

Smartphone Controlled Drone (Smone)

M A Abdul Rahman¹, Khallikkunaisa², Nagma Feroz³, Rajat C⁴, Md Muzakki⁵

^{1.} CSE., HKBK College Of Engineering, Bangalore, India , abtube7@gmail.com (Student)

^{2.} CSE., HKBK College Of Engineering, Bangalore, India, khallikkunaisa.cs@hkbk.edu.in (Assoc.Professor)

^{3.} CSE., HKBK College Of Engineering, Bangalore, India, nagmathebornfire@gmail.com (Student)

^{4.} CSE., HKBK College Of Engineering, Bangalore, India , rajatshekar06@gmail.com (Student)

^{5.} CSE., HKBK College Of Engineering, Bangalore, India, muzakki.mohd@gmail.com (Student)

Abstract: *A quadrotor helicopter or quadrotor is a multirotor helicopter that is lifted and propelled by four rotors. Quadcopters are classified as rotorcraft, as opposed to fixed-wing aircraft, because their lift is generated by a set of rotors (vertically oriented propellers). Our drone will be controlled with a smartphone application. The quadcopter will have one security camera and the camera will be the fpv live camera which will transmit the footage to the receiver. And even the camera output will be sent to the smartphone.*

Keywords: *Quadcopter, Vertically Oriented Propellers, Transmitter and Receiver, ESP8266 (Wi-Fi microchip), FPV (first person view), Flight controller*

I. INTRODUCTION

The purpose of the “Smone” project was to create a quadcopter that can be controlled by user input sent over the phone’s Wi-Fi connection or 4G internet connections. Furthermore, the purpose was also to be able to receive live video feedback over the internet connection, thus making the drone an inexpensive option compared to other, equivalent drones that might cost thousands of dollars [1].

A. Project Goals

- Allow a user to remotely control and fly a quadcopter.
- Allow a user to receive feedback via the phone.
- Create an Android and PC app that allow for future development.

B. Outcomes

- Future development through the use of Android java endpoints—these endpoints are software packages that include features and functionality.
- Demonstration of Computer Engineering related skills.

C. Deliverables

- Working microcontroller code that receives data from the Android phone to control drone motors.
- Functional Android app that can control the microcontroller and receive data wirelessly from a PC.

- A PC java application that remotely controls the drone from any computer.

II. OVERVIEW

The “Smone” project utilizes a combination of hardware and software that is relevant to a computer engineering senior project. The project includes the Java PC program, the Android java app, the MSP430 Microcontroller, the MSP430 microcontroller C code, the electronic speed control modules (ESC), a 3000 mAh Li-po. The ESCs regulate the three phase AC signal being sent to the drone motors, which has sizeable efficiency advantages compared to a DC motor.

A. Marketing Requirements

- The device shall be remotely controlled.
- The device shall be easily controlled.
- The device shall be able to send back live video feedback.
- The device shall be able to fly.
- The device shall be controlled with keyboard controls.

B. Engineering Requirements

- The system shall have an Android application.
- It shall have the ability to be a TCP client and connect to a TCP server.
- It shall have the ability to be a TCP client and connect to a TCP server.
- It shall be able to have multiple threads that handle sending data, receiving data
- Simultaneously, while also communicating with the microcontroller in real time.
- It shall have the ability to reconnect with the server if the connection is lost.
- It shall have the ability to take pictures remotely and store them on the data storage.
- It shall be able to connect and send data over UART to the microcontroller.
- It shall revert to an idle sleep mode when not doing anything.

- It shall have the ability to send back live video feedback.
- It shall log diagnostic data to the device storage in the event of a crash.
- The system shall have a microcontroller.
- It will run at a 16 MHz clock and run at 9600 baud.
- It shall output four changeable PWM signals at 400Hz that can all run at different
- Widths that rely on the same timer.
- It shall go into low power mode two when not interrupted.
- The device shall have an emergency landing function in the event that the USB connection is lost.

III. CRITERIA

The engineering requirements above will provide direction for the development of the Android drone app. Here are the most important characteristics and attributes of the app with highest priority first

A. Stable and Controllable flight

The drone must be able to achieve stable flight and be able to not only take off, but also land and move in multiple directions.

B. Flight Duration

The drone should be able to fly for at least a few minutes and not crash during that time.

C. Autonomous Control Functions

The drone programs must have autonomous software functions that automatically do certain tasks such as takeoff and landing.

D. User Interface Friendliness

User friendliness is a function of how easy it is for the user to control the drone from the PC. The keystroke input and connection process must be simple and smooth. Once the device is connected the user should only have to connect the two devices over the network and then only have to focus on controlling the drone. Getting video feedback is also nice for user friendliness so they can see where the drone is going.

E. Fast and Reliable Wireless Control

The drone must be controllable in real time and therefore the system must prioritize getting the commands to the microcontroller and reconnecting the Wi-Fi if it is disconnected.

F. Battery Duration

A lengthy battery lifetime is important for giving the drone the most airtime possible. The Android phone must also last at least as long as the battery lifetime so that the microcontroller doesn't lose connection due to a low phone battery.

G. Future Development:

The Android app must be written in a maintainable way so that future additions are simple to add. The app needs to be setup so that endpoints can be added without major difficulty.

IV. DESIGN AND JUSTIFICATION

Designing the Android Controlled Drone was based on a process of identifying the most important functionality that the drone would need to have; the drone was built around the fact that it would need to be able to lift a decent amount of weight. The drone parts were selected for cost and simplicity due to the fact that the software was going to be a significant challenge and lowering costs was one of the primary goals for this project. Selecting which hardware would fulfill these requirements was based on creating spreadsheets where important attributes like weight, cost, complexity, and others could be easily compared [5].

After plugging in several sets of data into the equation and spreadsheets, while trying different combinations of compatible hardware, the hardware was chosen based on the lowest comparison coefficient. This hardware set contained the drone kit, the microcontroller, the Android phone, the ESCs, the motors, and the batteries. As a side note, the selection for the phone was a personal phone since cost was zero and cost is an important factor.

Furthermore, the Android application needed a more advanced software flow diagram because the Android application needed threads for receiving from the server, transmitting to the server, running the GUI, writing serial data to the microcontroller, and taking photos and videos; the threads were essential because the app needed good real time performance.

The Android app[3][4] required a significant amount of planning and development, especially with how to get it to work with the microcontroller—it was not an easy task to plan out how to write serial data. Moreover, the USB software UART was developed by looking at examples online and testing various libraries and drivers.

In addition, the Primus V3R Flight Controller required a lot of coding because there were challenging performance constraints on the microcontroller; it had to receive data over UART and simultaneously handle four PWM outputs. Therefore, designing how the interrupts would operate was essential for making the microcontroller run as fast as possible. One problem that had to be considered, for example, was the fact that making the UART interrupts service routine too long would interfere with the PWM timer. This would create fluctuations in the PWM output and ultimately make it inconsistent whenever new input from UART was received—the motors would behave unexpectedly and lead to instability. The fluctuations were also verified with an oscilloscope. In effect, any kind of fluctuation in the PWM output could send the drone into a spiral. Overall, a large portion of this development was accomplished through trial and error and experimentation due to a large number

of unpredictable problems. The PWM output had to be calibrated for each motor.

V. SYSTEM ARCHITECTURE

The project architecture can be modeled with a top level diagram that shows how all the components of the architecture are connected. This includes the network, USB, motor, and ESC connections. Because this is the general architecture, it does not cover the details of how the applications specifically operate, although the hardware and software sections cover this thoroughly.

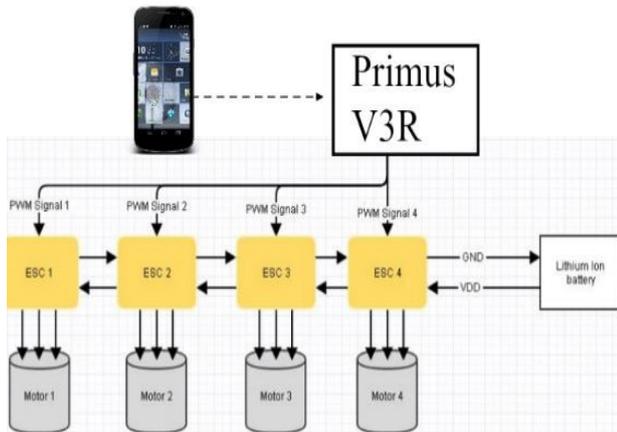


Fig 1. High Level Connectivity Flow Chart

The general system architecture is fairly comprehensible, but relies on significantly more complicated software. The software on the Android phone abstracts most of the complexity away.

The hardware diagrams for the project includes a diagram of the PRIMUS V3R connect as well as the layout for the ESCs. In Figure 2 below, the hardware connections and pins are labelled specifically.

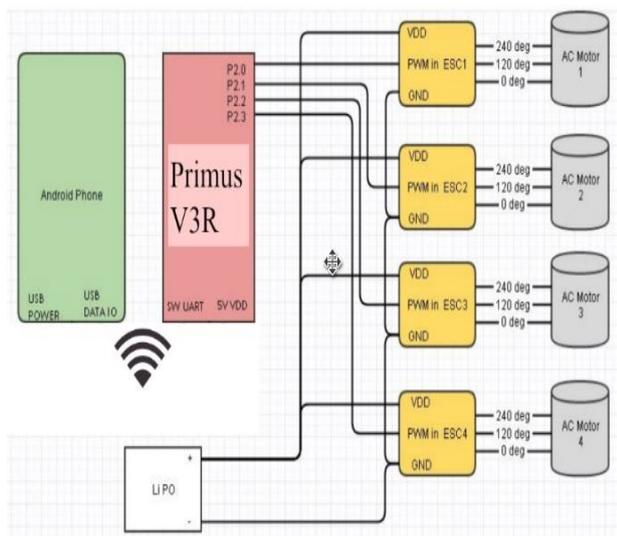


Fig 2. Hardware pin connections diagram

The key purpose of the ESCs, as was indicated by research, is to convert the PWM signal, in an efficient way, to a three phase power output that can also protect against current spikes and regulate voltages in a safe way; in this case, safety refers to protecting the drone against crashing in flight and thus protecting people. Rapid changes in voltages could cause the drone to lose stability. The spikes could be caused by changes in battery temperature or a fast change in motor speeds. Furthermore, the AC motors spin dangerously fast, and so protecting against spikes, voltage changes, and fluctuations is important for safety purposes, especially if the drone is in flight respectively.

VI. HARDWARE COMPONENTS OVERVIEW

A. Primus V3R Flight Controller

The Primus uses a 5V input and supplies a PWM output on pin 2.0, 2.1, 2.2, 2.3 at 3.7V. In addition, it runs at its fastest speed of 16 MHz to improve control of the ESCs. It runs on a Launchpad and is supplied 5V over a USB input. Lastly, it utilizes 4mA of current during maximum draw, at 16 MHz clock speed, and 230 uA in standby mode [1].

B. Electronic Speed Controller

The ESCs take in an 11.1V power input and up to 3A of current. They receive a PWM signal from anywhere between 50-450Hz between 5 and 3V. The ESC creates a three phase power output that changes depending on the PWM. The PWM inputs function between 45% and 70%, where 45% represents the lowest standby mode, and 70% represents maximum power. This gives a range of 25 different levels that the microcontroller can communicate to the ESCs. The ESCs have a built in voltage regulator that holds the voltage above 10V and can supply a maximum total current of 15A. The ESCs will control the motors at a rate of 920 RPM/V. Lastly, the ESCs create various tones to indicate the status of the ESC including whether it detects an input and whether there is a problem or abnormality [2].

C. AC Motors

There are four brushless AC motors that are connected to the ESCs. They can spin at several hundred rotations per minute, at a top speed of 3500 RPM, and all four can lift a total of 5 to 6 pounds at maximum output. Two of the motors spin clockwise and the others spin counter clockwise in order to cancel out the torque created by spinning blades.

VII. HARDWARE DESIGN DECISIONS

There were a variety of hardware options for this project that could have been used. The real considerations were whether to buy custom components or buy a quadcopter kit; the quadcopter kit was selected. Although the custom components were less expensive than the 3D Printed quadcopter kit, it wasn't guaranteed that all the individual components would work together; there may have been some incompatibility problems [1][2]. Furthermore, the initial plan didn't include the use of ESC modules at all—the original plan only included BJTs

connected to the PWM signal to control DC motors since the transistors are simpler, lighter, and cheaper. However, research revealed that the AC power motors are significantly more efficient, and one transistor would not be able to produce the required AC output from a DC lithium ion battery.

VIII. EVALUATION OF HARDWARE DESIGN CONCEPTS

Although the hardware components selected worked relatively well for a while, the ESCs in particular proved to be somewhat inconsistent and unreliable. The first set of ESCs was working effectively and able to control the four motors equally such that the quad copter could hover and fly as intended. These motors could be calibrated by attaching the drone to a rope—the rope prevented the drone from moving too much so that it would be possible to narrow in on the values needed to create an equal amount of lifting force on beach motor.

However, when doing further development, two of the ESCs stopped working. One ESC had a short and glowed red hot until the power was disconnected, which also destroyed one of the batteries. It appears that one of the capacitors melted, and the proceeding short drained the battery below its minimum operating voltage—going below this voltage breaks the battery and prevents it from charging again. The second ESC simply stopped working and would no longer take an input. The next step was the replacement of these ESCs, but the problem was that the next two ESCs that were installed were very different from the other two from the first set of ESCs.

It was difficult and time consuming to calibrate the motors based on these new ESCs. The drone had to be tied down and the motors had to be adjusted very slowly so that the drone would be stable; this was made especially difficult because the ESCs and motors require different PWM signals to achieve the same rotations per minute and therefore the same lift. For example, an increase in PWM for one ESC didn't increase the rotations per minute of the other motors by the same amount in. So an increase in 1% for one motor was equivalent to an increase in 1.45% for another motor, for instance. This made it almost impossible to calibrate because floating point operations of the PRIMUS are inviable for a real time system. Lastly, without a higher PRIMUS clock speed, the PWM signal could not be increased by smaller increments and 1% was the best possible increment. Despite problems with the second set of ESCs, the drone did work for a while and would work if there was a similar set of ESCs.

IX. SOFTWARE ARCHITECTURE

TCP server will send data to the TCP client over WIFI. The client parses the incoming data and creates a thread handler that transfers the data to the serial UART writing thread, which is sent to the Primus V3R. Once the data is received by the Primus V3R, it changes the PWM values as shown in Figure 5. Then, once the timer triggers, decides whether or not to create a high or low value on a respective pin. This ultimately changes the output to the ESCs [1] [2].

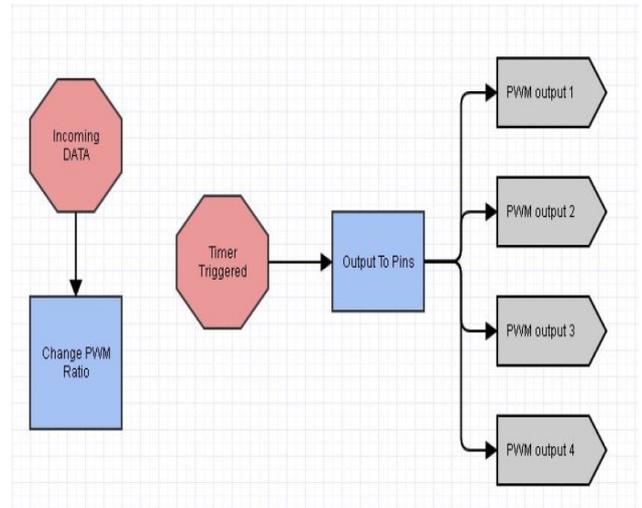


Fig 3. Software Architecture of Primus V3R

X. BEHAVIOR MODEL

To get started, the user must enter the IP address and port number, and then press connect, Once the server handshake is complete, client waits for server data or an error. If the client receives a certain command, it might write to USB output so that the output increases or decreases the PWM. Lastly, if the client disconnects, then the client tries to reconnect to the server until it is able to reconnect.

XI. ANDROID APPLICATION

The application [3] [4] is made in html/css/JavaScript with Cordova. Cordova is a technology who permits to create an application with web technology compatible with android, iOS and more. The users think it is a standard android app but in fact this is a web view. It's easy to develop applications like this for web developers. The android smartphone used for control smone should have magnetometer, accelerometer and gyroscope in order to read the sensors and determine the movement of the smartphone for sends the controls to the quadcopter.

Using the screen as virtual joysticks like the rest of apps, smone app also uses the inclination of the smartphone for pitch, roll and yaw. Throttle is controlled with a finger on the screen, at any moment you can lift the finger from the screen for altitude hold. Arm and disarm are volume buttons.

The App works on old smartphones from android +v2.3, but works better with new smartphones.

MECHANICAL DESIGN

The mechanical design [1] [2] [5] for the senior project was based on the 3D Printed quadcopter kit, which came with the following: a quadcopter chassis, the necessary screws and bolts, the motor blades, the “blade to motor” attachments, solder-able connectors to the ESCs, and the motors themselves. The solder-able connectors and “blade to motor” attachments made it easier to replace damaged components. Furthermore, the exposed connectors were

heat-shrunk and firmly held down to the chassis with zip ties to prevent a short—the short could have caused a toxic and dangerous battery explosion or fire. Furthermore, the battery was not just soldered onto the voltage terminals of the drone. Wires were soldered to the positive and negative terminals on the drone chassis on one end. The other ends of the wires were soldered to a plug, which could connect to the battery. The plug could then be disconnected and reconnected from the battery, which made it possible to manually disable the power source in case of an emergency.

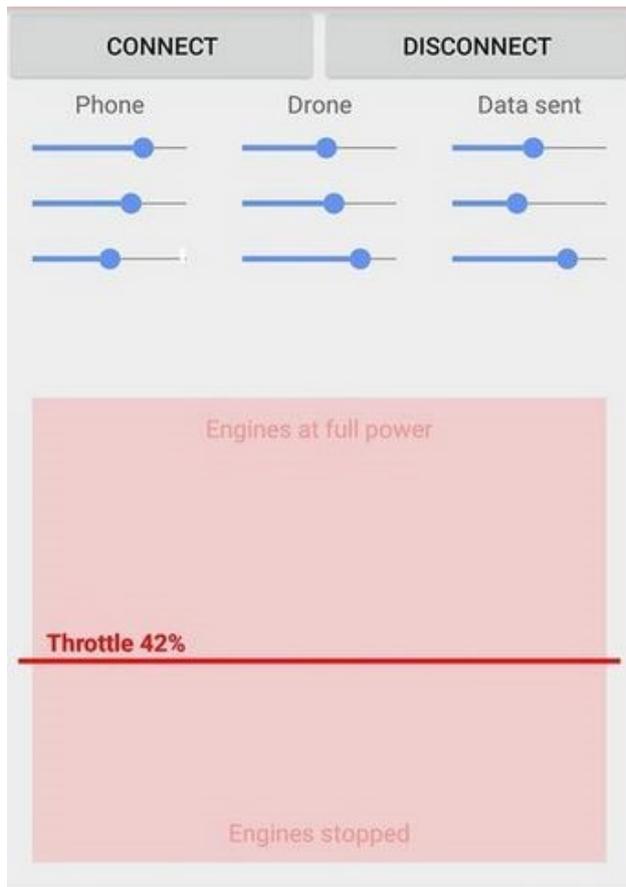


Fig 4. Android Apk picture

Next, the battery was attached to the drone with several sticky pads and Velcro straps. The final piece of the mechanical design was the attachment of the Android phone. The Android phone stayed on the bottom-middle of the drone and was surrounded with shock absorbing Styrofoam to protect against crashes. This meant that the center of gravity was as low and central as possible. The Velcro straps held the Styrofoam and phone to the bottom of the drone and a small hole was cut in the bottom of the Styrofoam to enable the camera to function without being blocked. Lastly, the Primus V3R microcontroller was placed on top of the drone and held down with zip ties to give easy access to the pins. Figure 9 is a picture of the final mechanical design of the drone.



Fig 5. Final Picture of Drone

ACKNOWLEDGMENTS

The authors would like to thank Dr. Loganathan R, HOD, Dept. Of Computer Science and Engineering HKBKCE, and the project coordinator for their inspirational guidance and support. And cannot forget the constant encouragement and help provided by internal guide Khallikkunaisa Associate professor, Dept. Of Computer Science and Engineering, HKBKCE.

REFERENCES

- [1] Yang, W., Chun, M., Jang, G., Baek, J. and Kim, S. (2017). A study on smart drone using quadcopter and object tracking techniques. 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT).
- [2] RUSSELL, A. (2018). Android Drone: REMOTE QUADCOPTER CONTROL WITH A PHONE. COMPUTER ENGINEERING SENIOR PROJECT. [online] Available at: <http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1225&context=cpep> [Accessed 29 Apr. 2018].
- [3] Instructables.com. (2018). The Drone Pi. [online] Available at: <http://www.instructables.com/id/The-Drone-Pi/> [Accessed 29 Apr. 2018].
- [4] Instructables.com. (2018). Fione 3.0 Drone Quadcopter. [online] Available at: <http://www.instructables.com/id/Fione-30/> [Accessed 29 Apr. 2018].
- [5] Kilby, T. and Belinda Kilby (2015). Getting Started with Drones. Maker Media, Inc.