks or with the Android application. The following tests were performed to ensure that every component of the project is functioning as it should be.

1. Ensure that the hardware functionality is working properly independently and also with the android application by referring to 2.2.5 Power Up and 2.2.6 Unit Testing.
2. The android application functions such as login, and register will work based on information in the database. The two virtual joysticks will work as what was mentioned in *2.2.6 Unit Testing*. The drone's flight information is sent to database and displayed on both the "flights" option on the app and on the web interface correctly.

2.3 Project Specifications

2.3.1 System Interface

The hardware that is built by using the Arduino Uno R3, nRF24L01 transceiver and joysticks are capable of functioning independently without the need of the android application. Although the hardware does not depend on the application, the application would only be used to register, sign in and view the drone's flight information when it is used without the hardware. The web interface can also operate without the need of the hardware; however, the purpose of the web interface is to register or view the drone's flight information.

2.3.2 User Interface

The user interface of the android application is user friendly. When launching the application, the main page will consist of two fields for both username and password to be entered and as well as the option to register or login. If the user has successfully entered their username and password they will be brought to the menu page which will consist of two options, one being the controller and the other being the flight activity. When choosing the controller option, the user will be confronted with two virtual joysticks which are capable of controlling the drone. The flight activity option will allow the user to view their drone's flight information such as flight duration and the current date the drone has been flown.

2.3.3 Hardware Interface

The project consists of an Arduino Uno R3, an nRF24 transceiver and two joysticks. The transceiver is used for communicating with the drone and the two analog joysticks are used to control the drone's throttle, rudder, aileron and elevator.

2.3.4 Software Interface

The android application is the software we have designed in order to control the Eachine H8 drone. Both the application and the Arduino sketch are used hand-in-hand, however, the Arduino sketch is capable of functioning without the android application. The web interface was designed to allow users to simply register and view their drone's flight information.

2.4 Project Schedule/Progress Report

2.4.1 Project Schedule

Phase 1

· Creating Project Proposal

Wednesday (9/7/16) – Thursday (9/8/16)

· Creating Budget

Wednesday (9/14/16) – Thursday (9/15/16)

· Acquiring components, Progress Report

Wednesday (9/21/16) – Thursday (9/22/16)

· Mechanical Assembly, Second Progress Report

Wednesday (9/28/16) – Thursday (9/29/16)

· PCB Fabrication

Wednesday (10/5/16) – Thursday (10/6/16)

· Interface wiring, Placard design

Wednesday (10/12/16) – Thursday (10/13/16)

· Preparing demonstration

Wednesday (10/19/16) – Thursday (10/20/16)

· Writing progress report/demo project

Wednesday (10/26/16) – Thursday (10/27/16)

· Edit build video

Wednesday (11/2/16) – Thursday (11/3/16)

· Writing progress report/status meeting

Wednesday (11/9/16) – (11/10/16)

· Practice presentations

Wednesday (11/16/16) – Thursday (11/17/16)

· Conduct Presentations

Wednesday (11/23/16) – Thursday (11/24/16)

· Build instructions

Wednesday (11/30/16) – Thursday (12/1/16)

· Project videos, Status meeting

Wednesday (12/7/16) – Thursday (12/8/16)

Phase 2

· Group meeting

Monday (1/9/17) – Tuesday (1/10/17)

· Initial integration

Monday (1/16/17) – Tuesday (1/17/17)

· Software Requirement Specifications(SRS)

Monday (1/23/17) – Tuesday (1/24/17)

· Progress report

Monday (1/30/17) – Tuesday (1/31/17)

· Project status

Monday (2/6/17) – Tuesday (2/7/17)

· Progress report of independent progress

Monday (2/13/17) – Tuesday (2/14/17)

· Project status

Monday (2/20/17) – Tuesday (2/21/17)

· Progress report/project integration

Monday (2/27/17) – Tuesday (2/28/17)

· Testing

Monday (3/6/17) – Tuesday (3/7/17)

· Project status

Monday (3/13/17) – Tuesday (3/14/17)

· Prepare for demonstration

Monday (3/20/17) – Tuesday (3/21/17)

· Complete presentation

Monday (3/27/17) – Tuesday (3/28/17)

· Complete final report

Monday (4/3/17) – Tuesday (4/4/17)

· Write video script

Monday (4/10/17) – Tuesday (4/11/17)

· Project videos

Monday (4/17/17) – Tuesday (4/18/17)

2.4.2.1 Progress report

Recent project activities:

We have contacted the person who designed the Arduino code for NRF24L01 communication and realised that Syma x12s may be using a different protocol than the Syma x12. It resulted in the Syma x12s' incompatibility with our Arduino sketch. As a group we have decided to purchase a new quadcopter, Eachine H8, which was one of the supported drones for the Arduino sketch. (https://www.amazon.ca/Eachine-Quadcopter-Headless-Drone-Black/dp/B00XHOOWA0).

Current Objectives:

As of today, our primary goal would be to establish communication between Arduino and Android Application.

Problems / Opportunities, Solutions:

For almost a month and a half we had a problem with binding our quadcopter and Arduino. After finding out that Syma X12s might not be compatible with latest version of NRF24L01 breakout board we came up with a new solution. With some help from Goebish (https://github.com/goebish/nrf24_multipro), who has a lot of experience in deviation, we decided to purchase another quadcopter (Eachine H8, as it was stated above). It allowed us to bind the drone with Arduino and control it with the joysticks.

Financial Updates:

Pair of Joysticks for Arduino: $18.01

Eachine H8 mini: $30

All of the prices do not include HST nor Shipping estimates. However, they will be included in the final updated version of Budget Report after we would establish the connection between quadcopter and Android Application.

Additional Links:

https://www.makehardware.com/2016/07/04/how-to-control-your-drone-from-a-computer/https://www.makehardware.com/2016/07/04/how-to-control-your-drone-from-a-computer/

https://brainy-bits.com/tutorials/arduino-joystick-tutorial/https://brainy-bits.com/tutorials/arduino-joystick-tutorial/

2.4.2.2 Progress Report

Recent Project Activities:

As we now have full control over the drone with the two analog joysticks, we are now focusing on manipulating the drone by Android Application. We have also created a website where anybody is able to sign up, log in and view their drone's flight information. During this process, we had created several PHP files in order to display the user's respective flight information. In the drone application, we had modified our controller where it would be more user friendly than our initial controller we had designed. We have also added a small feature within the app where the user is able to delete their existing flight information from the database when the user clicks on the "Delete All" button.

Current Objectives:

Our current and main objective is to establish communication between our Arduino and Android application in order to control our Eachine H8 drone. We are testing 2 ways of manipulating the drone, such as using OTG Cable, which has direct connection between Arduino and Android device, and Bluetooth Module HC-05. The latter has more complications in order to achieve full communication, but it will be our prior way of connection. If the connection won't be established between Arduino and Android device by using the Bluetooth Module, we will be using an OTG cable since it is an easier way to communicate.

Problems/Opportunities, Solutions:

The main problem we are encountering is the communication between the Arduino and our Android application. We have two options in which we can establish the connection, the first option would be to use an OTG cable and establish a serial connection and the second option would be to use Bluetooth by using the HC-05 Bluetooth Module. As the solution, we are testing basic communication through lighting up the LED connected to the Arduino circuit right now.

Financial Updates:

OTG Cable Micro USB male to USB Female: $6.99 with shipping + Tax (Amazon Prime)

OR

HC-05 Bluetooth Module: $18.65 with shipping + Tax (CanadaRobotix)

Additional Links:

https://www.allaboutcircuits.com/projects/communicate-with-your-arduino-through-android/

http://www.instructables.com/id/Android-Bluetooth-Control-LED-Part-2/

2.4.2.3 Progress Report

Dear Kristian,

You can find below our project's integration status report followed by current objectives and goals in merging process:

Software Controlled Drone project includes three main components: main circuit (Arduino Uno with transceiver), software (Android Application) and web interface (website). The main collaboration between significant pieces is currently in progress because we have encountered some obstacles in establishing connection between the quadcopter and Arduino. However, we were able to merge some of our work progress in one and complete testing of several subjects.

Kevin Libdan was managing database structure and connection to online servers on application side, while I developed web interface for users to check their information online with option to sign up. In the end, user is able to sign up/sign in the Android Application and send some fake data (not connected to the drone) to the server's database, while at the same time user can input their credentials to login on website and check flight information, the same 'fake data' he or she just entered in the application.

Other part of the project is merging our software and hardware together. Kevin have worked on communication with OTG cable and Arduino while I tried to establish the connection through Bluetooth Module HC-05. After some research and testing on modules, we have came up with a conclusion that NRF24L01 and Bluetooth Module HC-05 can not work together simultaneously due to frequency interference. For yet unknown fact we have not understood why exactly they are not working but our simplest guess would be because they both set on 2.4 GHZ frequency; therefore, it is not possible establishing a connection without interference. Due to this fact, our next step in integration process will be basic testing using OTG cable and Arduino, such as sending text or lighting up the LED.

Sincerely,

Denis

2.4.2.4 Progress Report

Recent Project Activities:

We have been testing serial communication between our android application and microcontroller in order to control our Eachine H8 drone. We have also been modifying our controller in our android app to establish a connection with our Arduino whenever the two devices are connected with the OTG cable, as well as sending data to the microcontroller whenever the joysticks are moved. With some help from online resources and past projects (https://github.com/rmahenthiran/Micro-Drone-Flight-Control), we were able to control our drone with our android application.

Current Objectives:

Our current objective is to successfully control the drone with our app without the need of holding down the physical joysticks.

Problems/Opportunities, Solutions:

Although we were able to control our drone with our android application, it was not functioning properly as expected. The main problem we are encountering right now is that the android application would only take control of the drone when the physical joysticks initiate the binding sequence first. This would mean that we would have to hold down the "throttle" joystick while we control the drone with the app. The solution for this problem could be removing the spring from the physical joystick which would allow us better testing but still stop us from achieving desired result of binding the drone without using physical joystick. Another solution could be rewriting a binding function in the Android Application, therefore only using the virtual joysticks for initiating the binding sequence.

Another problem we have encountered is the connection with the smartphone that uses USB type C. Since we are using OnePlus Two as one of our testing smartphones, we have tried to establish connection with the drone using OTG cable. The problem is that USB type C adapters are new on the market and not all of them are functioning. We have purchased several adapters/cables but none of them worked. Currently, there are no solutions for this problem, but OnePlus support team suggested using official OnePlus microUSB adapter (link below) and USB-C adapter to establish the connection. When the microUSB adapter arrived it could read the data from any USB stick connected to the Samsung phone which uses microUSB as the main charging port; however, after connecting it to the USB-C adapter and then to OnePlus Two, it would not recognize anything. As of the result, we have purchased one more USB-C adapter (link below) and waiting for it to arrive.

Moreover, as it was mentioned in the last progress report, we have encountered a problem with using Bluetooth module HC-05 and Nordic semiconductor simultaneously. Due to their frequency interference, we could not use them at the same time so it was suggested that we try a different channel on the Bluetooth module. We have researched information about HC-05 and found out that it uses SPP (Serial Port Profile) which is made to send bursts of data between two devices but there are no supportive links on how to use different channels to prevent interference. Therefore, we have eliminated the option of using the Bluetooth HC-05 module and stick to the OTG connection.

Financial Updates:

We have purchased a total of 3 new adapters for OnePlus Two smartphone, but none of them would be included in the project except of the last one acquired(only if test will be successful). Here are links for the adapters we have purchased:

https://www.newegg.ca/Product/Product.aspx?Item=9SIAA0D4C44428&nm_mc=KNC-GoogleAdwordsCA&cm_mmc
GoogleAdwordsCA-_-DSA-_-CategoryPages-_-NA&gclid=Cj0KEQjw-73GBRCC7KODl9zToJMBEiQAj1JgfxBICDaeKat

https://www.amazon.ca/Type-Adapter-EZOPower-Micro-Female/dp/B013IZMZ3I

https://oneplus.net/ca_en/oneplus-otg-cable

Additional Links

https://www.allaboutcircuits.com/projects/communicate-with-your-arduino-through-android/

3. Conclusion

The Software Controlled Drone project will meet all the requirements specified in the technical report. We have created hardware that is capable of controlling and communicating with the Eachine H8 mini drone. The MYSQL database is able to record the user's flight duration, the date of when the drone has been flown and as well as the user's account information such as first and last name, username and password. The android application is capable of allowing users to sign in or sign up for an account if they do not have one and allow users to control the Eachine H8 drone. The android application is also capable of updating the database with the user's personal runtime information and giving the option to users where they will be able to view their flight information within the app. The web interface is capable of allowing users to sign up and sign in to their account and view their drone's flight information which is stored in the database.

Initially, the hardware and software component of the project were going to be merged by means of Bluetooth; however, after some research and testing on modules, we have come up with a conclusion that the NRF24L01 transceiver and Bluetooth Module HC-05 cannot work together simultaneously due to frequency interference.

4. Appendices

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