

AUTODESK
Instructables

How to Build a Rc Drone and the Transmitter Using Arduino

By [chamod_perera](#) in [CircuitsArduino](#)



Introduction: How to Build a Rc Drone and the Transmitter Using Arduino



Making a drone these is a simple task these days, but it will cost you much. So I'm gonna tell you how to build a drone using Arduino with low cost. Also I'm gonna tell you how to build the transmitter of the drone too. So this drone is fully homemade. You don't need to buy any flight controller boards or transmitters.

Supplies

We need these items to make the drone,

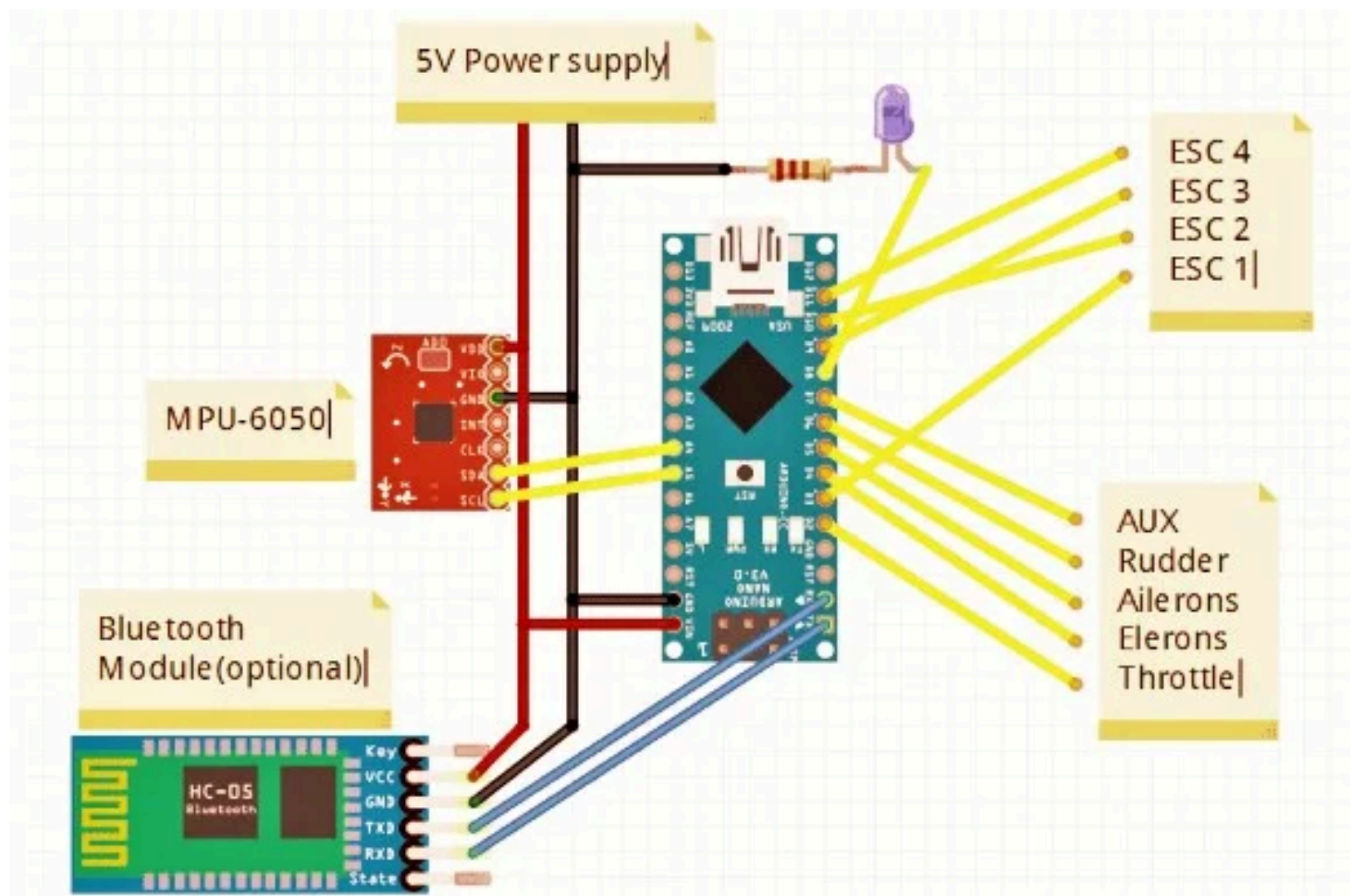
- **For the drone-**

1. **Frame**– The “backbone” of the quadcopter. The frame is what keeps all the parts of the helicopter together. It has to be sturdy, but on the other hand, it also has to be light so that the motors and the batteries don’t struggle to keep it in the air.
2. **Motors**– The thrust that allows the Quadcopter to get airborne is provided by Brushless DC motors and each of them is separately controlled by an electronic speed controller or ESC.
3. **ESCs** – Electronic Speed Controller is like a nerve that delivers the movement information from the brain (flight controller) to the arm or leg muscles (motors). It regulates how much power the motors get, which determines the speed and direction changes of the quad.
4. **Propellers** – Depending on the type of a quad, you can use 9 to 10 or 11-inch props (for stable, aerial photography flights), or 5-inch racing props for less thrust but more speed.
5. **Battery** – Depending on your setup maximum voltage level, you can choose from 2S, 3S, 4S, or even 5S batteries. But, the standard for a quad that is planned to be used for aerial filming (just an example), you will need a 11.4 V 3S battery. You could go with the 22.8 V 4S if you are building a racing quad and you want the motors to spin a lot faster.
6. **Arduino board(Nano)**
7. **IMU(MPU 6050)** – A board that is basically (depending on your choice) a sum of various sensors that help your quad know where it is and how to level itself.

- **For the transmitter-**

1. NRF24L01 Transceiver Module
2. NRF24L01 + PA + LNA
3. Potentiometer
4. Servo Motor
5. Toggle Switch
6. Joystick
7. Arduino Pro Mini

Step 1: SCHEMATICS



This is the main blueprint of your operation.

How to connect the ESCs:

- Signal Pin ESC 1 – D3
- Signal Pin ESC 3 – D9
- Signal Pin ESC 2 – D10
- Signal Pin ESC 4 – D11

How to connect the Bluetooth module:

- Tx – Rx
- Rx – Tx

How to connect the MPU-6050:

- SDA – A4
- SCL – A5

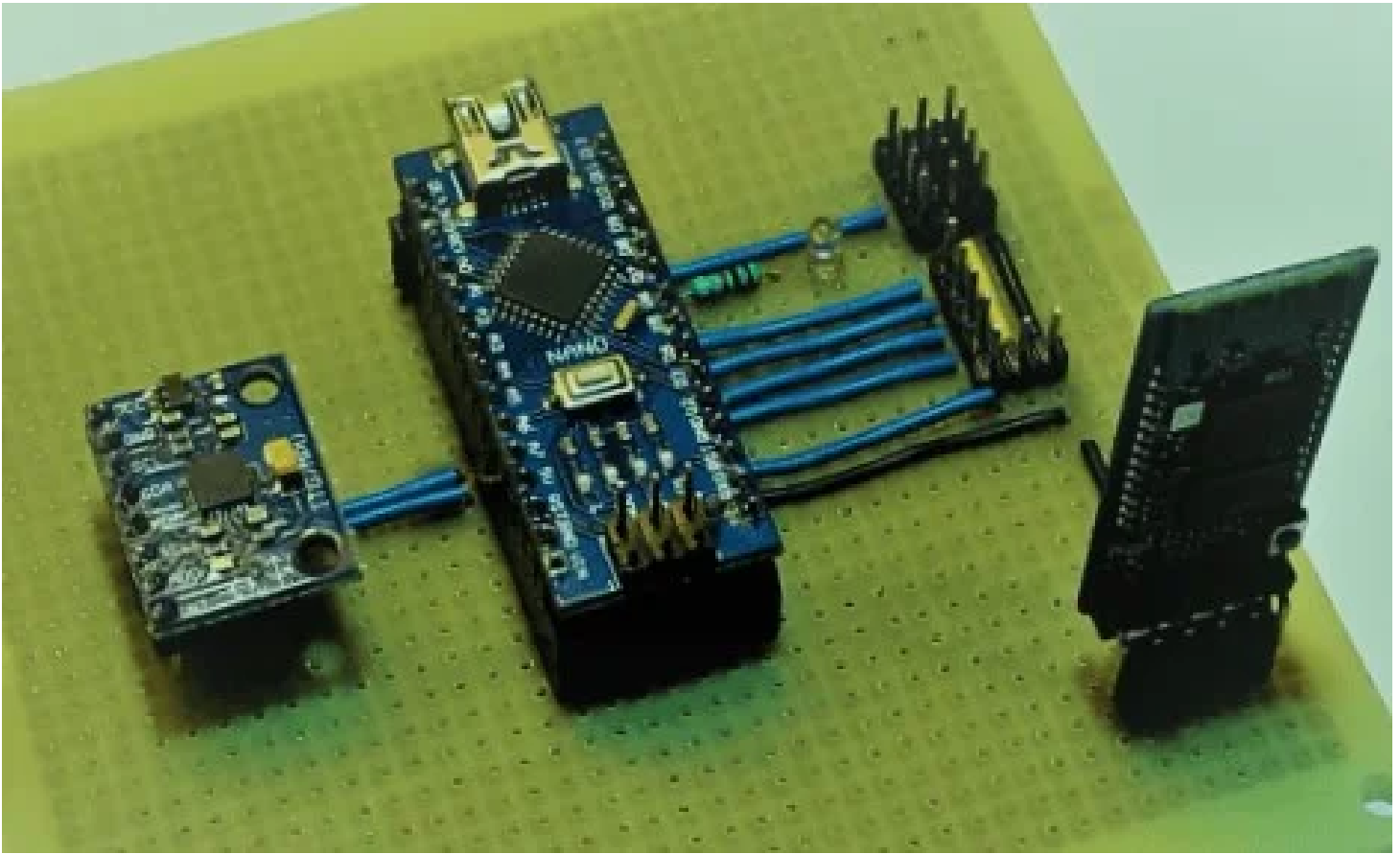
How to connect the LED indicator:

- LED Anode Leg – D8

How to connect the receiver:

- Throttle – 2Elerons – D4
- Ailerons – D5
- Rudder – D6
- AUX 1 – D7 **You need the MPU-6050, the Bluetooth module, the receiver, and the ESCs, to be grounded. And, to do that, you need to connect the all GND pins to the Arduino GND Pin.**

Step 2: SOLDER EVERYTHING TOGETHER



- The first thing you need to do is to take the female headers and solder them to the prototype board. This will house your Arduino board.
- Solder them right in the center so that there's room for the rest of the headers for the MPU, Bluetooth module, Receiver, and the ESCs, and leave some space for some additional sensors you may decide to add in the future.

- The next step is soldering the Receiver and ESCs male headers right from the Arduino female headers. How many male ESC header rows you will have, depends on how many motors your drone will have. In our case, we are building a quadcopter, meaning there will be 4 rotors, and an ESC for each. That further means 4 rows with each having 3 male headers. The first header in the first row, will be used for the Signal PID, the second for the 5V (though, this depends on your ESCs having a 5V pin or not, if not, you will leave these headers empty), and the third header will be for the GND.

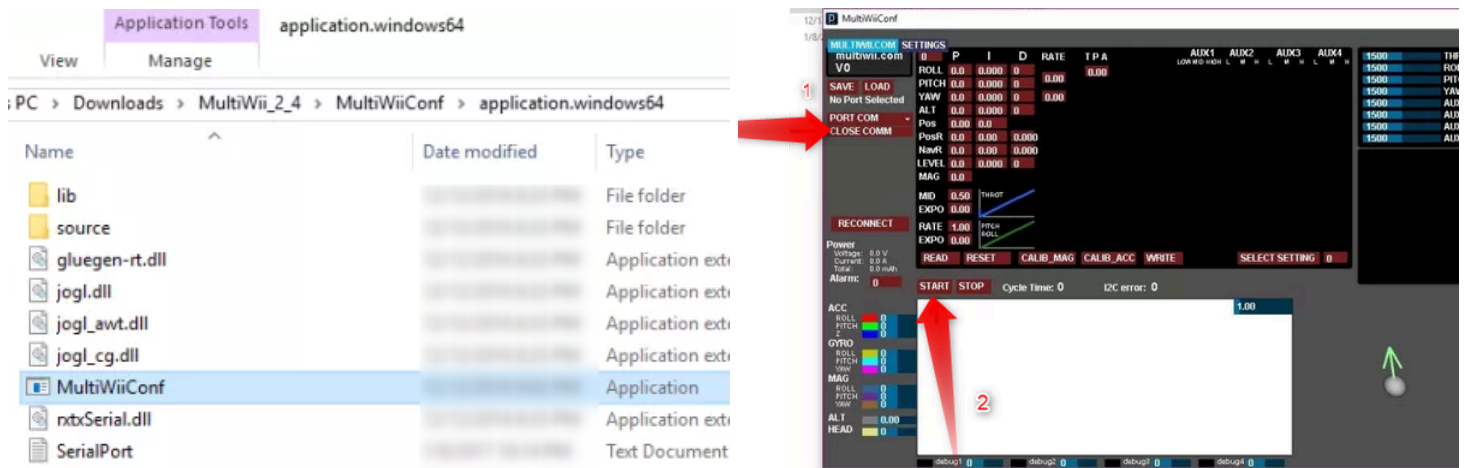
When the ESCs soldering part is over, you can move on to the Receiver headers soldering part. In most cases, a quad has 4 channels. These are Throttle, Pitch, Yaw, and Roll. The remaining free channel (the fifth one), is used for Flight mode changes (the Auxillary channel). This means that you will need to solder male Headers in 5 rows. And, each but one will have one header, while just one of those rows needs 3 headers in a row.

- all the grounds were connected with the Arduino grounds. That includes all ESC grounds, Receiver ground (Throttle signal header completely on the right), and the Bluetooth module and MPU grounds .
- Then, you need to follow the schematics and the connections we explained above. For example, the MPU (SDA – A4, and SCL – A5), and for Bluetooth (TX – TX and RX – RX) of Arduino. After that, just follow the connections as we wrote them: Signal pins of ESC1, ESC2... to D3, D10... of Arduino. Then the Receiver signal pins Pitch – D2, Roll – D4... and so on. Furthermore, you need to connect the Long Lead of the LED (positive Terminal) to the Arduino D8 Pin, as well as add the 330-ohm resistor in between the Ground of Arduino and the LED Short lead (negative terminal). The last thing to do is to provide a 5V power source connection. And, for that, you need to parallel connect the Black wire (ground of the battery) to the ground of all your components, and the Red wire to Arduino, MPU, and Bluetooth Module, 5V pins. Now, the MPU 6050 needs to be soldered to male headers to the ones you plan on using. After that, turn the board 180 degrees and connect all your components to the respective headers on the prototype board.
- Power it up and your Arduino is ready for adding codes through a computer!

The screenshot shows a Windows File Explorer window with the address bar set to 'This PC > Downloads > MultiWii_2_4'. The left sidebar shows the 'MultiWii_2_4' folder expanded, with 'MultiWii' and 'MultiWiiConf' highlighted. The main pane displays a list of files and folders with columns for Name, Date modified, Type, and Size. A red arrow points to the 'MultiWii' file in the list.

Name	Date modified	Type	Size
MultiWii	12/12/2016 8:32 PM	File folder	
MultiWiiConf	12/12/2016 8:32 PM	File folder	70 KB
change	12/12/2016 8:32 PM	Text Document	70 KB

The screenshot shows the Arduino IDE interface. The 'Tools' menu is open, and 'COM3' is selected under the 'Serial Port' section. The 'MultiWii - config.h' file is open in the editor, showing the 'Edit' menu. A red arrow points to the 'COM3' port selection.



1. First, you need to [download the MultiWii 2.4](#). Then extract it.
2. Enter the MultiWii folder, and look for MultiWii icon and run it
3. Use the Arduino IDE to find the "Arduino File" or Multiwii file with ".ino". Any "CPP file" or "H file" are the support files for our Multiwii Code so don't open those. Just use the Multiwii.ino file.
4. When you open the file, you will find many tabs Alarms.cpp, Alarms.h, EEPROM.cpp, EEPROM.h and many more. Find the "config.h"
5. Scroll down till you find 'The type of multi-copter' and then by deleting the "///" you mark is as defined and running. Quad X because we are assuming that you are using the "X" rotor configuration on your quad.
6. Now scroll down and look for "Combined IMU Boards" and activate the type of the Gyro+Acc Board you are using. In our case, we used the GY-521 so we activated that option.
7. If you decide to add other sensors such as a barometer or an Ultrasonic sensor, all you have to do is to "activate" them here and they will be running.
8. Next is the "Buzzer pin", There, you need to activate the Flight indicator options (the first 3 ones)
9. Unplug the Arduino board from the Flight controller and then connect it to your computer using USB. Once out of the FC and connected to your computer, you will find TOOLS and select the type of your Arduino board (in our case Arduino Nano).
10. Now find "Serial Port" and activate the COM Port the Arduino Nano is connected to (our case, COM3). Finally, click on the arrow and upload the code, and wait for the code to be transferred.
11. When the upload is finished, unhook the Arduino from USB, insert it back to its place in the FC board, and connect a 5V battery so that the entire FC is powered up, and then wait till the LED on the Arduino is red. That means it has finished booting and that you can connect it to your computer again. Now, find the Multiwii 2.4 folder, then the MultiwiiConfig, and locate the folder that is compatible with your OS. In our case, it is the "application.windows64".
12. Now start the MultiwiiConf application And, that's it! You will immediately notice how you move the FC, the values for the Accelerometer and Gyroscope data on the screen. The orientation of your FC is shown at the bottom. In this interface, you can change the PID values and fine-tune your quad to match your personal preferences. And, you can also assign the flight modes to certain Auxillary switch positions in this interface. All you have to do now is find a place for your Arduino FC on the frame and it is ready to hit the skies.

Step 4: Frame



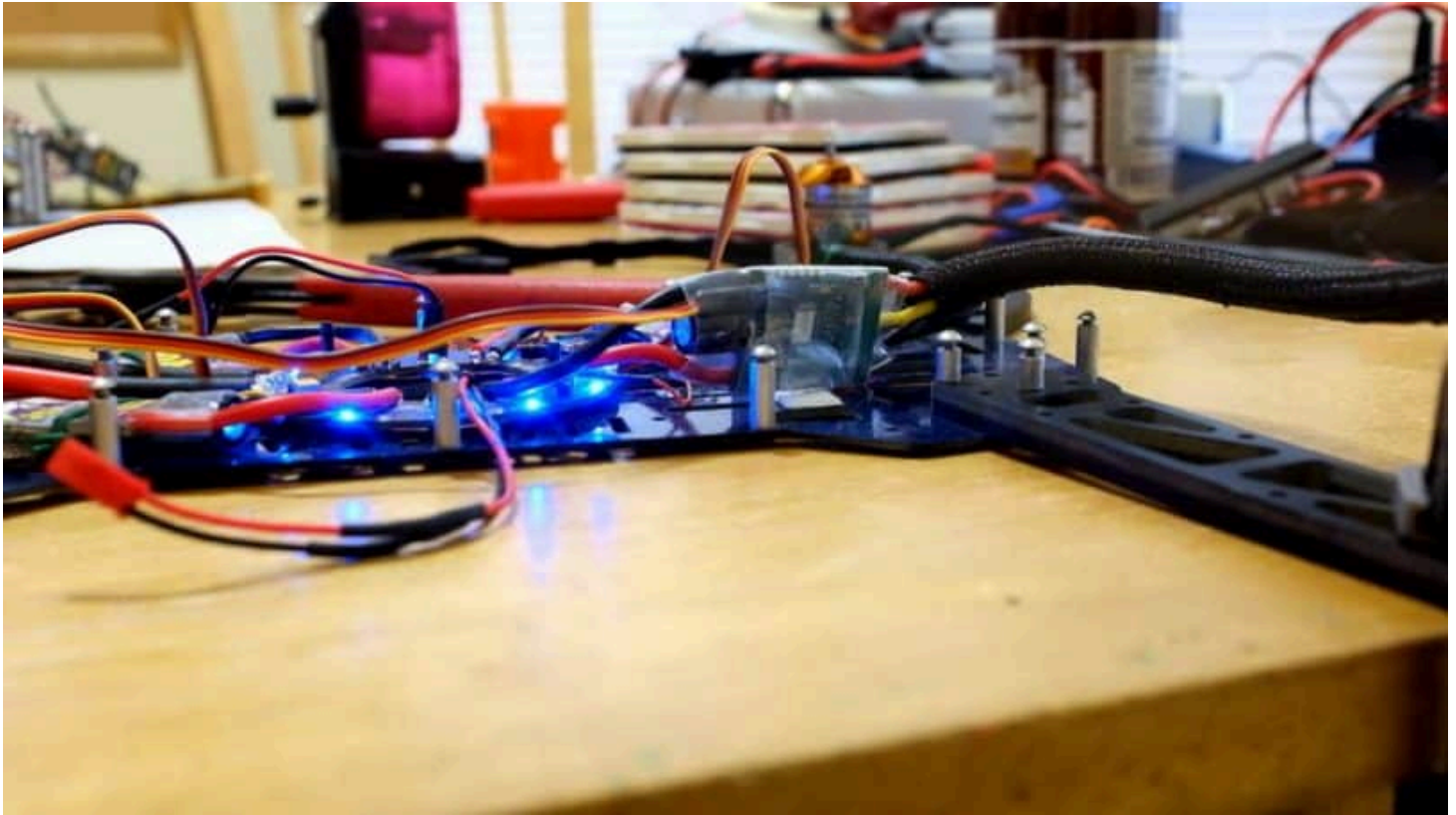
- Now you have to do is to set all parts to the frame. you can buy a frame or [you can make one at home](#).

Step 5: Assembling the Motors and Speed Controllers



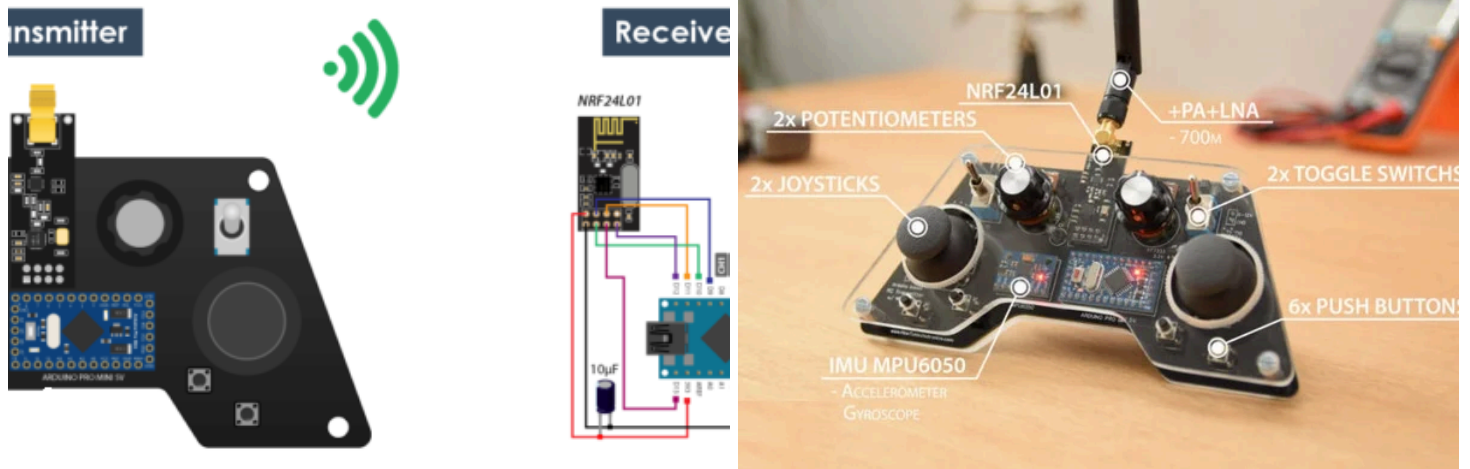
- First you need to do is drill the holes in the frame for the motors, according to the distance between the screws holes on the motors. It would be good to make another hole that will allow the clip and shaft of the motor to move freely.
- It is recommended to connect the speed controllers on the bottom side of the frame due to several reasons which involve the functionality of the drone. These reasons, among others, include that it will “unload” the upper side of the drone where other components should be added.

Step 6: Adding the Flight Controller and the Battery



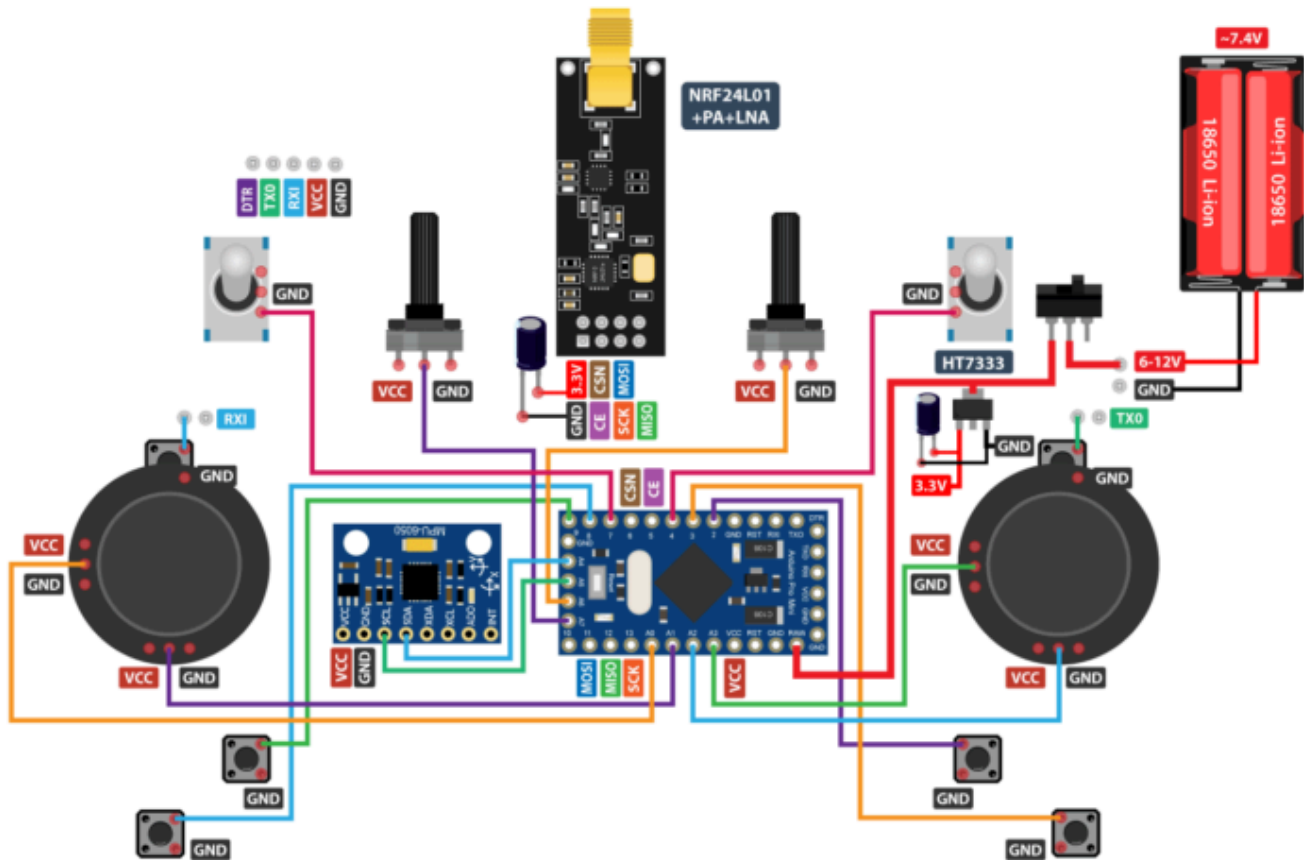
- Now assemble our home made flight controller(arduino receiver) to the center of the drone frame.
- It is recommended to put a small piece of sponge on the underside of the flight controller because it absorbs and reduces the vibrations from the motors. Thus, your drone will be more stable while flying, and stability is key to fly a drone.
- Now add the lipo battery to the bottom of the frame and make sure that the drone is balanced to the center.
- now your drone is ready to take off

Step 7: Making the Transmitter



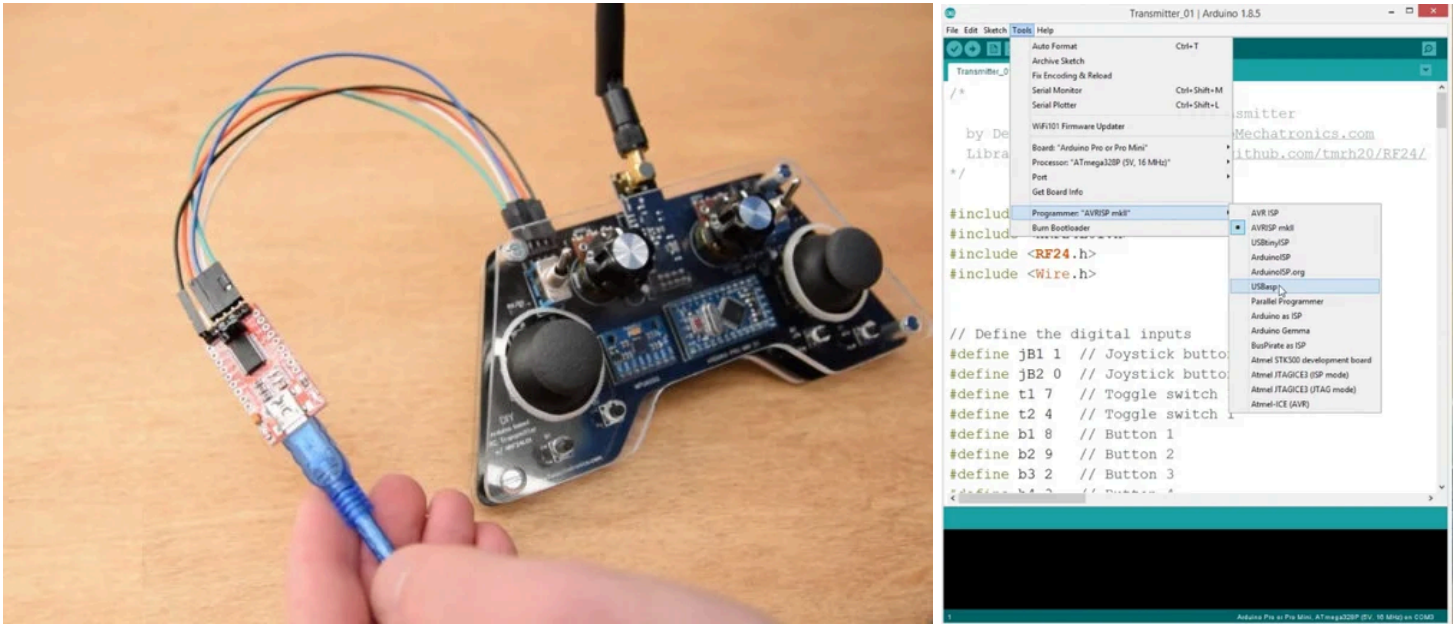
- The radio communication of this controller is based on the NRF24L01 transceiver module which if used with an amplified antenna it can have a stable range of up to 700 meters in open space. It features 14 channels, 6 of which are analog inputs and 8 digital inputs.
- It has two joysticks, two potentiometers, two toggle switches, six buttons and additionally an internal measuring unit consisting of an accelerometer and a gyroscope which can be also used for controlling things with just moving around or tilting the controller.

Step 8: Circuit Diagram



- The brain of this RC controller is an Arduino Pro Mini which is powered using 2 LiPo batteries producing around 7.4 volts. We can connect them directly to the RAW pin of the Pro Mini which has a voltage regulator that reduced the voltage to 5V. Note that there are two versions of the Arduino Pro Mini, like the one I have that operates at 5V and the other operates at 3.3V.
- On the other hand, the NRF24L01 module strictly needs 3.3V and it's recommended to come from a dedicated source. Therefore we need to use a 3.3V voltage regulator which is connected to the batteries and convert the 7.4V to 3.3V. Also we need to use a decoupling capacitor right next to the module in order to keep the voltage more stable, thus the radio communication will be more stable as well. The NRF24L01 module communicates with the Arduino using SPI protocol, while the MPU6050 accelerometer and gyro module uses the I2C protocol.
- You have to solder all parts together according to the diagram. you can design and print a circuit which makes easier.

Step 9: Coding the Transmitter



- For programming a Pro Mini board we need an USB to serial UART interface which can be hooked up to the programming header located on the top side of our controller.
- Then in the Arduino IDE tools menu we need to select the Arduino Pro or Pro Mini board, select the proper version of the processor, select the port and select the programming method to "USBasp".
- Here's the complete Arduino code for this DIY Arduino RC Transmitter
- Upload it to the arduino pro mini.

Step 10: Coding the Receiver

- Here is a simple receiver code where we will receive the data and simply print it on the serial monitor so that we know that the communication works properly. Again we need to include the RF24 library and define the objects and the structure the same way as in the transmitter code. In the setup section when defining the radio communication we need to use the same settings as the transmitter and set the module as receiver using the `radio.startListening()` function.
- Upload it to the receiver

Step 11: Taking Off the Drone



- Firstly, position your drone on the ground and prepare it for operation. Grab your flight controller and then start your first flight carefully and safely.
- However, it is highly recommended to throttle up the drone slowly. Moreover, for the first time, make sure to fly it at a lower altitude.
- I hope that this article will help you to build your homemade drone.
- Don't forget to like this and leave a comment.