NRF24_MultiWii_Drone

Release 9e41ef1

John Santos

Aug 31, 2025

CONTENTS:

Add your content using `reStructuredText` syntax. See the reStructuredText documentation for details.

CHAPTER

ONE

NRF24-MULTIWII-DRONE

1.1 Introduction

This documentation provides an overview of the NRF24 MultiWii Drone project, including its design, components, and step-by-step tutorials for building both the drone and its controller. It is structured to help users understand the project from its initial concept to the final assembly, including the challenges faced and solutions implemented in various prototypes.

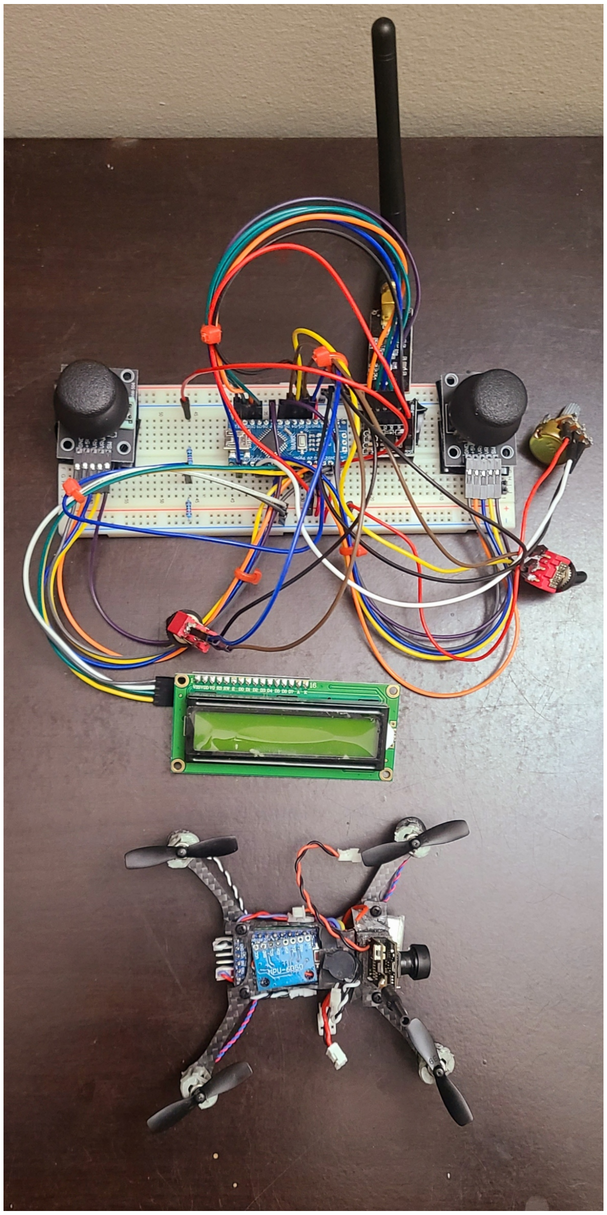
To learn more about the different parts of the project, explore the sections below:

1. **Prototypes** Learn more about the various prototypes and design changes of the drone. See the full details in the *Prototypes* section.
2. **Drone Tutorial** Learn more about the step-by-step tutorial for building the NRF24 MultiWii Drone. See the full guide in the *Drone Tutorial* section.
3. **Controller Tutorial** Learn more about the step-by-step tutorial for building the NRF24 MultiWii Drone Controller. Check out the complete guide in the *Controller Tutorial* section.
4. **Reflection** See the personal reflection on the project, including the motivation, challenges, and lessons learned. Read the full reflection in the *Reflection* section.

1.2 Prototypes

This page will provide an overview of the design of the prototypes that will start with the initial prototype and the reasons behind the design changes that lead to the next prototype.

1.2.1 Protoype 1



The first protoype is the initial attempt at building the NRF24-MultiWii-Drone. However, there were weaknesses in the design that led to failure.

The main issues are listed below:

1. **Too heavy**
    - Motor fasteners that were improvised using drywall anchors were too heavy (Motors will be glued directly in the next prototype).
    - The wires were too thick using 24AWG.

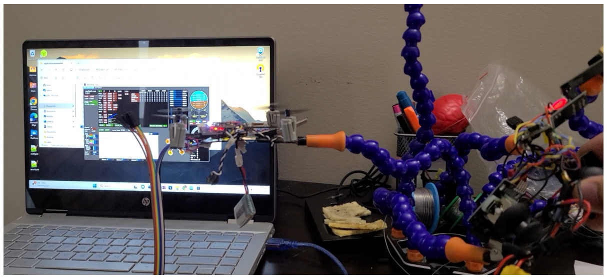
- The perforated board was too large with lots of unused space.
- The soldered pin connectors (NRF24L01, perforated board, buzzer) are too heavy (These will be removed and wires will be soldered directly in the next prototype).
- 1N5819 diodes are too heavy (Using 1N4148 surface mount diodes in the next prototype).
- Using velcro is too heavy (Use super glue instead).

2. **Small Propellers/Less Thrust**
    - Using 2 blade (faster) 37mm propellers (Use 4 blade propellers for more thrust).
3. **Conductive Carbon Fiber Frame**
    - Possible short circuits when mounted on the carbon fiber frame (Use kapton tape for better insulation with the electrical components).
4. **Poor Solder Connections**
    - Damaged solder tips (oxidized) resulted in poor solder connections with possible decrease in conductivity and connectivity between components and possible short circuits.
    - Replace solder tips and properlly resolder the connections in the next prototype.
5. **Need Soldered Controller with Enclosure**
    - The controller uses a breadboard with weak connections and unmanaged wiring. Requires a proper enclosure with soldered connections for better reliability.

Additional materials and replacement needed for the next prototype:

1. Arduino Pro Mini Atmega 328P 5V/16MHz (un-soldered)
2. MPU6050 from DFRobot (un-soldered)
3. **Motor Encoder Circuit**
    - Surface mount 1N4148 diodes
    - Solder tips + high quality thin solder
4. Larger 4 blade propellers
5. Kapton tape for insulation
6. Copper Sheet for proper grounding

1.2.2 Protoype 1.1



The protoype 1.1 is the second attempt at building the NRF24-MultiWii-Drone. However, there were still weaknesses in the design that led to failure.

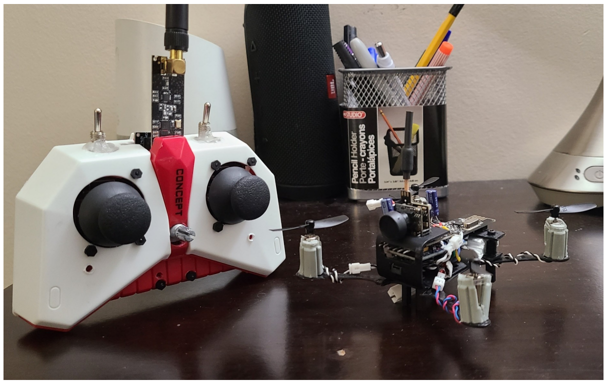
The main issues are listed below:

1. **Kapton Tape Does not Provide Proper Insulation**
    - After adding one layer of kapton tape, it seemed that the electrical components, mainly the IMU shorted and was damaged. This resulted in the gyro unable to properly register the movements in MultiSim. The next prototype will include a proper enclosure for the electrical components of the drone. Although the main frame will still be carbon fiber, a separate box enclsoure will be used to cover the electrical components properly using light materials.
    - Furthermore, investigating the user of hot glue for proper insulation. However, initial research suggests this may lead to further design failures.
2. **Controller is Too Complex**
    - "Simplicity is the ultimate sophistication." - Leonardo da Vinci
    - Unnessary components in the controller atleast for the MVP such as the 16x2 LCD to track voltage and the potentiometer should be removed. The controller should be simplified to only include the basic components for communicating with the drone and this includes the Arduino Nano controller, the radio module + PA + LNA components, the two joysticks, the two SPDT switches, and the SPST switch with the 3.7V batteries.
3. **Controller Battery Has Too Much Current**
    - Prolonged use of the controller lead to overheating of the components and controller failure. Theory is that the battery packs too much current which the components could not handle resulting in breakdown. The two batteries are connected in parallel which are 3.7V 1000mAH. Looking into the use of 3.7V and 600mAH batteries instead.
4. **Remove the Grounded Copper Sheet**
    - This may not be needed as I have not encountered any issues with the drone resetting. This solution was suggested online, but I should not implement solutions to problems that does not exist in my design.

Additional materials and replacement needed for the next prototype:

1. Arduino Pro Mini Atmega 328P 5V/16MHz (un-soldered)
2. Arduino Nano
3. Radio Modules NRF24L01 + PA + LNA
4. 3.7V 600mAH batteries (2x)

1.2.3 Prototype 1.2



The prototype 1.2 is the third attempt at building the NRF24-MultiWii-Drone. This prototype addresses the issues found in the previous prototypes and introduces new design elements. The primary issue in this prototype is that the motors do not respond despire the joystick movements being correctly translated in MulitSim. The drone keeps resetting and attempts to calibrate.

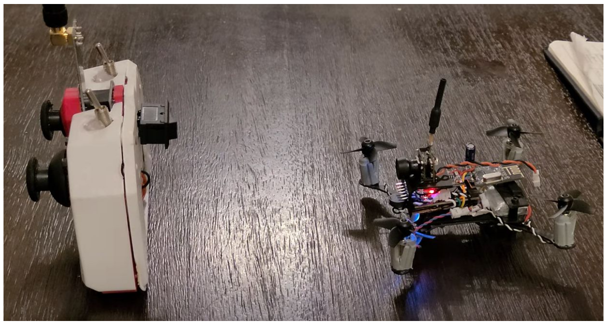
The possible issues are listed below:

1. **The battery discharge rate is too low (25C) and that a proper drone battery with a higher discharge rate (30C or higher) is needed.**
    - Recommended to use "Turnigy Nano-Tech" batteries or similar for their high performance.
2. The power for the radio is not consistent and requires a 10uF filtering capacitor at the NRF24 power inputs.
3. **The power for the Arduino Pro Mini is not consistent and requires a 100uF filtering capacitor.**
    - Confirm Arduino Pro Mini 3.3V 8MHz, or 5V 16MHz is required.
    - Research shows that Arduino Pro Mini 5V 16MHz is recommended to be compatible with MultiSim.
4. **The power lines has a large AWG (small thickness) where the current cannot be supplied properly.**
    - Recommended to use solder with lead and keep solder enclosed after use to avoid contamination/oxidation.
    - For the motor driver, ensure the proper components are rated for this circuit. These components are being used but requires confirmation.
    - 0603 10K SMD resistor 103, SI2300DS-T1-GE3CT-ND N-Channel Mosfet 30V 3.6A, IN4148 diode surface mount.
5. **Motor PWM signals could be too weak to drive the motors.**
    - Requires oscilloscope to confirm suspicion.
    - This factor can be set in the MultiWii software, *float adjustmentFactor* on line 1069 of output.cpp.
6. Potential EMF noise or leaks is affecting the IMU readings?
7. The Arduino Pro Mini is faulty which was purchased from "Hutomwua". The previous prototypes was working which was purchased from "Robojax".

Additional materials and replacement needed for the next prototype:

1. Arduino Pro Mini Atmega 328P 5V/16MHz (un-soldered) from Robojax specifically.

1.2.4 Prototype 1.3



The prototype 1.3 is the fourth attempt at building the NRF24-MultiWii-Drone. This prototype was successful in motor response from the controller movements. However, there were still issues where the drone movements were erratic and unstable. There is enough motor throttle to lift the drone, but the drone was unable to properly lift due to instability and lack of motor synchronization.

The possible issues are listed below

1. **The drone is not calibrated properly.**
    - The drone needs to sit on a flat surface for a proper calibration.
    - The accelerometer and the gyroscope needs proper calibration.
    - Adjust settings in MultiWii configuration with max smoothness.
2. **Motor direction is wrong.**
    - This can be verified by feeling if the air is being pushed upwards.
    - Record slow motion video to see the direction of the motors.
3. The motor RPMs are not the same and unsynchronized.
4. **The forward direction of the MPU6050 is in the opposite direction.**
    - Rewire orientation of the motors to have face the MPU6050 in its forward direction.
5. **The drone is still too heavy and certain weights of the components are not balanced causing the center of gravity to be offset.**
    - Remove heavy motor mounts and just rely on superglue to attach the motors.

1.3 Drone Tutorial

This page will show the step-by-step tutorial for building the NRF24 MultiWii Drone.

1.3.1 Materials

1. Arduino Pro Mini Atmega328P 5V/16MHz (x1)
2. MPU6050 Gyroscope/Accelerometer (x1)
3. NRF24L01+ Transceiver Module (x1)
4. Lilypad 5V Buzzer Speaker (x1)
5. Micro 600TVL FPV Camera with 5.8GHz 25mW Transmitter (x1)
6. 6x15mm 0.8mm Shaft Coreless Motors 19000KV (x4)
7. 1N4148 Surface Mount Diodes (x4)
8. S12300DS Surface Mount N-Channel MOSFET (x4)
9. 10KOhm Surface Mount Resistor (x4)
10. 4 Blade 31mm Propellers (x4)
11. LiPO Battery 3.7 220mAH + charger + JST Battery Connector (x1)

Common

1. 30AWG Wires
2. Perforated Board (8x2cm)
3. Copper Sheet
4. Kapton Tape
5. Super Glue
6. Carbon Fiber Sheet
7. Mini USB to TTL Serial Converter Adaptor
8. M2 Nylon Hex Spacer Standoff Kit with Male and Female Screw Nut etc.
9. FPV Goggles

1.3.2 Frame

TBA

1.3.3 Electrical

1.3.4 Software

Libraries needed * RF24 * LIQUID_CRYSTAL_I2C * TimerFreeTone_v1.5

1.4 Controller Tutorial

This page will show the step-by-step tutorial for building the NRF24 MultiWii Drone Controller.

1.4.1 Materials

1. Arduino Nano Atmega328P 3V (x1)
2. NRF24L01+PA+LNA Transceiver Module with 8-pin Breakout Adapters (x1)
3. 16x2 I2C LCD (x1)
4. Joystick Module for Arduino Dual Axis Sensor (x2)
5. 3-pin Switch (x2)
6. 100KOhm Potentiometer (x1)
7. Power Switch (x1)
8. Power Connector (x1)
9. Charging Connector (x1)
10. 3.7V LiPO Battery 1000mAH (x2)

1.4.2 Frame

TBA

1.4.3 Electrical

1.4.4 Software

Libraries needed * RF24 * LIQUID_CRYSTAL_I2C * TimerFreeTone_v1.5

1.5 Reflection

1.5.1 Log Entry: June 8, 2025

The inspiration behind this project stems from *Max Imaginations' Mini FPV Drone Tutorial*. Watching that video sparked the idea to create a similar design with my own custom modifications to better understand the process of engineering design. Having recently graduated with a bachelor's degree in electrical engineering, I wanted to build more experiences in the field of electrical engineering. I found that by embarking on this project, I could apply my knowledge I gained in my studies to a practical application and also explore new concepts and ideas along the way.

It was not easy to commit to the project in the beginning because deep down I knew that I would be spending a lot of time and resources on this project even more so that this project requires a lot of electrical and mechanical components that could be quite expensive. Furthermore, I am also working full-time as a software engineer and finding free time to work on this project was a challenge. Alot of the time spent on this project was during the weekends and late nights after work which required a lot of sacrifice.

However, I was determined to see this project through to better myself as an electrical engineer and to create something new and challenging despite my lack of personal experiences in building drones or electrical engineering projects in general.

The start of the project was slow as I had to first list all the electrical and mechanical components I needed to build the drone and its controller. I wanted to gather as much detail as possible such as the cost, size, and weight of each component to assess its viability for the project. Most of the components I had used remains the same as Max Imaginations' Drone, but I wanted my own custom design of the frame.

After listing the components for the drone and the controller, I had started to replicate the wooden frame design by Max Imagination, but after recreating the frame, I found that it was too small and simple looking for my liking. I wanted a more streamlined and modern design so I explored other designs online that uses carbon fiber. I eventually skecthed out a design that I liked and draw a blueprint of the components for my frame as a cutout template. I then moved on to purchasing a 0.5mm thick carbon fiber sheet. I printed the cutout template and made the cutouts on the carbon fiber sheet for each component of the frame using a dremel tool. The frame was eventually built using carbon fiber and M2 screws and nuts to hold the components in place.

The next step was building the electrical components of the drone. I had to first study the circuit provided by Max Imaginations' Drone and I also redraw the circuit in EasyEDA for my own references. After drawing the circuit schematics for the drone and the controller, I then purchased the components from Amazon, Digikey, and Mouser. This stage was also a bit slow as I had to wait for the components to arrive before I could continue with the project. There were a lot of delays and some parcels lots and I had to reorder some of the components from different sources.

Once all the electrical components arrived, I started building the circuit for the drone by soldering the components together. I had little experiences with soldering electrical components so I had some challenges in this area. Primarily, the soldering iron oxidized quickly which is probably due to poor techniques I am employing or poor solder material that I am using. I had to switch solder tips quite frequently, but this wasn't very efficient as I ran out of solder tips and I was forced to use an ineffective soldering iron which lead to poor soldering quality. This stage started to take a bad turn for me as the poor soldering quality affected the overall quality of the design as the circuit was prone to short-circuiting and the components not working as expected.

I reached a stage where I had the circuit for the drone built and assembled and programmed the Arduino Nano, but there were lots of failures with MultiSim. The drone was able to be programmed, but it did not interface well with MultiSim as the MPU6050 Gyro movements was not translated well by the software which looked like a lot of noise happening. Furthermore, the MPU only seemed to move digitally when grounding pin D12 on the Arduino Nano which was not expected. At this stage I was not satisfied with the current work on my drone as I am starting to see failures and weaknesses in my design. At this point I had to step back and reflect on my design and make revisions to resolve the issues I was facing. I've documented the issues with this design which I have indicated as "prototype one" under the *prototypes section*.

In this stage, I have planned to recreate the design with the revisions and name it as "prototype 1.1" for a better outcome as the previous design.

1.5.2 Log Entry: July 22, 2025

This log will provide details to the outcome of prototype 1.1 and the revisions made to the design. This prototype had a much more compact design with shorter connections and lighter wires. Furthermore, the controller had a proper enclosure recycled from a toy controller enclosure with modifications to fit the new components. However, this prototype also failed the final testing. The initial work for this prototype started by assembling the electrical components by resoldering the drone circuit using 28 AWG silicon copper wires. After the drone circuit is complete, the next task was to modify an old toy controller enclosure to fit the new controller components. This task involved drilling and cutting certain parts of the enclosure using a dremel tool. Once the enclosure was fitted, the new components were installed and soldered.

In this stage, the drone circuit and the controller circuit are both assembled and programmed for testing. The first test on the drone had some success in which the IMU was able to ready the gyro movements. However, the flight controls were unresponsive to the joystick movements of the controller. I found later on that the radio modules from the controller were faulty. This test involved setting up a simple communication between two of the spare radio modules on a breadboard. After reordering a new set of radio modules, I was able to get clear communication between these new modules. Moving forward, the drone was also able to capture the signals sent from the controller as shown in MultiSim. Lastly, I was not fully confident that the drone enclosure was properly insulated, so I tested the drone responding without it's enclosure. This testing was successful as the motors properly relayed the controller's signals with the expected motor speed based on the throttle joystick movements. However, the drone failed once I tested with the drone enclosure attached. I suspect, this had shorted the drone because of its carbon fiber frame. I had regretted not testing the insulation of the frame even with Kapton tape. Again at this point, I needed to step back again and assess the failures of this prototype and make more revisions to resolve these issues. More details of this assessment under the *prototypes section*.

In this stage I have ordered replacements for the damaged components such as the Arduino Nano, Arduino Pro Mini, NRF24L01 radio modules, and the controller battery. I have planned to recreate the design with the revisions and name it as "prototype 1.2" for a better outcome as the previous design.

1.5.3 Log Entry: August 18, 2025

This log will provide details to the outcome of prototype 1.2 and the revisions needed to be made for the design. After implementing this prototype, the behaviour of the drone was quite erratic. The drone kept resetting when throttle is decreased near zero. It seems to keep resetting and attempts to recalibrate the IMU. The motor responses were rare and seldom spins. When the motor does respond, it responds in bursts and then the drone resets. Looking at the MultiSim outputs, the AUX1 and AUX2 were correctly translated including the joystick movements for roll, pitch, yaw, and throttle levels. So I don't think the issue has to do with the controller, rather with the drone itself.

I suspect possible issues with the drone circuit listed below.

1. The battery discharge rate is too low (25C) and that a proper drone battery with a higher discharge rate (30C or higher) is needed.
2. The power for the radio is not consistent and requires a 10uF filtering capacitor at the NRF24 power inputs.
3. The power for the Arduino Pro Mini is not consistent and requires a 100uF filtering capacitor.
4. The power lines has a large AWG (small thickness) where the current cannot be supplied properly.
5. Motor PWM signals could be too weak to drive the motors.
6. Potential EMF noise or leaks is affecting the IMU readings?

Looking for a battery replacement greater than 25C was quite a challenge as most of the batteries I found were 25C or lower. It was recommended to use “Turnigy Nano-Tech” batteries, but these were always not available and kept being out of stock due to their high demand for the high performance. Although I did manage to find a 30C LiPO battery, not from Turnigy, but from AMZZN, it still didn’t resolve the main issue. The drone behaved a bit better, but most of the resetting issue persisted.

I moved forward by adding filtering capacitors at the radio module and Arduino Pro Mini power inputs, but this did not resolve the issue. I also ensured the solder for the power connections and the radio connections were solid and well connected, but this also did not resolve the issue. I even created a new motor driver with new MOSFETs and diodes, but this still did not resolve the issue.

I then suspected, it has to do with the software and that the PWM signals were too weak to drive the motors. Unfortunately, I do not have an oscilloscope to measure the PWM signals properly, but I found the factor in the MultiWii software that tunes the PWM outputs. Following the tutorial by Max Imaginations, the variable should be *float adjustmentFactor = 0.255* in line 1069 of output.cpp. I set this value to 0.255 from the default of 1, but this did not resolve the issue. As a future work, it seems that it will be best to purchase an oscilloscope to confirm this suspicion.

Moving forward, I retraced my steps and assessed the instructions for properly arming the drone using MultiSim. First the AUX1 and AUX2 should be set high for arming the drone and the beeper respectively. Once set, the drone accelerometer should be calibrated and the read and write to save the changes. To arm and calibrate the drone, the AUX1 should be set low and the throttle and pitch controls are set to zero until the calibration noises are in effect. Once in effect, wait atleast 3 seconds to apply the calibrations. With these steps, the drone still didn’t manage to respond.

My next suspicion is the hardware itself. I purchased the Arduino Pro Mini from “Hutomwua” which did not work, I intend to purchase the Arduino Pro Mini from “Robojax” which worked in the previous prototype (v1.1) where the motors actually responded. I have purchased the same Arduino Pro Mini and will try again in the next version. Furthermore, I will confirm the specifications for the MOSFETs, diodes, and resistors are properly rated for the circuit and the power requirements. More details of this assessment under the *prototypes section*. I have planned to recreate the design with the revisions and name it as “prototype 1.3” for a better outcome as the previous design.

1.5.4 Log Entry: August 31, 2025

This log will describe the details to the outcome of prototype 1.3 and the revisions made to the design. Based on the last suspicion that the hardware purchase for the Arduino Pro Mini was the source of the issue, this suspicion turned out to be true. Looking at the Arduino Pro Mini purchased from “Hutomwua”, the board frequency was only 10MHz. This did not meet specifications for a 5V 16MHz board. I repurchased the Arduino Pro Mini from “Robojax” which was 5V 16MHz and this resolved the issue. The drone was able to respond to the controller’s joystick movements.

However, the design is still not perfect as further issues continued. Although the motors responds to the joystick movements, the drone behaves erratically unable to make a proper liftoff.

The possible issues are listed below.

1. **The drone is not calibrated properly.**
    - The drone needs to sit on a flat surface for a proper calibration.
    - The accelerometer and the gyroscope needs proper calibration.
    - Adjust settings in MultiWii configuration with max smoothness.
2. **Motor direction is wrong.**
    - This can be verified by feeling if the air is being pushed upwards.
    - Record slow motion video to see the direction of the motors.
3. The motor RPMs are not the same and unsynchronized.
4. **The forward direction of the MPU6050 is in the opposite direction.**
    - Rewire orientation of the motors to have face the MPU6050 in its forward direction.

5. **The drone is still too heavy and certain weights of the components are not balanced causing the center of gravity to be offset.**

    - Remove heavy motor mounts and just rely on superglue to attach the motors.