ORQC: Oculus Rift Quadcopter Controller

Matthew Grayford, Gustavo Gonzalez, Craig Thompson, Gunnar Skotnicki

University of Central Florida, Orlando, Florida, P.O.Box 160000 Orlando, FL 32816, USA

Abstract – **The objective of this project is to develop and test an alternative control scheme for Quadcopter technology in an attempt to create a more immersive experience than is currently available. The goal is to create an affordable product that provides a novel experience. The group chose this project because we wanted to make something entertaining and exciting and were interested in drone technology as well as virtual reality as a growing field.**

I.INTRODUCTION

Virtual reality has caught much light as of late due to technological advances with minimization of computer components. Screens you can place in front of your eyes that allow you to see what does not exist and bring it to life before you. The Oculus Rift has been a forerunner to this endeavor for the past few years followed by Sony's Project Morpheus and a few others. These devices are bringing the digital world ever closer to an explorable reality.

Along with these devices Unmanned Aerial Vehicles have seen great advancement in the hobby world. Quadcopters have become ever more present in videography and everyday style of RC toy for a newer generation of youth. They come with more possibilities and features ranging from racing to long distance flights allowing for many avenues of use and enjoyment.

Another and final factor to choosing this project was that we wanted something appropriate for a reasonable age group to use. The age distribution in the year 2000 for Central Florida leaned heavily for ages in the 10-30 year old range. Retail for stores selling RC and electronics also reflect a large populace of the same age range purchasing games and RC vehicles as well. Combining these two products together only further feeds the market around the items themselves.

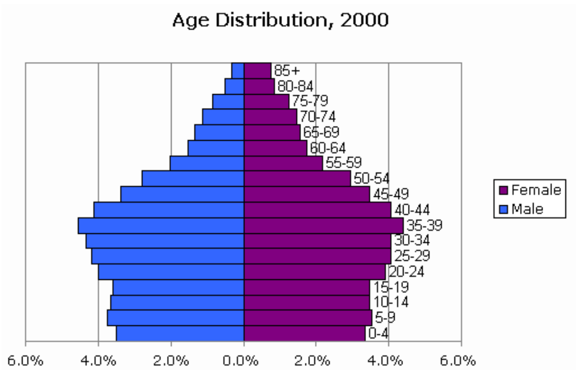

Figure 1 Age Distribution Census Chart Orange County Florida

The rest of this paper describes the key components of the project and their purpose and use within. Key design choices and outcomes are noted as well within the report. Components to be covered include the Oculus rift, the Quadcopter, the Camera for the Quadcopter, and the PCB design choices for the Glove.

II. OCULUS RIFT

There are currently two major models of the Oculus rift. The Development Kit 1(DK1) and Development Kit 2 (DK2.) The most notable improvements between the DK1 and DK2 models however, is the quality of Optics. While the DK1 has 110° FOV (vs 100° FOV for DK2), the DK2 offers a variety of refresh rates rather than the static 60hz as well as a much higher resolution of 960x1080 per eye. As well as better persistence rates. To achieve our highest possible quality we will most likely be looking more closely into the DK2 model for our purposes. To make the best use of the HD capabilities of the Raspberry Pi: Camera module we plan on using, our best approach is to make the most out of the possible resolution rather than having to scale down and lower the quality of our video feed. The quality, and range of freedom in our refresh rates will likely be worth the tradeoff of lessened support, as the DK2 has been available for a shorter time, apps and support software available to it are not as widely available and detailed compared to the DK1, as well as the increased costs associated with the newer model.

III. QUADCOPTER

A few design changes were made throughout the course of prototyping, our original Quadcopter we had planned to use fell short in the features and quality we were looking for. After looking through alternative affordable options we decided on the X525 Model Quadcopter.

Figure 2 HobbyPower X525 Model Quadcopter

The fiberglass frame helps cut down on weight while still being durable. The base of the frame is also sufficiently large enough for us to mount our Flight Controller + Arduino controller combination on top without affecting the balance of the craft. The copter is assembled with 4 XXD 2212 KV1000 brushless motors, 4 30A ESC, 4 1045 Nylon CW CCW Propeller Prop (10” Diameter”). All of this is being powered by the Turnigy 2200mAh 3 Cell 11.1v Li-po Battery Pack. Most of the parts are of a decent quality for their cost. The total build was around $100.

Alongside the copter, the flight controller we will be using is the KK2.15 Multicopter Flight Controller ($25). This flight controller is easily interfaceable with most generic Tx/Rx pairs, which makes it easier to incorporate the Arduino Controller.

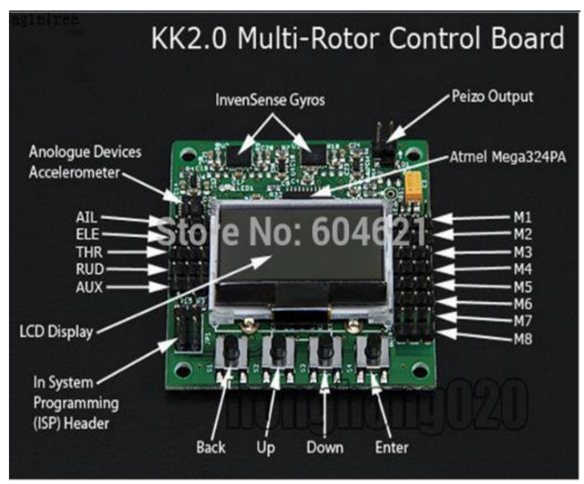

Figure 3 KK 2.0 Multi-Rotor Control Board

This flight controller is exceedingly more affordable than most brands, which often range from $50~100, and has the basic functionality we required. An onboard LCD screen with a menu to navigate the menus helps us set various settings and confirm visually that our signals are being sent and received correctly.

The onboard software handles things like self-leveling and simplifies the level of control we need from our custom Arduino setup. By connecting the signal pins of the throttle, rudder, aileron, and elevator pins directly to the Arduino pwm pins, we can send input into our flight controller to power and control the motors. By having our arduino manage the signals we gain access to a wider range of customization and can better manage the chaotic input medium of our glove mounted controller.

IV. QUADCOPTER CAMERA

The Raspberry Pi Camera module is an add-on built specifically for the Raspberry Pi. It clips on through on board sockets using a CSI interface specifically designed for interfacing with cameras. This tiny 25mm x 20mm x 9mm camera weighs just under 3g giving us a high range of mobility and keeping us that much closer to the minimum weight requirements we desire. This tiny module boasts of capabilities up to 5 Megapixels (2592 x 1944) resolution on static images, and a video recording support that goes up to 1080p30 which is more than enough for the Oculus Rift.

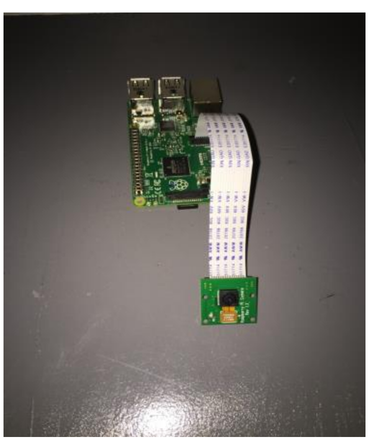

Figure 4 Raspberry Pi Camera With Board

V. GLOVE CONTROLLER

The glove controller itself consists of an Arduino Uno connected to the nrf24L01+ wireless transceiver module (This same setup is located on the Quadcopter as well , and is used as Tx/Rx pair), and the 10 DOF Adafruit Breakout Board (Containing an accelerometer, gyroscope, and barometer). Our design currently is only utilizing the accelerometer, upgrades to incorporate the remaining sensors are planned to help smooth the controls.

A small schematic showing a basic configuration of the wireless module is below.

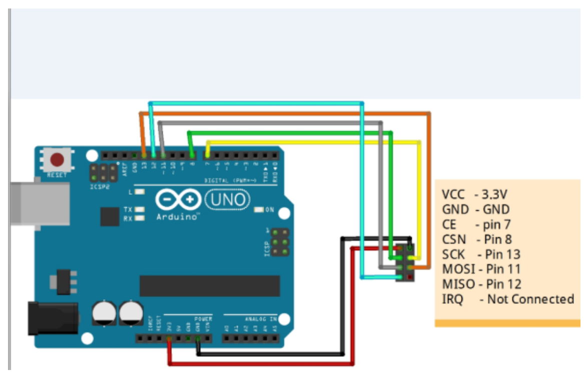

Figure 5 Arduino Wireless Module Within Quad

The rduino and wireless transceiver module will be sending batches of packets containing data through a basic pipeline to the identical module acting as a receiver on the copter, for the 4 input channels that are used to control the quadcopter. These 4 inputs are commonly labeled Throttle, Aileron, Elevator, and Rudder. An optional auxiliary input channel is available to access certain functionality provided by the flight controller software should we require it.

The accelerometer measurements are by default in m/s^2 in the x, y, z axis's. Changes in the measurements are scaled down into a simpler scale of values between 0~255 (0

~124 and 126~255 are negative and positive values respectively) and mapped into one of the 4 input channels (THR, AIL, RUD, ELE) to be sent to the receiver, The receiver rduino then takes an average of the batch to determine which input to feed into the flight controller, before taking a slight delay before checking for any newer packets.

This is done to ensure that the copter is not constantly changing motor speeds from every slight motion of the hand, while this helps smooth flight in general it comes at the cost of a slight delay in commands but this is unfortunately unavoidable due to the nature of our hand controller relative to more common mediums such as joysticks that have natural static positions.. If no signal is received after a certain period of time, or the shutdown signal is sent the copter will slowly reduce throttle and land at its current spot

VI. DESIGN

This project contains a complicated system with many components that must communicate with each other and function smoothly while still leaving room for future improvements and features that we discover and may wish to include through the development process. We have to find a way to have all components of the project work together. We need the quadcopter to work with the WiFi adapter; the WiFi adapter to work with the host machine; the host machine to work with the PCB glove; the PCB glove to work with the quadcopter. This is a grossly oversimplified cycle that needs to work in our project. However, at its core, our system will consist of mostly “stand-alone” modules (The oculus, the host, and the glove) that can function on their own.

We want to make sure the copter can handle staying in the air should it lose signal from our Hand controller, and that the Host machine has fail safes should it lose connection with our video stream and ensure a speedy reconnection. Having them be able to function semi autonomously will ensure that we avoid some minor issues like our copter crashing often during testing and such and make our lives much easier. It is our task to ensure that the Rift ,Copter, and Controller can function together and consistently. Below is a visual representation of the quadcopter communicating with the Raspberry Pi which communicates with the microcontroller on the Arduino

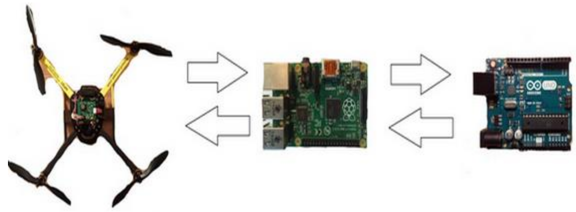

Figure 6 Quadcopter camera streams to Raspberry Pi which communicates with microcontroller and vise versa.

One of the largest challenges that our group faced was working with the Oculus Rift within a Linux environment. This challenge existed due to Oculus Rift dropping their Linux support in April 2015, this initially made it difficult to find the proper drivers for the Oculus. After finding archived tarballs of the Oculus SDK we found that the Makefile was for x86 systems and would not work with our ARM Raspberry Pis. We then made the decision to try to utilize the Oculus as a display, since we had acquired the Oculus prior to our knowledge of the Linux support withdrawal. We felt that the Oculus Rift would serve as a proof of concept, since within the coming year both Sony and HTC are introducing their own VR systems with more advanced hardware and Linux support.

VII. GLOVE PCB

The controller of the drone will be the Glove controller. It will allow the user to manipulate the drone with his/her hand with movement of the hand in multidirectional motions. Utilizing a gyroscope and accelerometer we will be able to accurately depict the motion of the hand and an intended direction for the quadcopter to move. The intent and purpose of this device is to have a Joystick like feel that is more immersive and responsive in nature than a joystick can offer.

Further implementation of the glove may allow the user to utilize other interfaces rather than just the drone like flight simulator applications or other games. This will ultimately depend on the driver software developed for it and the connection point to the desired host machine.

The goal of the glove is to make an intuitive controller that the user will be able to use without much training. Because of the possible goal of integrating the quadcopter-Oculus Rift combination in a laser tag environment, there will be a limited amount of time for each user and there will be multiple users wanting to play with the product. Thus, the easier it is for the user to learn and start playing with the product, the more appealing it will be thus creating more customers and more users. We also want the glove to be as natural and non-juxtapose to using the Oculus Rift as possible. We want the users to be able to feel as if they are flying the quadcopter, themselves, as opposed to having a screen close to their eyes while still using a conventional controller. We want it to be a seamless experience as opposed to a flashy activity.

Ideal functionality of the glove is to have it accurately represent the movements of the user with the movements of the quadcopter. So as the user raises their hand, the quadcopter will raise from the ground. As the user turns their hand, the quadcopter will turn. And so on. There will also ideally be buttons that act as shortcuts to taring. When the user places their hand on the ground and hits the “tare” button, there will be an automatic reset so that the glove’s RC component will translate as the glove being on the floor and thus the motors of the quadcopter will be off. There will also be buttons for setting power distribution amongst the motors in case there will be listing by the quadcopter. What we mean by “listing” is the quadcopter moving to the left, right, forward, or backward unwantedly. This is a common occurrence with quadcopters and thus buttons to correct such actions would be preferable.

Prototyping of the Glove PCB consisted of using the Arduino development board equipped with the ATMega328P microcontroller to communicate with the Adafruit 10-DOF breakout board and the nRF24L transceiver board. Adafruit’s 10-DOF will be the tool used to measure the elevation, tilt, and speed of the glove. The nRF24L will be the tool used to communicate the data being received from the 10-DOF to the receiving nRF24L that is integrated into the quadcopter.

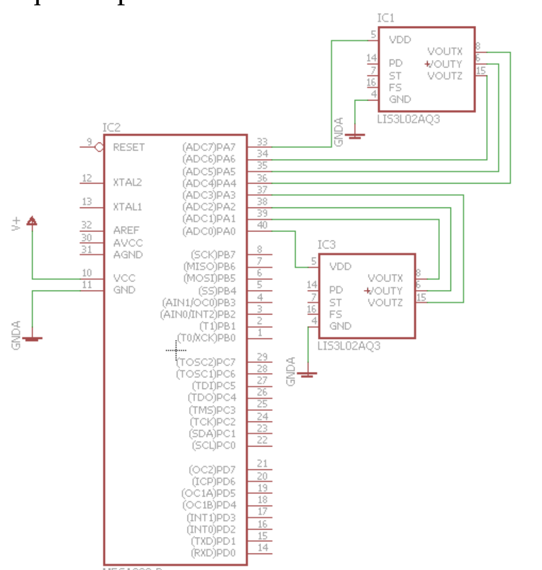

Figure 7 Initial Prototype of Glove Controller

Adafruit's 10-Degrees of Freedom board consists of a gyroscope, accelerometer, and barometer. Measuring the pitch and yaw of the hand will be done with the gyroscope. This will measure the amount of leaning done by the user's hand. The accelerometer will be used to measure the speed and direction of the user's hand. The quadcopter will react accordingly with as swift or as slow a movement as the user wishes to communicate with their hand. The barometer was an extra component that was saved as a possibility of measuring hand elevation, but was ultimately not used.

Choosing the nRF24L transceiver was based both off of the general range and robustness of the device and the amount of user friendly software and tutorials that were available to us. Prototyping has demonstrated that the signal can be easily sent and received from two operating nRF24L boards attached to Arduino development boards from across the Senior Design Lab. This will work perfectly given our possible goal of having drone laser fights.

The PCB will be designed using the free version of Eagle. Although the free version of Eagle does not support several layers of PCB, we will only be needing two. In order to have the ATMega328P communicate with a USB connection for debugging and possible software updates, a FTDI will be used to communicate between the two. Using the SparkFun FTDI Basic Breakout board and the SparkFun Arduino Pro Mini 328 as bare-boned essentials for having the components working, we designed the PCB with these open-sourced projects in mind.

SparkFun's FTDI Basic Breakout uses a micro USB port in their design. In favor of keeping the PCB as familiar as possible, we opted to replace the micro USB port with the same USB B port used by the Arduino development board. The 3.3V or 5V operating voltage of the FTDI Basic Breakout board assures us that given the same source, the replacement USB component will work well.

Similarly to using a familiar USB port, we are opting to replace SparkFun's ATMega328 choice with the Arduino development board's ATMega328P. The voltage input will be the same, so changing won't be a problem.

Lastly, we connected the freshly made FTDI and ATMega328P combination to our 10-DOF and nRF24L breakout boards. The decision to do this was made largely for the sake of time, and somewhat for the limitations with Eagle. Given that we would try to keep the board to a certain set of width and length dimensions, more layers would have had to be added. Our current license of Eagle would not allow this, so we continued with the intention of integrating the pre-made breakout boards into our design.

VIII. INTEGRATION

The individual modules of our project are for the most part independent of each other. This allows for easier testing of each component to ensure things are working as expected. The quadcopter simply serves as the base for which to mount our controller module and the camera module. The camera system module and control system modules are both wireless and can be removed from the copter and replaced with similar alternatives. The controller in particular can be simulated by another arduino/PC combination and sending manual controls.

Our integration was a very gradual process, as we planned from the start to design three systems that were not intertwined but could still benefit from one another. This allowed for our group members to work at maximum efficiency and complete their parts of the project without a fear of complex integration. The camera system's integration was a simple mounting process to the quadcopter, since the camera system operates completely separate from the rest of the quadcopter. The integration of the glove and the quadcopter was a more complex process, where the controller was integrated to a computer first in order to check and compare values. Once the controller was verified we fed the data straight to the quadcopter, allowing for a straightforward integration process after the controller was verified.

IX. CONCLUSION

The project's total success relies the full integration of each setup coming together to complete the system. If the Oculus Rift works but the glove does not, then we are left with a quadcopter that needs to be controlled via a regular RC controller. This will lose the objective of integration and complexity is added that the user now has to learn. If the glove works but the Oculus Rift does not, then nothing but a new controller is made and no steps towards new, exciting innovations have been made. From creation and testing we have managed to group all three components together without a disconnect from each component. With this new innovation, we hope to push the idea of integrating more real life items into a more expandable digital environment.

ACKNOWLEDGEMENTS

Many thanks to the supporters of our project, be they parents, friends, teachers, or co-workers. They provided not only tips and tricks for managing the project as a whole but also tips for software like EAGLE and KiCad.

References

Oculus Guidelines Website
http://static.oculus.com/documents/health-and-safety-warnings.pdf
Census Figure Website
http://www.censusscope.org/us/s12/c95/chart_age.html
QuadCopter Recommendations Website
http://knowbeforeyoufly.org/for-recreational-users/
WiiController PCB mapping
http://www.multiwii.com/
Arduino Pilot PCB
http://copter.ardupilot.com/
Linux Support drop
http://thenextweb.com/dd/2015/05/15/oculus-rift-drops-mac-and-linux-targets-windows-pcs-only/
Oculus Rift to raspberry pi website
http://torkeldanielsson.se/live-twocamera-video-stream-from-raspberry-pi-to-oculus-rift

Gustavo was born April 23rd, 1993 in Caracas, Venezuela. Moved to Florida from a young age, and has held an interest in technology from a young age. Graduated from Freedom High School in Orlando, Florida, and moved on to further his career at UCF as a Computer Engineering student, and has spent his entire college career at UCF. Focus on the project is coding the Rx/Tx

modules to control the copter as well as glove controller software.

Craig is an Electrical Engineering student who spent his entire college career at UCF. Interning for a contracting company. His focus on the project is to keep the quadcopter running with the additional components and to help with the construction and operation of the Glove.

Matthew Grayford is a Computer Engineering student. He has had 2 years of study at Embry Riddle Aeronautical University and 3 years at UCF. Currently he works as an independent contractor for Patient Point out of Ohio. He operates as their System Administrator for JIRA and Airwatch. His major parts within the project were assisting Craig with creation and setup of the Quadcopter and Glove Components as well as technical document creator.

Gunnar Skotnicki is a Computer Engineer student focusing on software design and cyber-security. Currently he has an internship with AVT Simulation working with their Information Assurance team to make sure systems are secure. Within the scope of the project he maintained the raspberry pi environments, configuring the camera, Oculus Rift, network implementation, and general system security.