IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Disaster ManagementDrone

Dhanashree Nikhare1st author
Electronics Dept

Swati Patil 2nd author
Electronics Dept

Riddhi Rane 3rd author
Electronics Dept

Dr.Shikha Nema 4th author
Guide

***Abstract*— Besides the military and commercial applications of drones, there is no doubt in their efficiency in case of supporting emergency management. SO Drone demand has increased in every sector.Monitoring of natural disaster is necessary during tsunami ,earthquake .so drone plays a very important role in natural calamity .By providing medicine ,food packets to the affected area .This paper evaluates a some initiatives using drones to support disaster management. This is a drone moodle which has an indigenous radio communication. this drone demonstrate the working principles of that radio communication. radio communication using SPI interface. Combining ARDUINO and NORDIC RADIO FREQUENCY 2.4GHZ SEMICONDUCTOR CHIP. Further this drone can be equiped with the accessories which is required for different operations like for investigation of desastered place a camera module, for sending medical supplies a gripper module, for tracking any vehicle or storm a gps module.**

***Index Terms*—Disaster Management,Flight controller,Arduino,RC controller,Radio frequency,NRF24L01 transmitter- receiver,Arduino IDE, Drone, Quadcopter.**

I. INTRODUCTION

Nowadays radio frequency are everywhere. From our cell phones to our TV and military equipments. Radio frequency (RF) is the oscillation rate of an alternating electric current or voltage or of a magnetic, electric or electromagnetic field or mechanical system in the frequency range from around 20 kHz to around 300 GHz. They are used in communication devices such as transmitters, receivers, computers, televisions, and mobile phones. This radio communication can be used in anything from a cartoansubmarinebecauseitisusingaradio frequency of2.4GHZ

2.4GHz radio, the chipset running that radio may support up to 100 simultaneous connections. Compare that to a dual band AP; one radio on 2.4GHz and one on 5GHz. Now we can support up to 200 simultaneous connections, 100 on each radio.

II. PROPOSED METHOD

ThenRF24L01+transceivermoduleismeanttoworkin2.4 GHz worldwide ISM waveband and uses GFSK modulation for data transmission. the info transfer rate are often one of 250kbps, 1Mbps and 2Mbps.

SPI Interface The nRF24L01+ transceiver module communicates over a 4pin Serial Peripheral Interface (SPI) with a maximum rate of 10Mbps. All the parameters like frequency channel (125 selectable channels), output power (0 dBm, 6 dBm, 12 dBm or 18 dBm), and rate (250kbps, 1Mbps, or 2Mbps) are often configured through SPI interface.

The SPI bus uses an idea of a Master and Slave, in commonest applications our Arduino is that the Master and thus the nRF24L01+ transceiver module is that the Slave. Unlike the I2C bus the number of slaves on the SPI bus is restricted , on the Arduino Uno you'll use a maximum of two SPI slaves i.e. two nRF24L01+transceivermodules.

Here are complete specifications:
Frequency Range 2.4GHz ISM Band Air rate 2 Mb/s
Modulation Format GFSK
Max. Output Power 0 dBm
Operating Supply Voltage 1.9 V to 3.6 V
Max. Operating Current 13.5mA
Min. Current(Standby Mode) 26µA
Logic Inputs 5V Tolerant

Communication Range 800+ m (line of sight)

Sr.No	Components	Cost
1	Quadcopter Frame	700
2	Propellers	200
3	Brushless Motor 1000Kv	1600
4	2.4GHz NRF24L01 Transceiver	250
5	2.4GHz NRF24L01 Receiver	250
6	3.3v Adaptor Board	60
7	Arduino Nano v3.0	450
8	11.1V 2200mAh Lithium Polymer Battery	1250
9	Dual Axis XY Joystick Module	160
10	ESC 30A	1520
11	Flight Controller	2590
12	Other Components	5970
	Total	15,000

Fig. 1. Components List

A. nRF24L01

HownRF24L01+transceivermoduleworks?
RF ChannelFrequency

The nRF24L01+ transceiver module transmits and receives data on a specific frequency called Channel. Also so as for 2 or more transceiver modules to talk with each other , they need to urge on the same channel. This channel could be any frequency withinthe2.4GHzISMbandortobemoreprecise, itwouldbe between2.400to2.525GHz(2400to2525MHz). Each channel occupies a bandwidth of but 1MHz. this provides us 125 possible channels with 1MHz spacing. So, the module can use 125 different channels which provides an opportunity to possess a network of 125 independently working modems in one place.

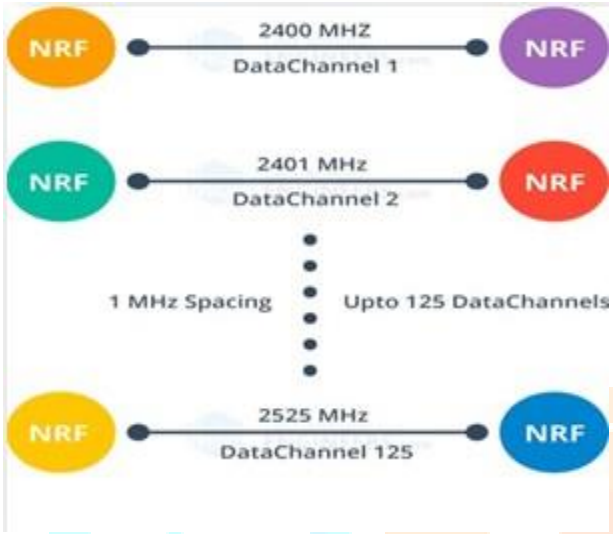


Fig. 2. RF Channel Frequency

nRF24L01+ Wireless Transceiver 2.4GHz 125 RF Channels 1MHz Spacing The channel occupies a bandwidth of but 1MHz at 250kbps and 1Mbps air rate . However at 2Mbps air rate , 2MHz bandwidth is occupied (wider than the res- olution of RF channel frequency setting). So, to make sure non¬overlapping channels and reduce cross¬talk in 2Mbps mode, you would like to stay 2MHz spacing between two channels.

RF channel frequency of your selected channel is about consistent with the subsequent formula:

Freq(Selected) = 2400 + CH(Selected)

For example, if you opt on 108 as your channel for data transmission, the RF channel frequency of your channel would be 2508MHz (2400 + 108)

nRF24L01+ Multiceiver Network The nRF24L01+ provides a feature called Multiceiver. It's an abbreviation for Multiple Transmitters Single Receiver. In which each RF channel is logically divided into 6 parallel data channels called Data Pipes. In other words, a knowledge pipe may be a logical channel within the physical RF Channel. Each data pipe has its own physical address (Data Pipe Address) and may be configured. This can be illustrated as shown below.

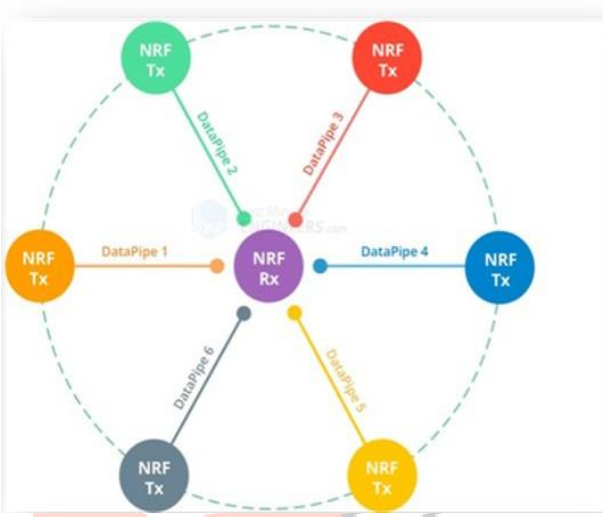


Fig. 3. Multiceiver Network

nRF24L01+ Wireless Multiceiver Network Multiple Transmitters Single Receiver nRF24L01+ Multiceiver Network – Multiple Transmitters Single Receiver To simplify the above diagram, imagine the first receiver acting as a hub receiver collecting information from 6 different transmitter nodes simultaneously. The hub receiver can stop listening anytime and acts as a transmitter. But this will only be done one pipe/node at a time.

Enhanced ShockBurst Protocol The nRF24L01+ transceiver module uses a packet structure referred to as Enhanced ShockBurst. This simple packet structure is weakened into 5 different fields, which is illustrated below.

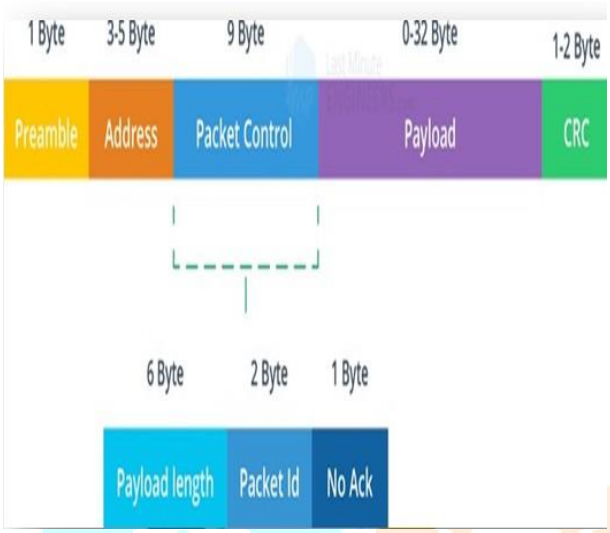


Fig. 4. nRF24L01+ Enhanced Shock Burst Packet Structure

nRF24L01+ Wireless Transceiver Enhanced ShockBurst Packet Structure nRF24L01+ Enhanced ShockBurst Packet Structure The original ShockBurst structure consisted only of Preamble, Address, Payload and thus the Cyclic Redundancy Check (CRC) fields. Enhanced ShockBurst caused greater functionality for more enhanced communications employing a newly introduced Packet Control Field (PCF).

This new structure is great for variety of reasons. Firstly, it allows for variable length payloads with a payload length specifier, meaning payloads can vary from 1 to 32 bytes.

Secondly, it provides each sent packet with a packet ID, which allows the receiving device to work out whether a message is new or whether it's been retransmitted (and thus can beignored).

Finally, and most importantly, each message can request an acknowledgement to be sent when it's received by another device.

So we'd wish to incorporate the essential SPI and thus the newly installed RF24 libraries and make an RF24 object. the 2 arguments here are the CSN and CE pins.

```
RF24 radio(7, 8) // CE, CSN
```

Next we'd wish to make a byte array which may represent the address, or the so called pipe through which the two modules willcommunicate.

```
const byte address[6] = "00001"
```

We can change the price of this address to any 5 letter string and this allows to choose to which receiver we'll talk, so in our case we'll have the same address at both the receiver and thus the transmitter.

In the setup section we'd wish to initialize the radio object and using the radio.openWritingPipe() function we set the address of the receiver to which we'll send data, the 5 letter string we previously set.

```
radio.openWritingPipe(address)
```

On the other side, at the receiver, using the radio.setReadingPipe() function we set an equivalent address and therein way we enable the communication between the 2 modules.

```
radio.openReadingPipe(0, address)
```

Then using the radio.setPALevel() function we set the power Amplifier level, in our case ready to set it to minimum as my modules are very on the brink of 1 another.

radio.setPALevel(RF24*P* *A*_*M* *IN*)

Note that if employing a higher level it's recommended to use a bypass capacitors across GND and three .3V of the modules in order that they need more stable voltage while operating.

Next we've the radio.stopListening() function which sets module as transmitter, and on the other side, we've the radio.startListening()functionwhichsetsthemoduleasreceiver.

```
// at the Transmitter
radio.stopListening()
//at the Receiver
radio.startListening()
```

In the loop section, at the transmitter, we create an array of characters to which we assign the message "Hello World". Using the radio.write() function we'll send that message to the receiver. the primary argument here is that the variable that we might wish to be sent.

```
void loop()
const char text[] = "Hello World"
radio.write(text, sizeof(text))
delay(1000)
```

By using the "" before the variable name we actually set an indicating of the variable that stores the info that we might like to be sent and using the second argument we set the quantity of bytes that we would wish to require from that variable. during this case the sizeof() function gets all bytes of the strings "text". At the highest of the program we'll add 1 second delay. On the opposite side, at the receiver, within the loop section using the radio.available() function we check whether there's data to be received. If that's true, first we create an array of 32 elements, called "text", during which we'll save the incoming data.

```
void loop() if(radio.available())
char text[32] = ""
radio.read(text, sizeof(text))
Serial.println(text)
```

Using the radion.read() function we read and store the info into the "text" variable. At the highest we just print text on the serial monitor. So once we upload both programs, we'll run the serial monitor at the receiver which we'll notice the message "Hello World" gets printed each second.

III. DESIGN

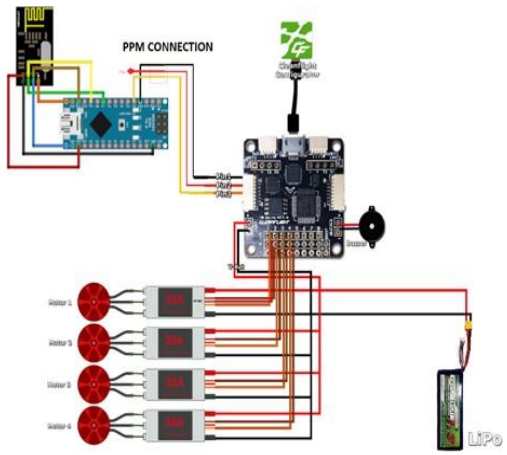


Fig. 5. Design Model of Quadcopter Drone

So design of this model which is shown Fig.4. For implimentation we require Frame, Propellers, Four Motors, Four ESC, Li-Po battery and main component i.e. Filght Controller. This schematic representation show the depth design model. Which show all the connection between electronics compo- nents. The flight controller of the quad copter is the main controlling components for the quadcopter to fly properly. It has 4 ESCs (Electronics Speed Controller) connected to the flight controller. Motors are powered using the li-po battery. The motor has 3 wires which are connected to the ESC three output pins. The connection of motors to the ESC is based on in which direction we need to rotate that motor. The rotation speed of motors controlled using the ESC. In this schematic thin red, orange, black wire signifies +5v, signal and ground respectively. And thick red and black connection represent the +11v and ground line by li-po battery.

In RC controller the receiver i.e. NRF24L01 are attached to the flight controller with Arduino by PPM (Pulse Position Modulation), which have only three line i.e. ground , power and signal. NRF receiver module is also connected to the Arduino board using RX-TX pin.

For controlling the Quadcopter drone, 2.4 GHz radio frequency transmitter and receiver, Arduino, esc, motor are used. Clean-flight configurator and Arduino IDE has been used to developed the control system simulation. The program was the main firmware for the flight control which helps the flight controller to work according the users requirements.

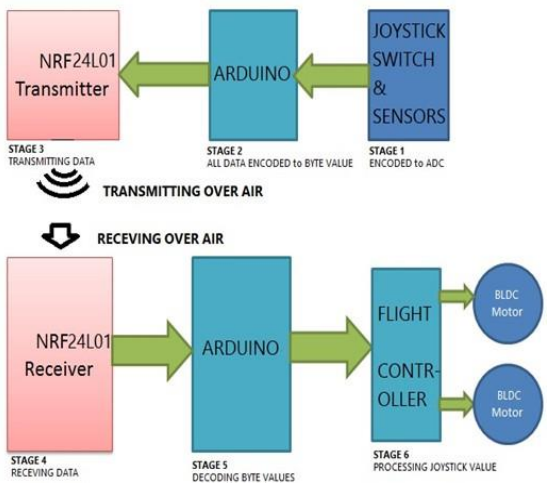


Fig. 6. Block Diagram of Entire Quadcopter Drone System

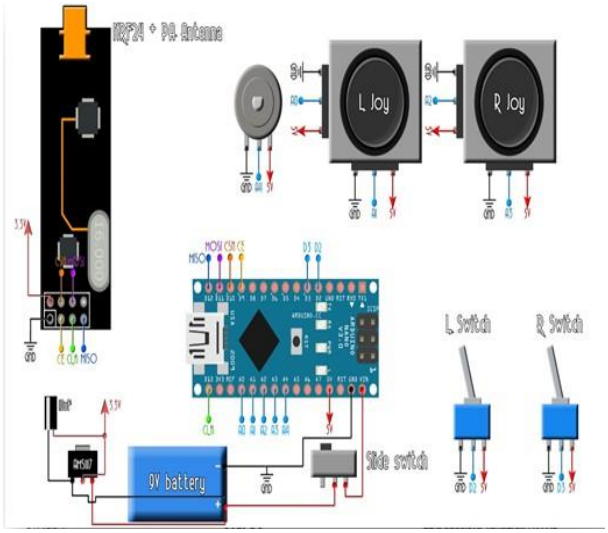


Fig. 7. Circuit Diagram of Transmitter

Fig.5 show the Block diagram of entire system with radio communication device. There are total 6 stages which complete the entire system.

The first three block are totally in transmitter side of radio communication system. Which consist a different type of sensor with joystick switch, Arduino uno and NRF24L01 transmitter. The sensor give analog output to the ADC which convert the analog input to its digital equivalent i.e. byte value and gives it to the microcontroller i.e. Arduino. The Arduino uno is used to take the digital input from the sensor. This input is given to a RF transmitter module that transmit the signal. The signal are transmitting over the air.

The second part of radio communication is receiver side. The NRF24L01 receiver are used with Arduino. The signal are transmitting over the air are receive by receiver. This data are decoded in Arduino and give it to the flight controller to perform the operation.

There are four way to move a drone, this is basic operation Roll - Moves drone left or right in the air, literally "rolling" drone.

Pitch - Tilts drone forward or backward.

Yaw - Rotates drone clockwise or counterclockwise, allowing to make circles in the air.

Throttle - Controls the amount of power sent to drone, which makes the drone go faster or slower.

Let's have a look at the pinout of both the versions of nRF24L01+ transceiver Module.

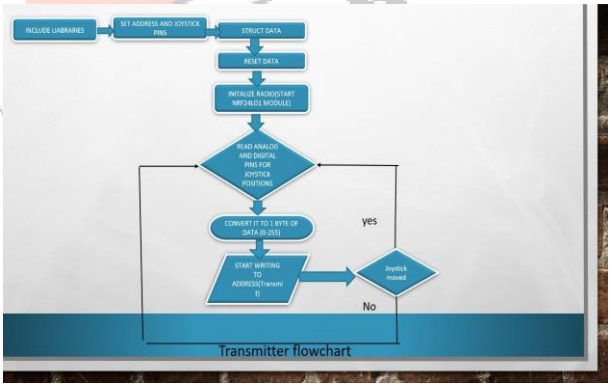
GND is the Ground Pin. It is usually marked by encasing the pin in a square so it can be used as a reference for identifying the other pins.

VCC supplies power for the module. This can be anywhere from 1.9 to 3.9 volts. You can connect it to 3.3V output from your Arduino. Remember connecting it to 5V pin will likely destroy your nRF24L01+ module.

CE (Chip Enable) is an active-HIGH pin. When selected the nRF24L01 will either transmit or receive, depending upon which mode it is currently in.

CSN (Chip Select Not) is an active-LOW pin and is normally kept HIGH. When this pin goes low, the nRF24L01 begins listening on its SPI port for data and processes it accordingly. SCK (Serial Clock) accepts clock pulses provided by the SPI bus Master.

MOSI (Master Out Slave In) is SPI input to the nRF24L01. MISO (Master In Slave Out) is SPI output from then RF24L01. IRQ is an interrupt pin that can alert the master when new data is available to process.



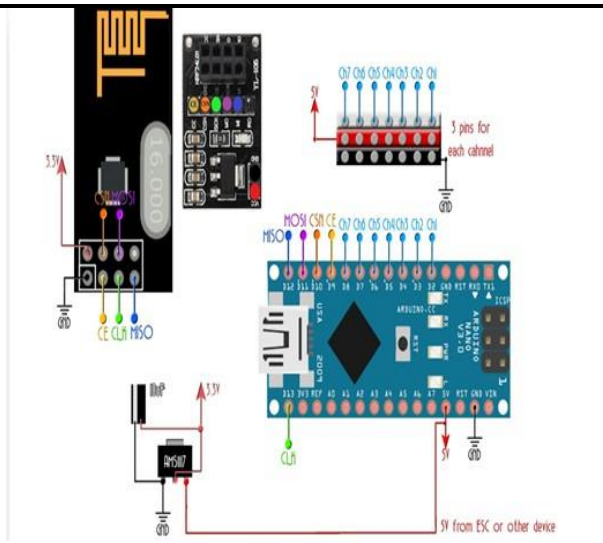


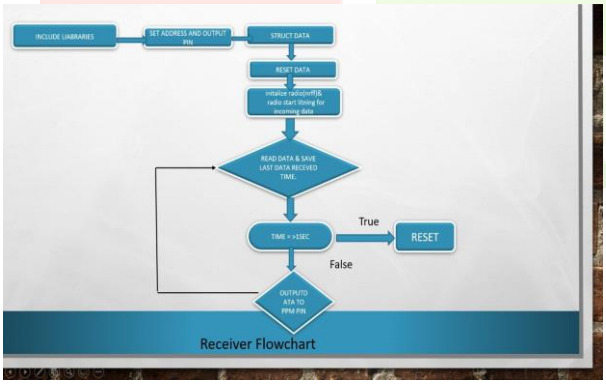
Fig. 8. Circuit Diagram of Receiver

Fig. 9. Custom made Transmitter

Wiring – Connecting nRF24L01+ transceiver module to Arduino NANO

Now that we have a complete understanding of how nRF24L01+ transceiver module works, we can begin hooking it up to our Arduino. To start with, connect VCC pin on the module to 3.3V on the Arduino and GND pin to ground. The pins CSN and CE can be connected to any digital pin on the Arduino. In our case, it's connected to digital pin7 and 9 respectively. Now we are remaining with the pins that are used for SPI communication. As nRF24L01+ transceiver module require a lot of data transfer, they will give the best performance when connected up to the hardware SPI pins on a microcontroller. The hardware SPI pins are much faster than 'bit-banging' the interface code using another set of pins. Note that each Arduino Board has different SPI pins which should be connected accordingly. For Arduino boards such as the UNO/Nano V3.0 those pins are digital 13 (SCK), 12 (MISO) and 11 (MOSI. If you have a Mega, the pins are different. You'll want to use digital 50 (MISO), 51 (MOSI), 52 (SCK), and 53 (SS).

There is joystick added in this circuit.Joystick it is used by microcontroller to determine value of channel, joystick are simple potentiometers which has 3 wires, one positive and one negative and third output. We supply 5V to both positive and negative and we draw our output from output and negative pin, this two (output and negative) pins go to Arduino which reads position of joystick or potentiometer using codes. We uploaded this type of conversion of potentiometer value to digital value which is used by microcontroller is known as ADC analogue to digital conversion and rest work is done by microcontroller by encryption and transmitting this value to another node. Here we have used 2 joysticks one is modified and other is non modified. Modified joystick will help us to control motor speed so when we down the joystick the speed of motor will be zero and as we increase in upper side it will increase speed. If we didn't set up of joystick module the speed of motor always remains 50 percentage and there will be possibility of drone crash. So basically joystick module used to control speed of motor.



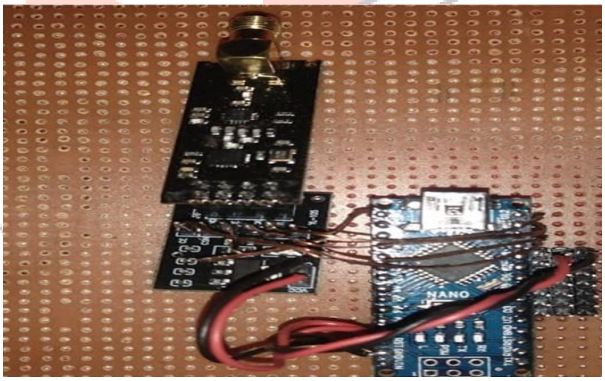


Fig. 10. Custom made Receiver

IV. CODES

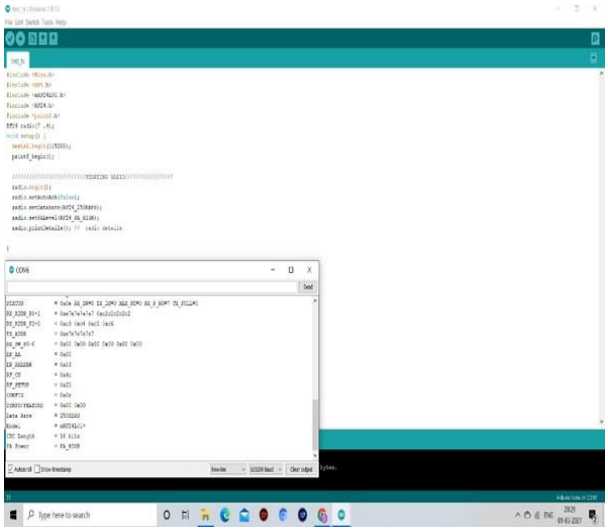


Fig. 11. Transmitter with output

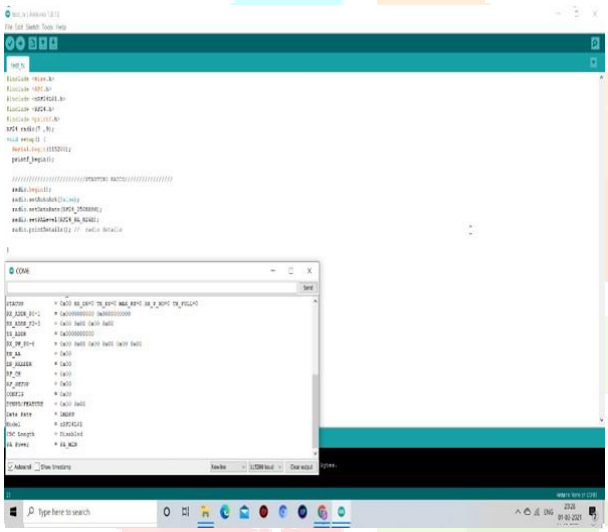


Fig. 12. Transmitter with no output

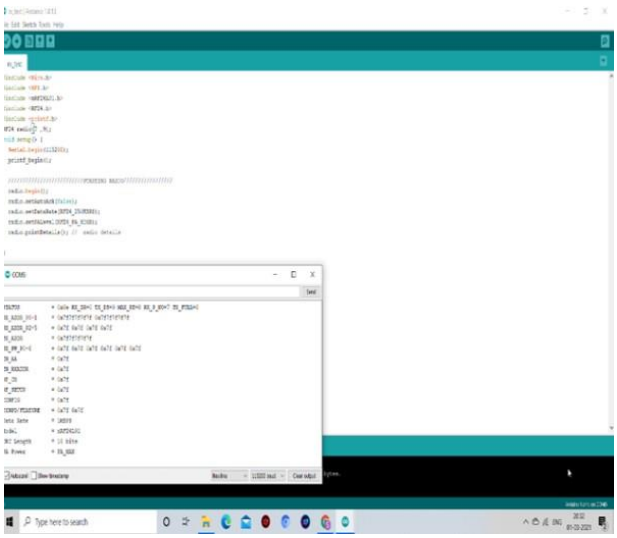


Fig. 13. Receiver with output

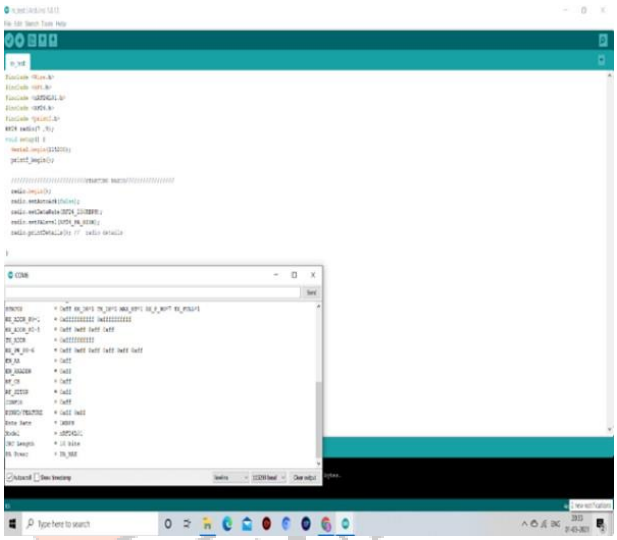


Fig. 14. Receiver with no output

V. TESTING

A. Motor Configuration

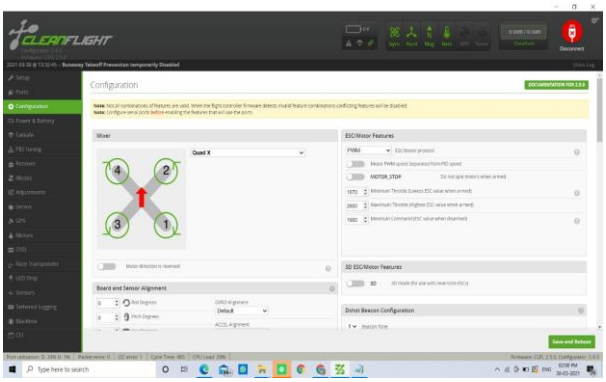


Fig. 15. Configuration with Cleanflight Software

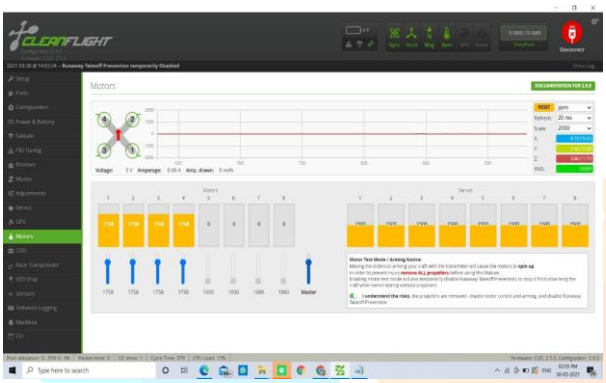


Fig. 16. Motors Configuration with Cleanflight Software

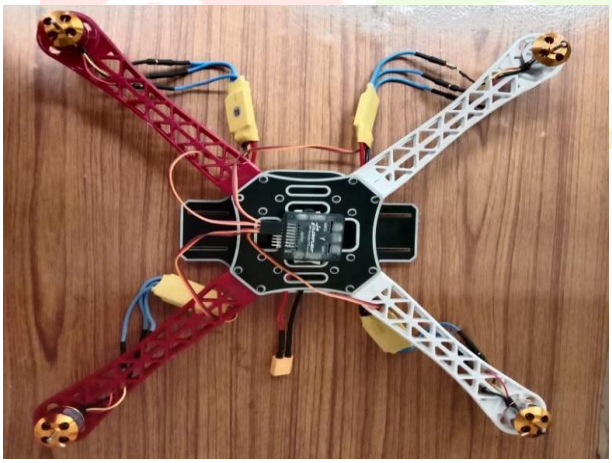


Fig. 17. Implimented Design of Drone

B. PID Setting

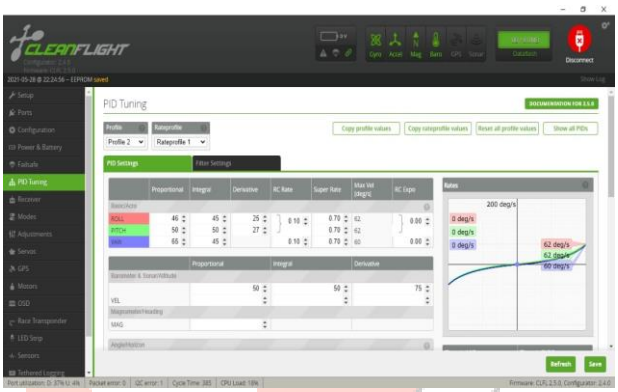
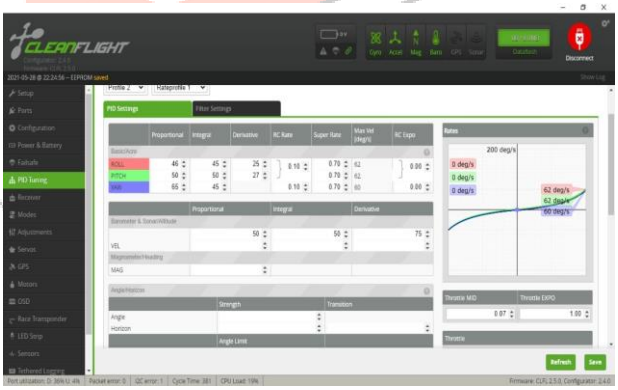


Fig. 18. PID Tunning



VI. ADVANTAGES

1)Rc communication works in any network.
2)Efficient drone controlling
3) Works on particular frequency.
4)6 channels we can modify according to our need
5)Cost is low

VII. APPLICATION

Rc system can be used in car ,boat , submarine ,walkie talkie,can be used in Disaster management

VIII. FUTURE SCOPE

We looked at a feasible way of constructing a basic model of Disaster management drone . It can be improved further to Video and image capture facility will be helpful for monitoring in during Natural calamities. In future we can aim to build a Drone with an advanced adaptive control system which is not restricted to a basic application but also telemetry and audio and video and multiple channels.These is a base model but our custom made RC controller benefit is it will work on particular frequency ,it don't interrupt channels frequency during communication.These is universal RC system. Using these we can make car,boat, submarine,walkie talkie.

IX. RESULT

A. Technical Details

Weight carrying capacity -3.1Gram
Range -1100meter/1km
Flying time - 15 min
distance -1100 meter Weight of drone -800 gram Total Channel -6

X. CONCLUSION

In this paper, we looked at a feasible way of constructing a basic model of Disaster management drone . It can be improved further to Video and image capture facility will be helpful for monitoring in during Natural calamities. In future we can aim to build a Drone with an advanced adaptive control system which is not restricted to a basic application but also telemetry and audio and video and multiple channels.

These is a base model but our custom made RC controller benefit is it will work on particular frequency ,it don't interrupt channels frequency during communication. These is universal RC system. Using these we can make car,boat, submarine, walkie talkie.

ACKNOWLEDGEMENT

It gives me immense pleasure to express my deep grat- itude and sincere thanks to Dr.Ms. Shikha Nema, H.O.D., and all faculty members from Department of Electronics and Telecommuniation, Usha Mittal Institute of Technology Mumbai for their valuable and useful support and comments for making this workshop a successful event. I'ld not forget to mention that their approach kept my working environment alive and their encouragement promoted me to do my task rigorously.

REFERENCES

[1] B. Kaplan, İ. Kahraman, A. Görçin, H. A. Ç ırpan and A. R. Ekti, "Measurement based FHSS–type Drone Controller Detection at 2.4GHz: An STFT Approach," 2020 IEEE 91st Vehicular Technology Con- ference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-6, doi: 10.1109/VTC2020-Spring48590.2020.9129525
[2] M. Donatti, F. Frazatto, L. Manera, T. Teramoto and E. Neger, "Radio frequency spoofing system to take over law-breaking drones," 2016 IEEE MTT-S Latin America Microwave Conference (LAMC), Puerto Vallarta, 2019, pp. 1-3, doi: 10.1109/LAMC.2016.7851290
[3] J. Marin, M. Heino, J. Saikanmäki, M. Mäenpää, A. -P. Saari- nen and T. Riihonen, "Perfecting Jamming Signals Against RC Sys- tems: An Experimental Case Study on FHSS with GFSK," 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, UK, 2020, pp. 1-5, doi: 10.1109/PIMRC48278.2020.9217129
[4] Jeremia, Stevie Kuantama, Endrowednes Pangaribuan, Julinda. (2018). Design and construction of remote-controlled quad-copter based on STC12C5624AD. International Conference on System Engineering and Technology (ICSET). 1-6. 10.1109/ICSEngT.2012.6339317
[5] Yamamoto, Naoki Naito, Katsuhiro. (2019). Proposal of Continuous Remote Control Architecture for Drone Operations. 10.1007/978-3-319-92231-7-7
[6] P. Sanjana and M. Prathilothamai, "Drone Design for First Aid Kit Delivery in Emergency Situation," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbat- ore,India,2020,pp.215-220,doi:10.1109/ICACCS48705.2020.9074487
[7] O. H. Graven, J. Srli, J. Bjrk, D. A. H. Samuelsen and J. D. Bjerknes, "Managing disasters-rapid deployment of sensor network from drones: Providing first responders with vital information," 2017 2nd International Conference on Control and Robotics Engineering (ICCRE), Bangkok, 2017, pp. 184-188, doi: 10.1109/ICCRE.2017.7935067
[8] Mahbub M (2019) Design and implementation of multipurpose radio controller unit using nRF24L01 wireless transceiver module and Arduino as MCU. Int Jr Robotic and Auto Eng: IJRAE-118
[9] M. Erdelj and E. Natalizio, "UAV-assisted disaster management: Appli- cations and open issues," 2016 International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 2016, pp. 1-5, doi:10.1109/ICCNC.2016.7440563.
[10] A. Y. Chung, J. Y. Lee and H. Kim, "Autonomous mission completion system for disconnected delivery drones in urban area," 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, 2017, pp. 56-61, doi: 10.1109/ROBIO.2017.8324394
[11] L. Apvrille, Y. Roudier and T. J. Tanzi, "Autonomous drones for disastersmanagement: Safety and security verifications," 1st URSI Atlantic RadioSci-enceConference(URSIAT-RASC),LasPalmas,pp.1-2,2015
[12] N. Nikhil, S. M. Shreyas, G. Vyshnavi and S. Yadav, "Unmanned Aerial Ve-hicles (UAV) in Disaster Management Applications," Third International Con-ference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli,India,pp.140-148,2020.

[13] C. Alex and A. Vijaychandra, "Autonomous cloud based drone system fordisaster response and mitigation,"International Conference on Robotics andAutomation for Humanitarian Applications (RAHA), Kollam, pp.1-4, 2016.
[14] O. H. Graven, J. Srli, J. Bjrk, D. A. H. Samuelsen and J. D. Bjerknes, "Managing disasters-rapid deployment of sensor network from drones: Providing firstresponders with vital information,"2nd International Con- ference on ControlandRoboticsEngineering(ICCRE),Bangkokpp.184- 188, 2017
[15] E. O'Keeffe, A. Moore and E. Mangina, "Drone-based Re-establishment ofCommunications for Humanitarian Rescue Organisations,"IEEE Radio andAntenna Days of the Indian Ocean (RADIO), Grand Port, pp. 1-2, 2018.