A PROJECT REPORT

ON

" Fly and Crawl Copter "

Submitted by

17ME15-Gharade Ammar

17ME16-Ghongade Pratik

17ME19-Jalgaokar Shehzad

17ME21-Kadam Mayur

In partial fulfillment for the award of the Degree

Of

BACHELOR OF ENGINEERING

IN

MECHANICAL ENGINEERING

UNDER THE GUIDANCE

Of

Prof. Aslam Hirani

DEPARTMENT OF MECHANICAL ENGINEERING

ANJUMAN-I-ISLAM

KALSEKAR TECHNICAL CAMPUS NEW PANVEL,

NAVI MUMBAI – 410206

UNIVERSITY OF MUMBAI

ACADEMIC YEAR 2021-2022

ANJUMAN-I-ISLAM

KALSEKAR TECHNICAL CAMPUS NEW PANVEL

(Approved by AICTE, recg. By Maharashtra Govt. DTE,

Affiliated to Mumbai University)

PLOT #2&3, SECTOR 16, NEAR THANA NAKA, KHANDAGAON, NEW PANVEL,NAVI MUMBAI-410206, Tel.: +91 22 27481247/48 * Website: www.aiktc.org

CERTIFICATE

This is to certify that the project entitled

“ **Fly and Crawl Copter** ”

Submitted by

17ME15-Gharade Ammar

17ME16-Ghongade Pratik

17ME19-Jalgaokar Shehzad

17ME21-Kadam Mayur

To the Kalsekar Technical Campus, New Panvel is a record of bonafide work carried out by him under our supervision and guidance, for partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Mechanical Engineering as prescribed by **University Of Mumbai**, is approved.

Internal Examiner | **External Examiner**

(Prof.______) | (Prof.____)

Head of Department | **Principal**

(Prof. _____) | (Dr._______)

ANJUMAN-I-ISLAM

KALSEKAR TECHNICAL CAMPUS NEW PANVEL

(Approved by AICTE, recg. By Maharashtra Govt. DTE,

Affiliated to Mumbai University)

PLOT #2&3, SECTOR 16, NEAR THANA NAKA, KHANDAGAON, NEW PANVEL,NAVI MUMBAI-410206, Tel.: +91 22 27481247/48 * Website: www.aiktc.org

APPROVAL OF DISSERTATION

This is to certify that the thesis entitled

" Fly and Crawl Copter "

Submitted by

17ME15-Gharade Ammar

17ME16-Ghongade Pratik

17ME19-Jalgaokar Shehzad

17ME21-Kadam Mayur

In partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Mechanical Engineering, as prescribed by University of Mumbai approved.

(Internal Examiner) **(External Examiner)**

___________________ ___________________

Date: __________

ACKNOWLEDGEMENT

After the completion of this work, we would like to give our sincere thanks to all those who helped us to reach our goal. It's a great pleasure and moment of immense satisfaction for us to express my profound gratitude to our guide **Prof. Aslam Hirani_** whose constant encouragement enabled us to work enthusiastically. His perpetual motivation, patience and excellent expertise in discussion during progress of the project work have benefited us to an extent, which is beyond expression.

We would also like to give our sincere thanks to **Prof. Zakir Ansari_** , Head Of Department, **Prof._Aslam Hirani** Project Co-Guide and **Prof. Atul meshram** Project co-ordinator from Department of Mechanical Engineering, Kalsekar Technical Campus, New Panvel, for their guidance, encouragement and support during a project.

I am thankful to **Dr._Abdul Razzak Honnutagi** , Kalsekar Technical Campus New Panvel, for providing an outstanding academic environment, also for providing the adequate facilities.

Last but not the least I would also like to thank all the staffs of Kalsekar Technical Campus (Mechanical Engineering Department) for their valuable guidance with their interest and valuable suggestions brightened us.

17ME15 GHARADE AMMAR RAFID

17ME16 GHONGADE PRATIK VIJAY

17ME19 JALGAONKAR SHEHZAD SHAWKAT

17ME21 KADAM MAYUR RATAN

ABSTRACT

Originally created by nir meiri and David Zarrouk ,and named as “Flying STAR, a Hybrid Crawling and Flying Sprawl Tuned Robot”. And was presented at “2019 International Conference on Robotics and Automation(ICRA).

Flying STAR (FSTAR) a recognizable hybrid flying quadcopter robt. FSTAR is a quad copter fitted with sprawling mechanism and propellers allowing it to both run and fly over obstacles or run underneath obstacles while touching the ground. The robot can reduce it’s width to crawl in confined obstacles while touching the ground.

We first described the idea behind this project in which design, mechanism and manufacturing details of components also the hardware and software requirements & it’s specifications also the calculations for selecting components, force strategy for flying quadcopter as well mechanism for crawling over land also discussed quadcopter coordinate systems & Electronic hardware computation.

Since the project is mostly designed for military purpose to camouflage/ hide & proceed further either by air or by land .

Table of content

Chapter 1
Introduction

Since the drone technology is new to is new to us ,even though in very short span ,this technology had made massive progress ,As today we can see the drones almost everywhere not only monitoring or inspection but also in film industry ,mining agriculture , environmental research, military defense or attacks ,drone technology has its own variety applications.

As traditional quadcopter can fly as per remote input given to it, and if camera attached it can take photos of desired fields .

But advancement in drone technology didn't keep the drone industry limited to flying only , technology advanced further and made the drone travel by land as well as dive deep into ocean or all combined motions together.

So now drone not only fly but also can crawl / run over land as well as cam swim through waters.

1.1 Objective of project

Since the industry of drone technology is rapidly expanding by that means the need of drones in society or industries is increasing, Also as per further demands more and more advancement is more likely to be expected, the drones that are attached with wheel or the drone that are attached with wheel or the drone that can go by land whenever possible are great utility.

In military operations wherever it ia needed this robot can travel by land as a guerrilla warfare and proceed further for enemy position and cam send the confidential details of enemy camp.

As this quadcopter by its sprawling mechanism can change its size or orientation it can pass various obstacles so being a drone it is more likely to be progressive copter that can pass over hindrance of any type & proceed further.

The quadcopter has both combined running and flying capabilities.

1.2 Literature review

Name of paper	Author	Year of Publishing	Inferences taken from paper
Flying STAR, a Hybrid Crawling & Flying Sprawl Tuned Robot	Nir Meri and David Zarrouk	2019	entire concept of project
Kinematic, Dynamic modelling and	Graduates of AIKTC ,new panvel	2016	PID control, graph of PID control, tuning

simulation of quadcopter			
Optimal motion planning for multi modal hybrid locomotion	H.J Jerry suh, Xiaobin Xiong	2019	flying driving robot could take different pathways
Quadcopter control using Arduino microcontroller		2018	Block diagram

Chapter 2
Hardware & Software requirements

2.1 Hardware requirements: -

- RS2205 2300 KV Brushless DC Motor (2CW,2CCW)
- Quadcopter frame (Material:- Black Carbon Fibre)
- Flight controller- SP Racing TE Flight Controller Integrate OSD ACRO version
- 2x pair of 3×5.2 inch propeller (2cw & 2ccw)
- 4x ESCs (30A) Simon K
- Arduino
- Servo Motor 180degree
- 2200mA 12v Lipo battery
- Arduino Bluetooth Module
- IMU Sensor
- HC05 Bluetooth Module

2.2 Software requirement -

- Clean Flight software
- Arduino IDE

Specifications and working of Hardware Components

4x Brushless DC motors(RS2205 2300 kV)

How to read BLDC motor label

A **'2300kv'** printed indicates R.P.M/volt and **2205** number is divided in two parts '22' indicates diameter of motor and '05' indicates height of motor. '/13T' represent the winding higher the T value more will be torque generated by motor.

Calculation for selection of BLDC motor

Selection of BLDC motor is very important parameter to be consider. It is the first step for designing quadcopter.

1. Selecting speed of motor:- a 2300kv speed motor is sufficient enough for our applications. To calculate the max. speed produce by BLDC motor we multiply **K.V x VOLT** in this case **2300kv** motor will produce **2300r.p.m**at **1V** and produce

27600 r.p.m for **12v**

2. Calculating torque produce by motor to generate thrust :-torque produce by motor should be enough to generate thrust to lift done general formula for thrust is,

Thrust required = 2.2 * the weight of drone(in kg)

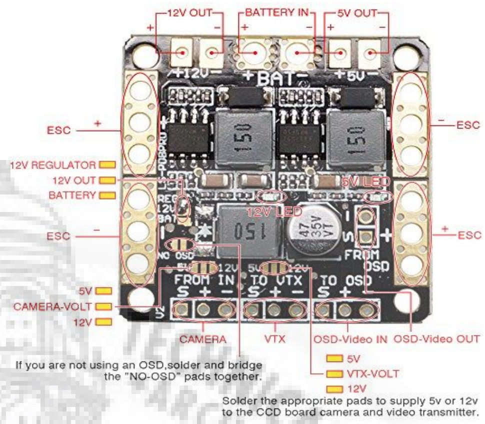
BLDC 2300kv single motor torque generates **1024gm force**
So total torque by 4xBLDC motors will around **4.1kg force**

Quadcopter frame Q

The Frame is a well thought out 300mm quad frame built from quality materials. The main frame is black carbon fiber.

Specs:
Width: **300mm**
Length : **250 mm**
Height: **55mm**
Weight: **200g (w/out electronics)**
Material :- **Black carbon fibre**

Flight controller:- SP Racing F3 Flight Controller Integrate OSD ACRO Version



Calculation for selecting propellers

Before selecting propellers, we have to first calculate the amount of thrust can be produced by all propellers. Thrust produced by a propeller is given by, We

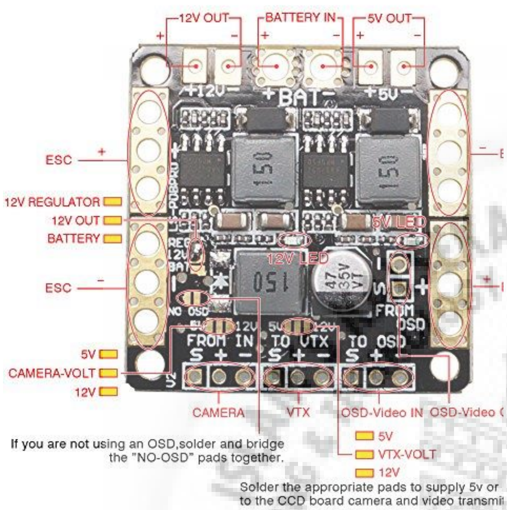
$$T = \frac{\pi}{4} D^2 \rho v \Delta v \qquad (2)$$

T=thrust [N]
D=propeller diameter [m]
v=velocity of air at the propeller [m/s]
Δv=velocity of air accelerated by propeller [m/s]
ρ=density of air [1.225 kg/m³]

can see in above equation diameter is very important parameter for selection propellers. So after calculation the suitable propeller came to be **3×5.2 .**Here 10 indicates the **diameter** of propeller and 3 indicate **pitch.** Generally large diameter give more thrust and **pitch** value give speed of **revolution per inch** higher the pitch value more speed. The proper selection is 3×5.2.

CC3D NAZE32 F3 POWER DISTRIBUTION BOARD

Features:-



Specification

Model	PDB With LC Filter & Dual BEC
BEC output	5V/12V 3A
Input Voltage	2-6S (max. 3A)
Length (mm)	36
Width (mm)	36
Mounting Distance(mm)	30.5
Weight (gm)	9

1. Latest UBEC, LC Filter
2. Integrated filter circuit boards, more stable, make FPV further and clearer
3. Compatible with CC3D/Naze32/F3 flight controller
4. Slim design, lightweight, and easy to use

Four ESCs(Electronic speed controller) LP

An electronic speed control or ESC is an electronic circuit with the purpose to vary an electric motor's speed, its direction and possibly also to act as a dynamic brake. ESCs are often used on electrically powered radio controlled models, with the variety most often used for brushless motors essentially providing an electronically generated three-phase electric power low voltage source of energy for the motor.

Selecting esc:

The 2300kv BLDC motor discuss above draws max **30A current**, so we have to choose the esc current rating of bit higher to avoid heating and damage to esc. The above ESC is a programmable and has **SIMON K** firmware, the best firmware for multi rotor vehicles. In the next chapter there is detail explanation of programing esc and calibrating esc.

Servo Motor:-TowerPro MG995

Model: MG995,

Gear Type: Metal

Rotational Degree: 180°

Operating Voltage: 4.8 – 7.2 V

Operating Speed at 4.8V: 20sec/60°

Operating Speed at 6.6V: 16sec/60°

Stall Torque at 4.8V: 10 kg-cm

Stall Torque at 6.6V: 12 kg-cm

Operating Temperature: -30 to 60°C

Dead Band Width: 5µs

Cable Length: 30mm

High resolution

Accurate positioning

Fast control response

Constant torque throughout the servo travel range

Excellent holding power

Color: Black

Motor Size:

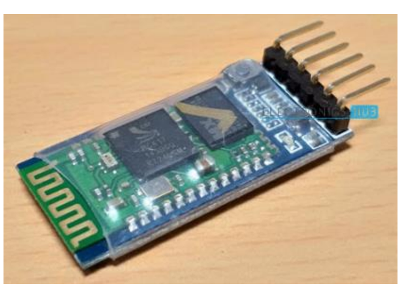
Length: 40.5mm

Width: 20mm

Height: 44mm

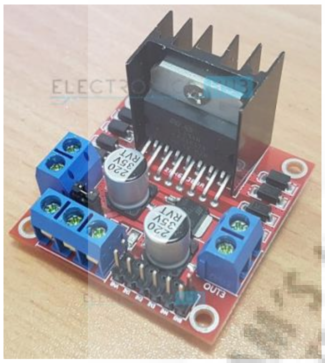
Weight: 60gm

HC05 – Bluetooth Module



The HC-05 Bluetooth Module is responsible for enabling Bluetooth Communication between Arduino and Android Phone.

L298N Motor Driver Module

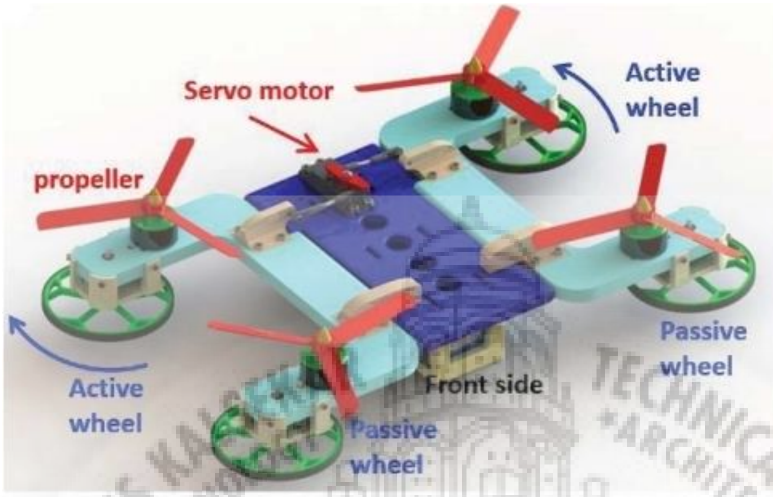


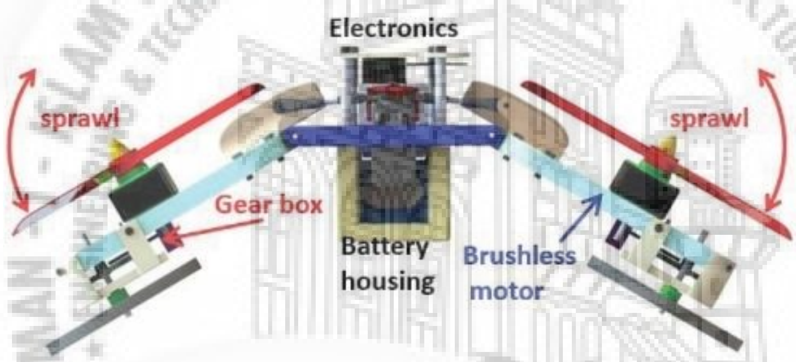
The L298N Motor Driver Module is responsible for providing the necessary drive current to the motors of the Fly and Crawl Quadcopter.

Chapter 3
Design and Manufacturing

The primary design goals of Fly and crawl Quadcopter is to achieve high performance in both crawling and flying, and its weight is lowered to increase maneuverability and its payload of sensors needed for search and rescue missions. In parallel ,the robot has to be energy efficient so as to increase its work range and work time without recharging.

The frame had made the housing in which the batteries and controllers are mounted, the sprawl mechanism which is actuated by a servo motor that simultaneously actuate the propellers and the wheels are tilted together with sprawl mechanism. Both sides of the sprawling mechanism are phased together and move symmetrically relative to its center body. When in contact with the ground, the back wheels provide propulsion during running whereas the forward wheels are passive.

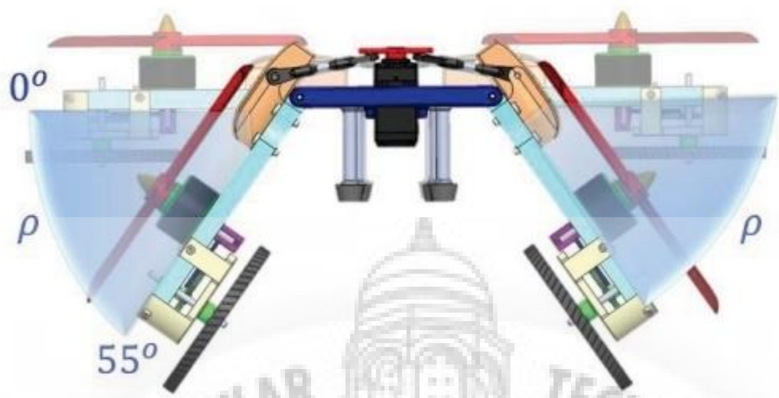




3.1.The Sprawling mechanism

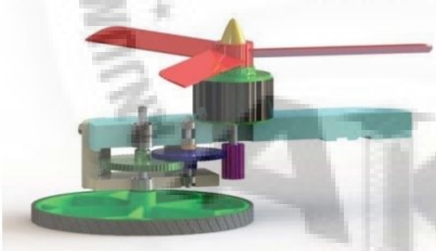
The sprawl angle (rho) is the relative angle between the plane of the wheels (and propellers) and the main body, as presented in Figure below. The sprawl angle is defined as rho=0 when the plane of the wheels is parallel to the body and the positive sense of the sprawl angle is downwards.

The sprawl angles at both sides are actuated symmetrically through the servo motor to insure an identical sprawl on both sides of the robot. The sprawl angle in this design can be varied in the range of 0 to 55 degrees as shown in Figure shown . The positive sense of the sprawl angle is defined downwards.



3.2. The motor housing and gear boxes.

The propellers of the robot are directly attached to the axis of the motors. The front wheels are passive (unactuated) whereas the back wheels are actuated using the same motors of the quadcopter via a 20:1 spur gearbox which we designed to reduce the speed of the wheels and increase the torque. The radius of the wheels is 3.85 cm and we used 7.62cm (5 inches) long, 13.2 cm pitch (5.2 inches) three-blade propellers.



The active wheels motor housing and gearboxes with 20:1 gear ratio.

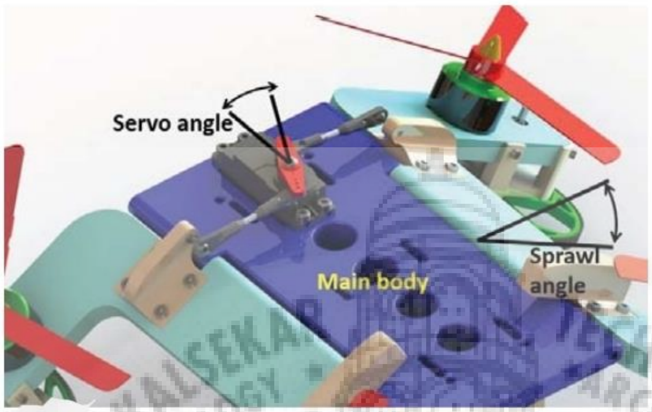
3.3. Actuation and control

The quadcopter is actuated using four bldc motors for flying and one servo motor that actuates the sprawl mechanism. In this design we used RS2205 2300KV brushless dc motors, the motor weighs 30 gm and can generate 1024 gm thrust .

The servo motor TowerPro MG995 with rotational range of 180 degrees, weight 55 gm and produces 10 kg-cm torque.

We used SP Racing F3 Flight Controller and it fitted with IMU sensor.

The speed controller is Simonk 30A BLDC ESCs .



. The sprawl mechanism, actuated using a servo motor, has a range of 55 degrees.

Chapter 4

Dynamic analysis

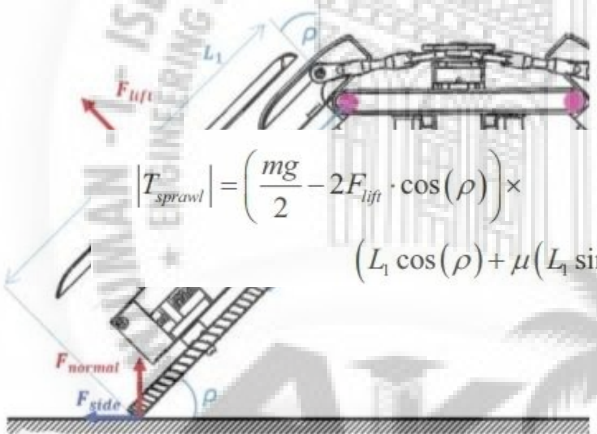
Force and Torque analysis

In this section, we calculate the forces acting on the robot and the torques that must be provided by the different motors of the robot when running over a horizontal surface and in flight mode. For the sprawling mode, we analyze the cases which require larger torques; i.e., the robot increases its sprawl and therefore lifts its center of mass (COM). We denote by Fnormal, Fside, the absolute values of the normal and side forces acting on one side of the legs,
The normal force Fnormal acting on the legs (assuming low acceleration) is:

$$|F_{normal}| = \frac{mg}{2} - 2F_{lift} \cdot \cos(\rho)$$

Where m is the total mass of the robot. The side force Fside when the robot is extending its sprawl is:

$$|F_{side}| = \mu\left(\frac{mg}{2} - 2F_{lift} \cdot \cos(\rho)\right)$$



. The forces acting on the robot during sprawling up. The normal force and the friction side forces resist the motion.

Note that Fside is pointed outwards when the robot increases the sprawl angle and inwards when it decreases the sprawl. A force diagram of the robot is presented in Figure . The torque acting on the sprawl joint Tsprawl is a function of the sprawl angle (rho) and whether or not the propellers are providing lift :

$$|T_{sprawl}| = \left(\frac{mg}{2} - 2F_{lift} \cdot \cos(\rho)\right) \times \left(L_1 \cos(\rho) + \mu\left(L_1 \sin(\rho) + L_2 \cos(\rho)\right)\right)$$

Where Flift is the lifting force generated by a single propeller. The lengths L1=10.3 cm and L2=2.9 cm are defined respectively as the distances from the sprawl joint to the ground in parallel and vertical to the sprawl arm (see Figure). The maximum torque required by the servo motor is when the sprawl is almost zero . By inserting Eq. (1) we obtain:

$$\left|T_{servo}\right| = 2\times 0.78\cdot\left(\frac{mg}{2} - 2F_{lift}\cdot\cos(\rho)\right)\times \left(L_1\cos(\rho) + \mu\left(L_1\sin(\rho) + L_2\cos(\rho)\right)\right)$$

The torque required by the servo motor is clearly at its maximum when the lifting force is zero. In this case, assuming that the coefficient of friction (COF) μ is 0.3 (plastic contact with a tile floor), the maximum torque generated by the servo Tservo is 78.4 Ncm (nearly half of the torque which the servo motor can produce).

The torque generated by the wheels is:

$$T_{wheels} = 20\cdot\left(T_{motor} - T_{propeller}\right)$$

where Tmotor is the torque that the motor can generate and the Tpropeller is the torque required to rotate the propellers. Therefore the combined thrust force that the two active wheels can generate, assuming sticking to the surface, is:

$$F_{thrust} = 2\frac{20\cdot\left(T_{motor} - T_{propeller}\right)}{R_{wheels}}$$

At speeds below 1m/s, the torque of the propellers Tpropellers can be neglected. Therefore, based on our parameters and assuming Tmotor=0.05 Nm, the thrust force is 52 N. As the robot increases its speed, the lift generated by the propellers increases as a function of the speed squared. If the total lift force exceeds the weight, the robot will start flying. The total lift force as a function of the sprawl angle is:

$$F_{lifit\_tot} = 4F_{lift}\cos(\rho)$$

Since the weight of the robot is nearly 9N, the required lift force at a zero sprawl angle by each motor is 2.25 N, which occurs at nearly 14000 RPM. By

Eq. (2), this speed corresponds to nearly 2.82 m/s running. Inversely, the maximum sprawl angle which will still allow flying can also be calculated using:

$$\rho = \text{acos}\left(\frac{mg}{4\max(F_{lift})}\right)$$

The maximum lift force using the four motors is obtained at 18000 RPM for a value of 12.4 N. By inserting into Eq. (10), we obtain a maximum sprawl angle of 43.4 degrees.

Chapter 5
Quadcopter working principle

The basic working of quadcopter drone is divided in to sub parts 'MECHANICS' & 'Electronic hardware computation.

5.1Mechanical principles

Multirotor aerial vehicles are available in different models viz, Tricopter, quadcopter, hexacopter and octocopters. In this project we have chosen a 'Quadcopter'. Again quadcopters came in two orientations 'x' type and '+' type and our copter is 'x' type frame.

· Drones are highly unstable in air due various factor affecting it like strong wind, magnetic field, air pressure etc. So it required a special flight controller which can handles and control the deformation of drone orientation with help of on-board sensors.

· Quadcopters generally use two pairs of identical fixed pitched propellers; two clockwise (CW) and two counter-clockwise (CCW). These use independent variation of the speed of each rotor to achieve control. By changing the speed of each rotor it is possible to specifically generate a desired total thrust, to locate for the Centre of thrust both laterally and longitudinally; and to create a desired total torque, or turning forces

The opposite pair of propeller rotates in same direction i.e 1-2 propeller rotate in counter clock wise while 3-4 propeller will rotate at clockwise direction. This particular orientation will result upward thrust to lift drone in air.

5.1.1 Force strategy for flying drone

A proper amount of force must be generated by individual propellers to make quadcopter climb, hover and decline. Here involve some calculations to achieve balance amount of force to lift the quadcopter.

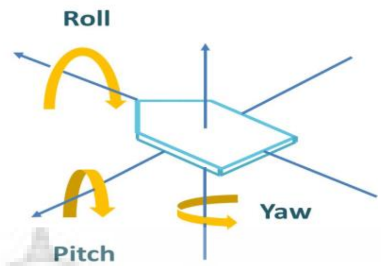
In a quad copter **F1, F2, F3, F4** are individual forces generated by respective propeller. **(Fi)** is the total amount of force act in same direction. Also the weight of quadcopter and gravitation force acting on it results in downward force and given by m*g.

- SUM(Fi) > m*g <=>climb
- SUM(Fi) = m*g <=>hover
- SUM(Fi) < m*g <=>decline

If the total force 'Fi' is **greater** than the downward force 'm*g' then a resultant force will be in **upward** direction and drone will climb. If total force 'Fi' is equal to the downward force 'm*g' then drone will hover and if the total force 'Fi' is **less** than the downward force 'm*g' then a resultant force will be in **downward** direction and drone will decline

5.1.2 Quadcopter Coordinate System

The quadcopter orientation can be defined by three angles: ROLL,PITCH and YAW. This angle is best understood with help of coordinate system as shown in fig.a.These angles determine orientation and therefore the direction the quadcopter will take.



• ROLL: - roll axis is shown in fig.a, this angle describes how the quadcopter is tilted **side to side**. Rotation about the roll axis is like tilting our head towards one of your shoulders. Rolling the multirotor causes it to move sideways.

• PITCH: - The pitch angle of the multirotor describes how the quadcopter is tilted **forwards or backwards** direction. Rotation about the pitch axis is like tilting our head in order to look up or down. Pitching the multirotor causes it to move forwards or backwards.

· YAW: - The yaw angle of the quadcopter describes its bearing, or, in other words, rotation of the quad perpendicular to the ground. Its is a vertical axis in coordinate system. Rotation about the yaw axis is like when we shake our head to say "no."

5.1.3Concept of changing ROLL, PITCH & YAW angle

Roll and Pitch:

To make the multirotor rotate about the roll or pitch axes, the flight controller makes the motors on one side of the multirotor spin faster than the motors on the other side. This means that one side of the multirotor will have more lift than the other side, causing the multirotor to tilt.

So, for example, to make a quadcopter roll right (or rotate about the roll axis clockwise), the flight controller will make the two motors on the left side of the multirotor spin faster than the two motors on the right side. The left side of the craft will then have more lift than the right side, which causes the multirotor to tilt.

Similarly, to make a quadcopter pitch down (rotate about the pitch axis clockwise) the flight controller will make the two motors on the back of the craft spin faster than the two motors on the front. This makes the craft tilt in the same way that your head tilts when you look down.

Yaw:

Controlling the quadcopter rotation about the yaw axis is a bit more complex than controlling its rotation about the roll or pitch axes. First, let's discuss how we *prevent* rotation about the yaw axis. When assembling and programming quadcopter, we set up the motors so that each motor spins in the opposite direction than its neighbors. In other words, using a quadcopter as an example again, starting from the front-left motor and moving around the multirotor clockwise, the motors' rotational directions alternate, **CW, CCW, CW, CCW**. We use this rotational configuration to neutralize, or cancel out, each motor's tendency to make the multirotor rotate.

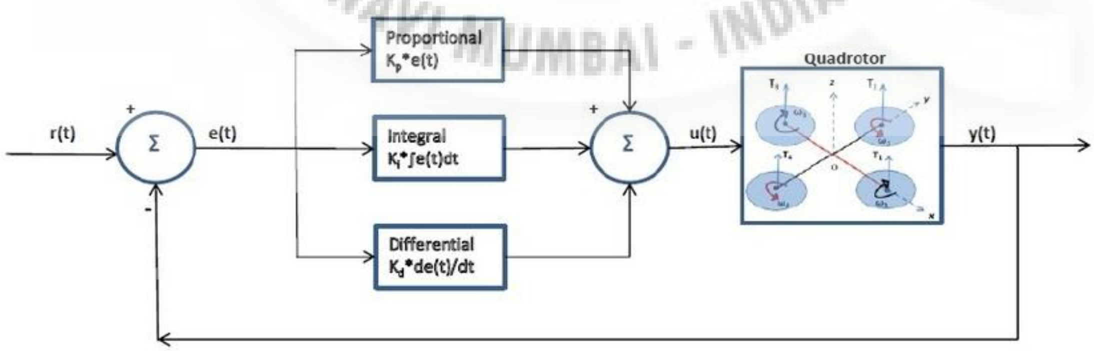
The quadcopter rotate yawby slowing down different pairs of motors

5.2Electronic hardware computation

Electronic hardware computation involves adjusting the PID value of processor to make quadcopter stabilize as much as possible. **CleanFlight** software allow users to change PID configuration values to adjust the performance of their quadcopters. In this post i will try to explain briefly what is PID, how does it affect the stability of the quadcopter (or multicopter), and how to tune the PID for quadcopter.

5.2.1 What is PID?

PID (proportional-integral-derivative) is a closed-loop control system that try to get the actual result closer to the desired result by adjusting the input. Quadcopters or multicopters use PID controller to achieve stability.



There are 3 algorithms in a PID controller, they are P, I, and D respectively. P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. These controller algorithms are translated into software code lines.

To have any kind of control over the quadcopter, we need to be able to measure the quadcopter sensor output (for example the pitch angle), so we can estimate the error (how far we are from the desired pitch angle, e.g. horizontal, 0 degree). We can then apply the 3 control algorithms to the error, to get the next outputs for the motors aiming to correct the error.

There are three parameters that a pilot can adjust to improve better quadcopter stability. These are the coefficients to the 3 algorithms we mentioned above. The coefficient basically would change the importance and influence of each algorithm to the output. Here we are going to look at what are the effects of these parameters to the stability of a quadcopter.

The effect of each parameter

The variation of each of these parameters alters the effectiveness of the stabilization. Generally, there are 3 PID loops with their own P I D coefficients, one per axis, so wewill have to set P, I and D values for each axis (pitch, roll and yaw).

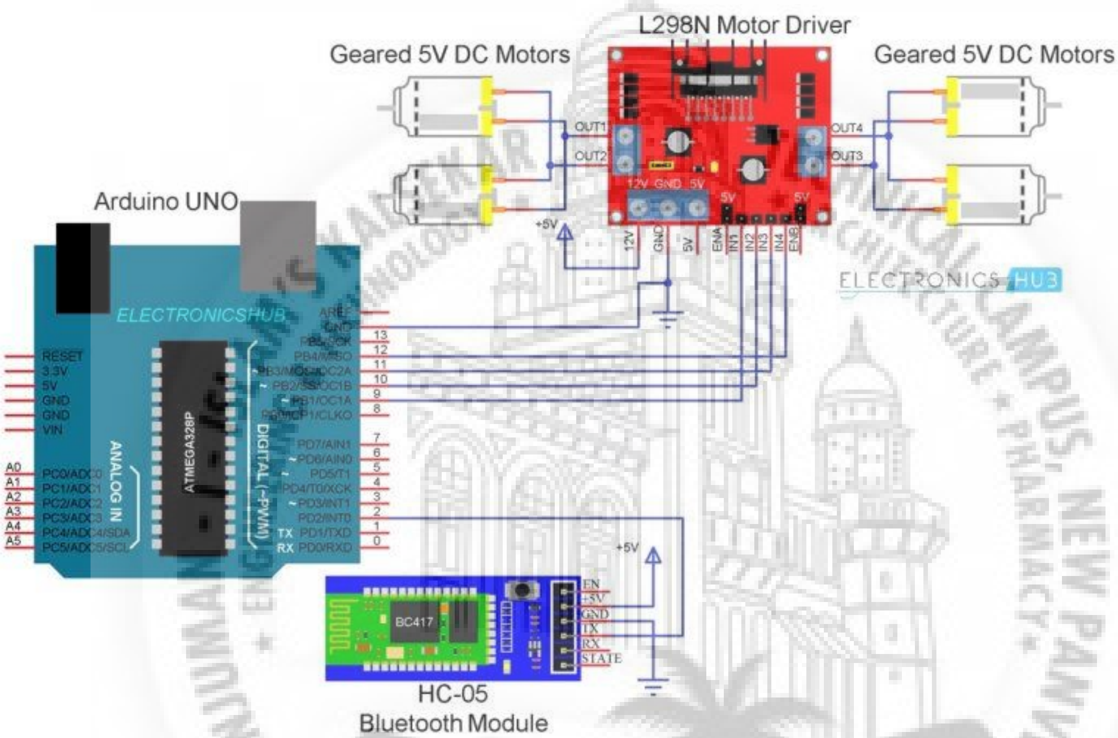
For a quadcopter, these parameters can cause these behaviours.

- **Proportional Gain coefficient** –quadcopter can fly relatively stable without other parameters but this one. This coefficient determines which is more important, human control or the values measured by the gyroscopes. The higher the coefficient, the higher the quadcopter seems more sensitive and reactive to angular change. If it is too low, the quadcopter will appear sluggish and will be harder to keep steady. the quadcopter starts to oscillate with a high frequency when P gain is too high.

- **Integral Gain coefficient** – this coefficient can increase the precision of the angular position. For example when the quadcopter is disturbed and its angle changes 20 degrees , in theory it remembers how much the angle has changed and will return 20 degrees. In practice if you make your quadcopter go forward and the force it to stop, the quadcopter will continue for some time to counteract the action. Without this term, the opposition does not last as long. This term is especially useful with irregular wind, and ground effect (turbulence from motors). However, when the I value gets too high your quadcopter might begin to have slow reaction and a decrease effect of the Proportional gain as consequence, it will also start to oscillate like having high P gain, but with a lower frequency
- **Derivative Gain coefficient –** this coefficient allows the quadcopter to reach more quickly the desired attitude. Some people call it the accelerator parameter because it amplifies the user input. It also decreases control

action fast when the error is decreasing fast. In practice it will increase the reaction speed and in certain cases an increase the effect of the P gain.

Block Diagram



6.1 Quadcopter to be controlled using Bluetooth module

```
#include <SoftwareSerial.h>
SoftwareSerial BT(2, 3);
// creates a "virtual" serial port/UART
// connect BT module TX to D2
// connect BT module RX to D3 with voltage dividers
// connect BT Vcc to 5V, GND to GND

/* For Quadcopter (Motors placed and numbered clock-wise):-
  PWM output 1 -> 11 (Motor 1)
  PWM output 2 -> 10 (Motor 2)
  PWM output 3 -> 9 (Motor 3)
  PWM output 4 -> 6 (Motor 4)
*/


// Declare motor connections for copter (pwm output only)
int MotorOne = 11;
int MotorTwo = 10;
int MotorThree = 9;
int MotorFour = 6;

// Declare speed of motors (0 to 255)
int SpeedOne = 0;
int SpeedTwo = 0;
int SpeedThree = 0;
int SpeedFour = 0;

// Common Speed of Motors (0 to 255)
int AllSpeed = 0;

// Addition / Subtraction Factor: In alignment with the mobile app - Speed Up /
Down;
// Adds / Subtracts 5 between 0 to 100 giving 20 instances of changes.

int SpeedFactor = 12; // Change factor of speed when speed is increased or
decreased
```

```
void setup()
{
 // set digital pin to control as an output
 pinMode(MotorOne, OUTPUT);
 pinMode(MotorTwo, OUTPUT);
 pinMode(MotorThree, OUTPUT);
 pinMode(MotorFour, OUTPUT);

 // Declare movement controls
 void speedUp();
 void speedDown();
 void forward();
 void backward();
 void strafeLeft();
 void strafeRight();
 void curveLeft();
 void curveRight();
 void stabilize();

 // set the data rate for the SoftwareSerial port
 BT.begin(9600);
}
char a; // stores incoming character from other device
void loop()
{
 if (BT.available())
   // if text arrived in from BT serial...
 {
   a = (BT.read());

   if (a == 'h') // Speed Up
   {
     speedUp();
   }
   if (a == 'i') // Speed Down
   {
     speedDown();
   }
   if (a == 'b') // Forward
   {
     forward();
   }
   if (a == 'c') // Backward
```

```
    {
     backward();
    }
    if (a == 'd') // Strafe Left
    {
     strafeLeft();
    }
    if (a == 'e') // Strafe Right
    {
     strafeRight();
    }
    if (a == 'f') // Curve Left
    {
     curveLeft();
    }
    if (a == 'g') // Curve Right
    {
     curveRight();
    }
    if (a == 'a')// a and 'a' are different. One is a variable and the other is the input
  type. Function used to stabilize motor speeds
    {
     stabilize();
    }
    // you can add more "if" statements with other characters to add more
  commands
   }



  }

  void speedUp() {
   if (AllSpeed < 240) // Max pwm 255 for 5 volts.
   {
    AllSpeed = AllSpeed + SpeedFactor;
    SpeedOne = AllSpeed;
    SpeedTwo = AllSpeed;
    SpeedThree = AllSpeed;
    SpeedFour = AllSpeed;
    analogWrite(MotorOne, SpeedOne);
    analogWrite(MotorTwo, SpeedTwo);
    analogWrite(MotorThree, SpeedThree);
```

```
    analogWrite(MotorFour, SpeedFour);
  }
 }

 void speedDown() {
  if (AllSpeed > 0)
  {
    AllSpeed = AllSpeed - SpeedFactor;
    SpeedOne = AllSpeed;
    SpeedTwo = AllSpeed;
    SpeedThree = AllSpeed;
    SpeedFour = AllSpeed;
    analogWrite(MotorOne, SpeedOne);
    analogWrite(MotorTwo, SpeedTwo);
    analogWrite(MotorThree, SpeedThree);
    analogWrite(MotorFour, SpeedFour);
  }
 }

 void forward() {
  SpeedOne = AllSpeed + SpeedFactor;
  SpeedFour = AllSpeed + SpeedFactor;
  analogWrite(MotorOne, SpeedOne);
  analogWrite(MotorFour, SpeedFour);
 }

 void backward() {
  SpeedTwo = AllSpeed + SpeedFactor;
  SpeedThree = AllSpeed + SpeedFactor;
  analogWrite(MotorTwo, SpeedTwo);
  analogWrite(MotorThree, SpeedThree);
 }

 void strafeLeft() {
  SpeedThree = AllSpeed + SpeedFactor;
  SpeedFour = AllSpeed + SpeedFactor;
  analogWrite(MotorThree, SpeedThree);
  analogWrite(MotorFour, SpeedFour);
 }

 void strafeRight() {
  SpeedOne = AllSpeed + SpeedFactor;
  SpeedTwo = AllSpeed + SpeedFactor;
```

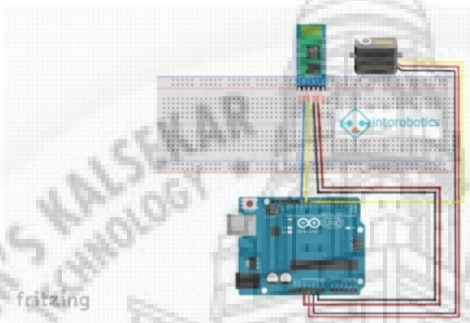
```
 analogWrite(MotorOne, SpeedOne);
 analogWrite(MotorTwo, SpeedTwo);
}

void curveLeft() {
 SpeedOne = AllSpeed + SpeedFactor;
 SpeedThree = AllSpeed + SpeedFactor;
 analogWrite(MotorOne, SpeedOne);
 analogWrite(MotorThree, SpeedThree);
}

void curveRight() {
 SpeedTwo = AllSpeed + SpeedFactor;
 SpeedFour = AllSpeed + SpeedFactor;
 analogWrite(MotorTwo, SpeedTwo);
 analogWrite(MotorFour, SpeedFour);
}

void stabilize() {
 AllSpeed = min(min(SpeedOne, SpeedTwo), min(SpeedThree, SpeedFour));
 SpeedOne = AllSpeed;
 SpeedTwo = AllSpeed;
 SpeedThree = AllSpeed;
 SpeedFour = AllSpeed;
 analogWrite(MotorOne, SpeedOne);
 analogWrite(MotorTwo, SpeedTwo);
 analogWrite(MotorThree, SpeedThree);
 analogWrite(MotorFour, SpeedFour);
                                    }
```

after pressing power up button, the pwm signal will raise Voltage thus increasing the power supplied to motors. In principle the quadcopter must take off after a certain speed has been attained.

Increase or decrease of speed will take place by addition / subtraction of 12. However, whenever a button is unpressed, a signal “a” will be sent to Arduino telling it to stabilize.

Increase or decrease of speed will happen up to 20 times only. Once the count increases/decreases in number of signals for speed control, the speed will increase or decrease.

6.2 Control Servo Motor with Bluetooth Module, Arduino and Android



How to setup the SG90 servo motor with Arduino UNO and the HC-06 Bluetooth module

Requirements:-

- 1 X Arduino UNO – the Bluetooth module is compatible with almost any Arduino model, but all the code and schematics in this tutorial are for UNO.
- 1 X HC-06 – this is a slave Bluetooth module very easy to use with Arduino using serial communication.
- 1 X SG90 Servo Motor – this is probably the most popular servo motor in the DIY community.
- 7 X male to male jumper wires
- 1 X breadboard
- 1 X Android smartphone having **ArduinoRC** application

To use the HC-06 module, simply connect the VCC pin to the 3.3V output on the Arduino, the GND pin to any of Arduino GND pins, then connect the TX pin of the Bluetooth module to pin 10 of Arduino UNO and RX pin of Bluetooth to pin 11 of Arduino.

For servo motor, connect the brown wire to any of Arduino GND pins, the red wire from the SG90 servo to the 5V output of the

Arduino, and the orange wire from the servo motor to digital pin 9 of Arduino.

Arduino Sketch and AT Commands

If the Bluetooth module is being used for the first time, you have to interrogate it to change some of the settings. The settings are changed with so-called AT commands.

The HC-06 module allows you to change a limited number of settings. You can change the name of the device, the PIN, and the baud rate.

You have to run the below AT commands in the IDE used with Arduino. These commands show you the version of firmware installed on the HC Bluetooth module, change the PIN, change the name of the module, and set the baud rate at 9600.

```
/**

 * @file        How To Control a Servo Motor With a Bluetooth Module, Arduino and Android

 * @author      Calin Dragos for intorobotics.com

 * @version     V1.0

 * @date        13.12.2016

 * @description  This is an Arduino sketch to setup the HC-06 Bluetooth module

 */


#include


#define ROBOT_NAME "Intorobotics"
```

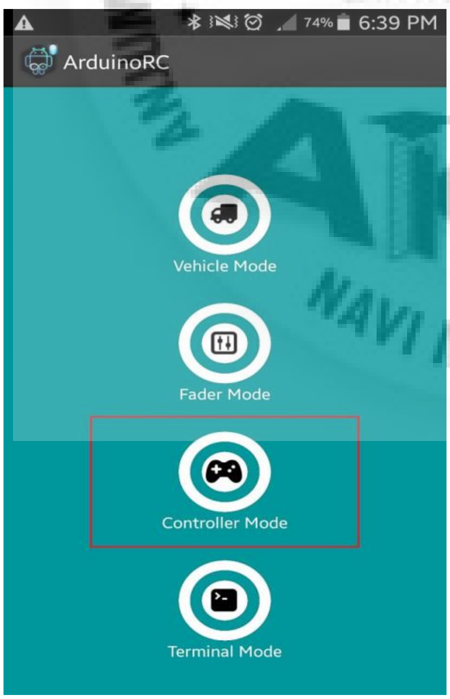
```
#define BLUETOOTH_SPEED 9600 //This is the default baud rate
that HC-06 uses

SoftwareSerial mySerial(10, 11); // TX | RX

// Connect the HC-06 TX to Arduino pin 10 RX.

// Connect the HC-06 RX to Arduino pin 11 TX.

void setup() {

   Serial.begin(9600);

  Serial.println("Starting the configurations!");

  mySerial.begin(BLUETOOTH_SPEED);

  delay(1000);

  // Should respond with OK

  Serial.print("AT test command is: ");

  mySerial.print("AT");

  waitResponse();

  Serial.println("----------------");
```

```
// Should respond with its version
Serial.print("AT version is: ");
mySerial.print("AT+VERSION");
waitResponse();

Serial.println("----------------");
// Set pin
Serial.print("Set pin: ");
mySerial.print("AT+PIN1234");
waitResponse();

Serial.println("----------------");
// Set the name to ROBOT_NAME
Serial.print("Set the name: ");
String rnc = String("AT+NAME") + String(ROBOT_NAME);
mySerial.print(rnc);
waitResponse();

//Set baudrate to 9600
//AT+BAUD1  OK1200  Sets the baud rate to 1200
//AT+BAUD2  OK2400  Sets the baud rate to 2400
//AT+BAUD3  OK4800  Sets the baud rate to 4800
//AT+BAUD4  OK9600  Sets the baud rate to 9600
```

```
  //AT+BAUD5  OK19200 Sets the baud rate to 19200
  //AT+BAUD6  OK38400 Sets the baud rate to 38400
  //AT+BAUD7  OK57600 Sets the baud rate to 57600
  //AT+BAUD8  OK115200  Sets the baud rate to 115200
  //AT+BAUD9  OK230400  Sets the baud rate to 230400
  //AT+BAUDA  OK460800  Sets the baud rate to 460800
  //AT+BAUDB  OK921600  Sets the baud rate to 921600
  //AT+BAUDC  OK1382400 Sets the baud rate to 1382400

  Serial.println("----------------");
  // Set baud rate to 9600
  Serial.print("Set baud rate: ");
  mySerial.print("AT+BAUD4");
  waitResponse();

  Serial.println("The configurations are done!");
}


void loop() {

  }
```

```
void waitResponse() {
  delay(2000);
  while (mySerial.available()) {
    Serial.write(mySerial.read());
  }
  Serial.write("\n");
}
```

The Android Application and the Arduino Sketch



After the application is installed on your device, you have to scan for devices, enter the PIN number set with the script above, and connect the Bluetooth module.
The interface layout provides 10 buttons specifically designed to send continuously commands while pressed. For now, we only use two of the buttons: one button to send “1”, and another to send “2”. So, use the settings of the application to set the value “1” and value “2” for two of the buttons. These values will be received by the Bluetooth module and used in the Arduino sketch to control the servo motor.

After the Android application setup is finished, we have to turn back to the Arduino and upload the code to control the servo motor. Below is the Arduino sketch to turn the servo motor at a specific position.

```
/**
 * @file       How To Control a Servo Motor With a Bluetooth Module,
Arduino and Android
 * @author     Calin Dragos for intorobotics.com
 * @version    V1.0
 * @date       13.12.2016
 * @description  This is an Arduino sketch to control the servo motor
SG90 with the Bluetooth module HC-06 and the Arduino Bluetooth
Controller application
 */


#include <SoftwareSerial.h>
#include <Servo.h>


SoftwareSerial mySerial(10, 11); // RX | TX


Servo servo;
```

```
int servoPin = 9;
int servoAngle = 0;   // servo position in degrees

char command;

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  Serial.println("You're connected via Bluetooth");
  servo.attach(servoPin);
 }


void loop() {
 if (mySerial.available())
  {
   command=(mySerial.read());
   if (command=='1')
   {
     Serial.println("Servo motor to 10 degrees");
     servo.write(10);
     delay(500);

   }
```

```
    else if (command=='2')
    {
     Serial.println("Servo motor to 120 degrees");
     servo.write(120);
     delay(500);
    }

  }

}
```

Chapter 7
Conclusion and Future scope

In this paper we presented our fly and crawl quadcopter that can travel on the ground and fly using the same motors .The robot is reconfigurable and is fitted with both wheels and propellers .

We can experiment the robot by flying it and running over ground . The robot in flying mode lands in and re-orients itself to enter underneath an obstacle and exit it.

Finally we tested the robot outdoors on a variety of surfaces . The robot can crawl over grass and ground .

Nowadays military has been using advanced technology to get at par level ahead of our enemies in which more deliberately biomimics is used. Since at some remote locations the army cannot go so in terms of guerrilla warfare it can use such robots to get to know about counterparts activities as soon as possible . Our quadcopter not only can spy through by hovering over enemies head but can also can crawl over land and reach much closer.

Chapter 8

References

1. Nir Meiri, and David Zarrouk, "Flying STAR, a Hybrid Crawling and Flying Sprawl Tuned Robot" 2019 International conference on Robotics and Automation (ICRA) palais de congres de Montreal, Montreal , Canada, May 20-24 ,2019
2. Shaikh Altamash, Padwekar Amir , Syed Adnan, "Kinematic and Dynamic Modelling and Simulation of Quadcopter", AIKTC ,New Panvel
3. Xiaobin Xiong, Aaron d Ames, Hyung ju suh, Andrew Singletary "Optimal Motion Planning for Multi-Modal Hybrid Locomotion".
4. " Quadcopter Control Using Arduino Microcontroller", RCC Institute of Technology Kolkata.

Websites

1. YouTube channel :-"painless 360" - https://youtube.com/playlist?list=PLYsWjANuAm4pnhS9StKcIbwSOdoMaR9-p
2. https://www.intorobotics.com/how-to-control-a-servo-motor-

with-a-bluetooth-module-arduino-and-android/