

TRIBHUWAN UNIVERSITY INSTITUTE OF ENGINEERING PULCHOWK CAMPUS

**PANCHHI**

BY

**BIMAL KHADKA**

**BIPLAB KARKI**

**JAY KISHAN PANJIYAR**

**NAVIN RAI**

A PROJECT REPORT WAS SUBMITTED TO THE DEPARMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN THE PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE BACHELOR’S DEGREE IN ELECTRONICS AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL AUGUST 2019

# PAGE OF APPROVAL

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING PULCHOWK CAMPUS

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled "Panchhi" submitted by Bimal Khadka, Biplab Karki, Jay kishan Panjiyar and Nabin Rai in partial fulfilment of the requirements for the Bachelor’s degree in Electronics & Communication / Computer Engineering.

Supervisor, Dr. Ram Krishna Maharjan Professor

Electronics and computer engineering

Internal Examiner, Ram Krishna Maharjan

Professor Pulchowk Campus

Internal Examiner, Dr. Diwakar Raj Pant

Reader

Pulchowk Campus

External Examiner, Suman Sharma

Lecturer Pulchowk Campus

**DATE OF APPROVAL: AUGUST.2019**

# COPYRIGHT

The author has agreed that the Library, Department of electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering may take this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering in any use of the material of this project report. Copying or publication or the other use of this report for financial gain without approval of the Department of Electronics and Computer Engineering, Pulchowk campus, Institute of Engineering and author’s written permission is prohibited.

Request for permission to copy or to make any other use of the materialin this report in whole or in part should be addressed to:

Head

Department of Electronics and Computer Engineering Pulchowk Campus, Institute of Engineering Pulchowk, Lalitpur

Nepal

# ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us during this project. We owe special thanks to a number of people who have devoted much of their time and expertise without which it would have been very difficult for us to complete our project entitled “Panchhi”.

We would like to take the opportunity to express our humble gratitude to our supervisor Dr. Ram Krihana Maharjan and Dr. Diwakar Raj Pant, under whom we undertook our project. Their constant guidance and willingness to share vast knowledge made us enhance our knowledge and helped us to complete the tasks.

Moreover, we would like to extend our special thanks to entire Electronics and Computer Engineering Department of IOE, Pulchowk Campus for providing technical support with suggestions which helped our project to foster. We feel obliged in submitting this project as part of our curriculum.

Although, there may be still many who are unacknowledged in this humble vote of thanks, there are none who remain unappreciated.

# ABSTRACT

Panchhi is an IoT controlled autonomous drone with wireless charging feature. The drone is aimed to be a multi-purpose drone applicable in industries like Traffic Monitoring, Agriculture, Military Defense, and Aerial Marketing and so on. The drone can be remotely controlled from any location via any forms of internet connectivity. We may control it manually as well as let it fly independently as we send GPS coordinates to it via our web application. The drone can be used in various tasks as per the need of the society. From social to business aspects, an aerial vehicle like drone comes as a boon.

# TABLE OF CONTENTS

[PAGE OF APPROVAL ii](#_bookmark0)

[COPYRIGHT iii](#_bookmark1)

[ACKNOWLEDGEMENT iv](#_bookmark2)

[ABSTRACT v](#_bookmark3)

[TABLE OF CONTENTS vi](#_bookmark4)

[LIST OF FIGURES viii](#_bookmark5)

[LIST OF TABLES ix](#_bookmark6)

1. [INTRODUCTION 1](#_bookmark7)
   1. [Problem Statement 1](#_bookmark8)
   2. [Project objectives 1](#_bookmark9)
   3. [Significance of Study 1](#_bookmark10)
   4. [Scope and Limitation 1](#_bookmark11)
   5. [Cost Estimate 2](#_bookmark12)
2. [LITERATURE REVIEW 3](#_bookmark13)
   1. [Brushless DC motor 3](#_bookmark14)
   2. [Electronic Speed Controller(ESC) 4](#_bookmark15)
   3. [IoT 5](#_bookmark16)
      1. [MQTT 5](#_bookmark17)
      2. [SSL 5](#_bookmark18)
      3. [TLS 5](#_bookmark19)

[2.3.4. X.509 5](#_bookmark20)

* 1. [Cloud Computing 5](#_bookmark21)

1. [PROCEDURE 7](#_bookmark22)
   1. [QUADCOPTER DESIGN 7](#_bookmark23)
   2. [Motors order 8](#_bookmark24)
   3. [Recognizing clockwise and counterclockwise propellers 9](#_bookmark25)
   4. [Testing motor spin directions 9](#_bookmark26)
   5. [Checking the motor numbering with the Mission Planner Motor test 10](#_bookmark27)
2. [METHODOLOGY 12](#_bookmark28)
   1. [TOOLS AND TECHNOLOGIES USED 12](#_bookmark29)
3. [Coding and implementation 14](#_bookmark30)
   1. [Communicating with Raspberry Pi via MAVLink 14](#_bookmark31)
      1. [Connecting the Pixhawk and Rpi 14](#_bookmark32)
      2. [Setup the RPi 15](#_bookmark33)
      3. [SETTING UP PIXHAWX 15](#_bookmark34)
      4. [Install the required packages on the Raspberry Pi 15](#_bookmark35)
      5. [Disable the OS control of the serial port 15](#_bookmark36)
      6. [Testing the connection 16](#_bookmark37)
4. [RESULTS 18](#_bookmark38)
5. [CONCLUSION AND FURTHER WORK 19](#_bookmark39)

[REFERENCES 20](#_bookmark40)

# LIST OF FIGURES

Figure 1: BLDC motor 3

Figure 2: electronic speed controller 4

Figure 3: Illustration of esc with bldc 4

Figure 4: Quadcopter 7

Figure 5: Direction of rotation 8

Figure 6: Quad X 8

Figure 7: Propeller direction 9

Figure 8: Mission planner: Motor test 11

Figure 9: Connection of Pi with Pixhawk 14

Figure 10:RASPICONFIG:ADVANCEDOPTIONS 16

Figure 11: COMMAND SCREEN 17

# LIST OF TABLES

TABLE 1: HARDWARE COST TABLE 2

TABLE 2: TECHNOLOGY USED 11

# INTRODUCTION

Panchhi is quadcopter that means it is driven using four motors. It is manually controlled by using IoT. A website is designed through which all the commands to run the quadcopter is sent. The website is hosted in Amazon Web Services server. It is also autonomous in the sense that after giving the GPS location it will navigate itself from the destination location to the desired location without the interference of user.

* 1. Problem Statement

The major problems of present time drones are numerous in real life scenario. The major problem that arises in it are listed below: -

1. Range of drones are limited
2. Faster battery discharge scenario
3. Only manual control of drone availability
4. Remote monitoring seems expensive and complicated

## Project objectives

Following are the objectives of this project:

* + 1. Making a quadcopter
    2. Replacing manual control using RF transmitter with IoT control
    3. Autonomous navigation of quadcopter

## Significance of Study

The study investigates the need of the system that helps the people relating to the aerial industry who aims to make changes to the society using the power of drones. Various other fields like agriculture, health, industry, business can even use our product.

## Scope and Limitation

The main scope of this drone is that it can lift up to the weight of 2kg and having camera attached to it we can use it as a remote surveillance.

Following are the limitation of this drone:

1. Due to weak connectivity of WiFi slow data transfer through IoT
2. Fast discharging of battery since it take huge amount of current to run the BLDC motors.
3. Less efficient wireless power transfer

## Cost Estimate

TABLE 3: HARDWARE COST TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Device | Quantity | Cost (Approx) |
| 1. | Raspberry pi | 1 | 5600 |
| 2. | Lipo battery | 1 | 2000 |
| 3. | ESC | 4 | 3800 |
| 4. | Brushless motor | 4 | 4000 |
| 5. | Frame | 1 | 2000 |
| 6. | Propeller | 4 | 800 |
| 7. | PCB components |  | 1000 |
| 8. | Jumper Wires & Connectors |  | 500 |
| 10. | Flight Controller | 1 | 9000 |
| 11. | GPS Module | 1 | 2800 |
|  | Total |  | 31,500 |

Software cost:

AWS EC2 [Linux Machine] (for Web Server): - NRP 1,500/ per month (730 hours, 1 CPUs, 4 GB Memory)

AWS EC2 [Linux Machine] (for Web Server): - NRP 2,000/ per month (730 hours, 2 CPUs, 4 GB Memory)

AWS S3 [Simple Storage Service]: - NRP 300/ per month (100 GB, 10 Requests, 10 GB per request)

AWS IoT Core [IoT Management]: - NRP 200/ per month (10,000+ messages) AWS Monitoring Services: - NRP 300/ per month Total: - 4,300/ per month

# LITERATURE REVIEW

* 1. Brushless DC motor

A brushless DC motor consists of a rotor in form of a permanent magnet and stator in form of poly-phase armature windings. It differs from conventional dc motor in such that it doesn’t contains brushes and the commutation is done using electrically, using a electronic drive to feed the stator windings.

As their name implies, brushless DC motors do not use brushes. With brushed motors, the brushes deliver current through the commutator into the coils on the rotor. So how does a brushless motor pass current to the rotor coils? It doesn’t—because the coils are not located on the rotor. Instead, the rotor is a permanent magnet; the coils do not rotate, but are instead fixed in place on the stator. Because the coils do not move, there is no need for brushes and a commutator.

With the brushed motor, rotation is achieved by controlling the magnetic fields generated by the coils on the rotor, while the magnetic field generated by the stationary magnets remains fixed. To change the rotation speed, you change the voltage for the coils. With a BLDC motor, it is the permanent magnet that rotates; rotation is achieved by changing the direction of the magnetic fields generated by the surrounding stationary coils. To control the rotation, you adjust the magnitude and direction of the current into these coils.



**FIGURE 1: BLDC MOTOR**

## Electronic Speed Controller(ESC)

An electronic speed control follows a speed reference signal (derived from a throttle lever, joystick, or other manual input) and varies the switching rate of a network of field effect transistors (FETs). By adjusting the duty cycle or switching frequency of the transistors, the speed of the motor is changed. The rapid switching of the transistors is what causes the motor itself to emit its characteristic high-pitched whine, especially noticeable at lower speeds.

FIGURE 2: ELECTRONIC SPEED CONTROLLER

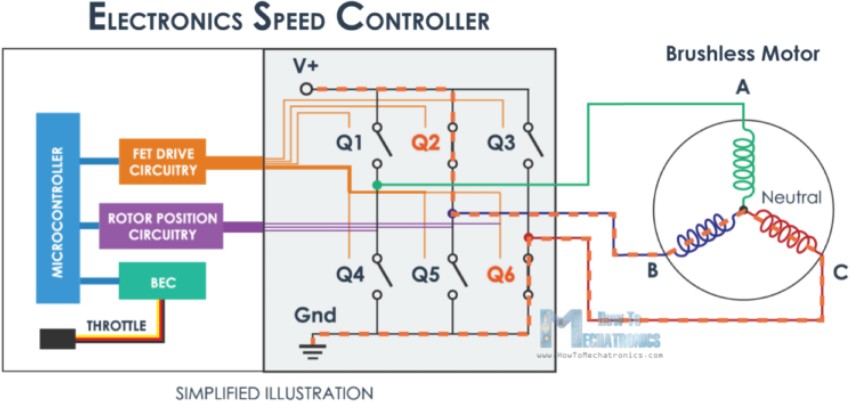


FIGURE 3: ILLUSTRATION OF ESC WITH BLDC

## IoT

IoT is an abbreviation to Internet of Things, which simply means the concept of connecting any non-living smart devices or some electronics devices with each other or to the internet so that they can show some level of smartness. Connected devices and sensors can provide some data which can help to trigger some solutions that makes daily life easier. IoT comes up with various forms like M2M (Machine To Machine) communication, V2V (Vehicle To Vehicle) communication, IIoT (Industrial IoT). IoT nowadays come up as a subset of IoE (Internet of Everything). An example of IoT can be a smart switch at India controlled by a person at US or a motion sensor turning off camera if it detects no motion for 5 minutes.

## MQTT

As of the availability of various IoT protocols in the industry, we have come up with the most used and lightweight protocol for the task of IoT communication. MQTT stands for Messaging Query Telemetry Transport, for connecting our drone to the cloud so that each and every data from the brain of drone, i.e. Raspberry Pi is monitored via cloud. Similarly, using web socket over MQTT in the web application, the application is connected to the drone through the path of cloud using MQTT. To make the website secured via browser, we have implemented SSL (Secured Socket Layer) protocol while for the IoT based security, the protocol used for the case of IoT security is TLS (Transport Layer Security) 1.2.

## SSL

SSL (Secured Socket Layer) is a standard security technology for establishing an encrypted link between a web server and a browser. Using public and private key cryptography algorithms, SSL is implemented for the sake of encrypting each data that comes from frontend to the backend of web application and vice-versa. SSL is known as the older version of **TLS**.

## TLS

TLS (Transport Layer Security) is the modern form of SSL, which uses private and public key cryptography algorithms along with validation of the public key too using another certificate named **X.509**.

2.3.4. **X.509**

X.509 is a standard defining the format of public key certificates. It is a certificate used along with private and public certificate to validate the standard of public key of certificate.

## Cloud Computing

Cloud computing is the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. All data storage, processing and retrieval take place in those servers known as

cloud. Hence, there is no burden to the local machine. In this way, we will be using cloud computing for our entire web, database hosting as well as other data processing tasks.

1. **PROCEDURE**

## QUADCOPTER DESIGN

Firstly, the frame was assembled which has four arms. Brushless DC motors were attached to the end of each arm. Each BLDC motors were then connected with the Electronic Speed Controller (ESC) and the escs were then connected to the common power supply. Polarity of the battery and escs were checked correctly before connecting them. Then power supply was given to test all the motors were receiving power from battery. Then flight controller was mounted at the middle of frame. The signal wires from all four escs were then connected to the flight controller. Then with the help of mission planner all the escs were calibrated and motors were tested whether they are working properly or not. After testing all motors and escs were found to



FIGURE 4: QUADCOPTER

be working properly. Then above the flight controller, Raspberry pi was mounted which was the main controller of the quadcopter. The Raspberry Pi and flight controller were 3.cable for

communication between them. Then GPS module was connected at the front of the quadcopter. Finally propellers were fitted on each motors.

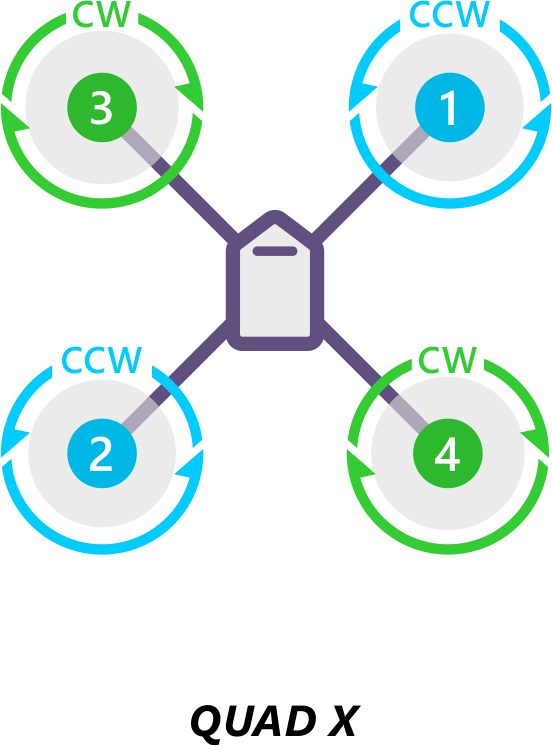
## Motors order

The diagrams below show motor order for each frame type. The numbers indicate which output pin from the flight controller should be connected to each motor/propeller. The propeller direction is shown in green (clockwise, CW) or blue (counter-clockwise, CCW).

FIGURE 5: DIRECTION OF ROTATION



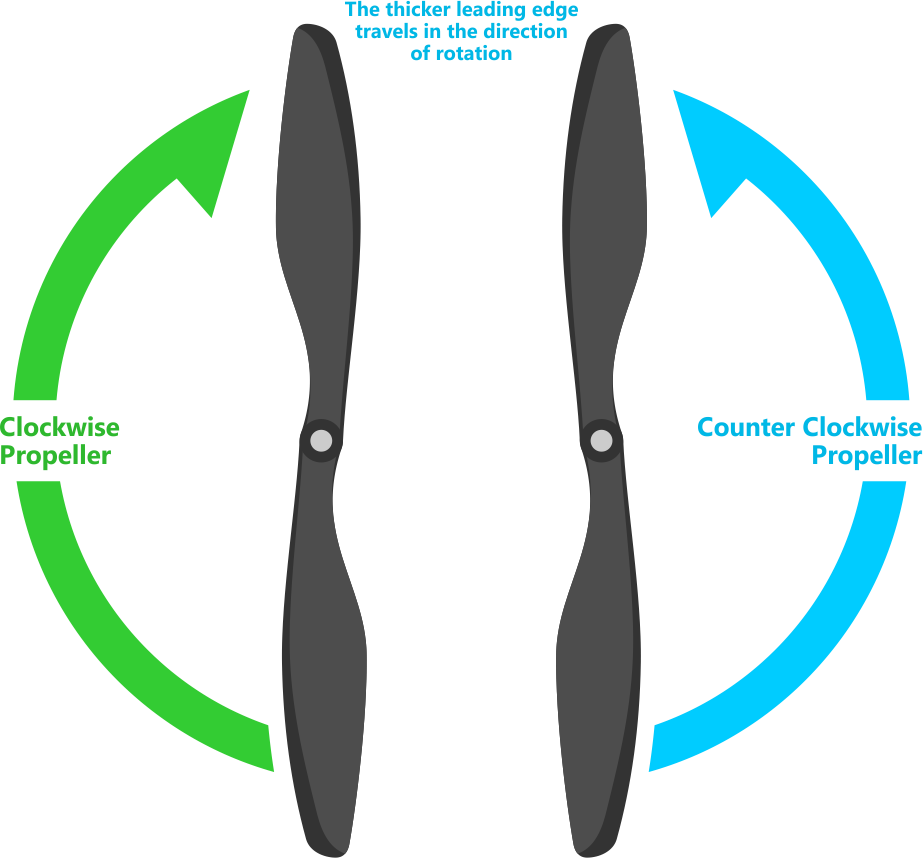
FIGURE 6: QUAD X



## Recognizing clockwise and counterclockwise propellers

The diagrams above show two types of propellers: clockwise (called pushers) and counterclockwise (called pullers). The most reliable to recognize the correct propeller type by its shape as shown below. The thicker edge is the leading edge which moves in the direction of rotation. The trailing edge is more radical scalloped and usually thinner.

FIGURE 7: PROPELLER DIRECTION



## Testing motor spin directions

Once ESC calibration was completed, motors directions were checked whether they were spinning in the correction direction.

Steps:

1. Make sure there are no propellers on your copter!
2. Connect battery.
3. Arm copter
4. If it fails to Arm the motors will not spin, it has probably failed the Pre-Arm Safety Check.
   * Pre-Arm safety check failure is also indicated by the red arming light double flashing and then repeating.
   * If the Pre-Arm check fails go to the Prearm Safety Check Page and correct the problem or disable the check before continuing.
5. Reverse any motor spinning in the wrong direction.

## Checking the motor numbering with the Mission Planner Motor test

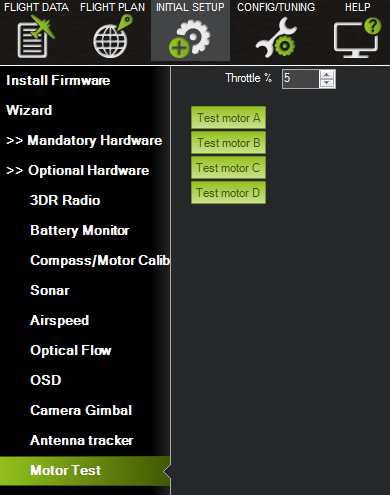
An alternative way to check that the motors have been hooked up correctly is to use the “Motors” test in the Mission Planner Initial Setup menu.

When connected to the vehicle via MAVLink, you can click on the green buttons shown above and the corresponding motor should spin for five seconds. Letters correspond to motor numbers as shown in the example below.

* Take off your props first!
* If no motors turn, raise the “Throttle %” to 10% and try again. If that doesn’t work, try 15%

The first motor to spin will be the one located directly forward in the case of + configuration, or the first motor to the right of straight forward in the case of X configuration. The motor test will then proceed in a clockwise rotation.

FIGURE 8: MISSION PLANNER: MOTOR TEST



# METHODOLOGY

For this project, we have planned to work following the methodologies below for the application of ideas, knowledge, skills and tools to a wide range of processes in order meet the requirements of our project and take it to successful completion.

## TOOLS AND TECHNOLOGIES USED

The development of Panchhi was possible by use of several tools provided in the table below.

TABLE 4: TECHNOLOGY USED

|  |  |  |
| --- | --- | --- |
| S.N. | Name | Purpose/Description |
| 1. | Raspberry Pi | A small computer board with high level of computing capability. We use it for IoT  based cloud as well as Mav Proxy based telemetry connection with flight controller |
| 2. | Pix hawk | Pixhawk is an independent open-hardware project that aims to provide the standard for readily-available, hiqh-quality and low-cost autopilot hardware designs for the academic, hobby and developer communities.Pixhawk is an independent open-hardware project that aims to provide the standard for readily-available, hiqh- quality and low-cost autopilot hardware  designs for the academic, hobby and developer communities. |
| 3. | AWS (Amazon Web Services) | We used one of the most popular cloud platform AWS for our web and IoT platform since it provides highly reliable and secured platform for the software deployment as well as connecting devices  to the cloud. |
| 4. | Node.js | An open source, server-side JavaScript run-time environment which was used to  hold the entire backend system of our website |
| 5. | Python | We used python to control entire raspberry  pi featuring connection from Pi to cloud and Pi to flight controller |

|  |  |  |
| --- | --- | --- |
| 6. | HTML,  CSS,Javascript/Jquery | These scripting and markup languages were  used to provide overall design and functionality to the website |

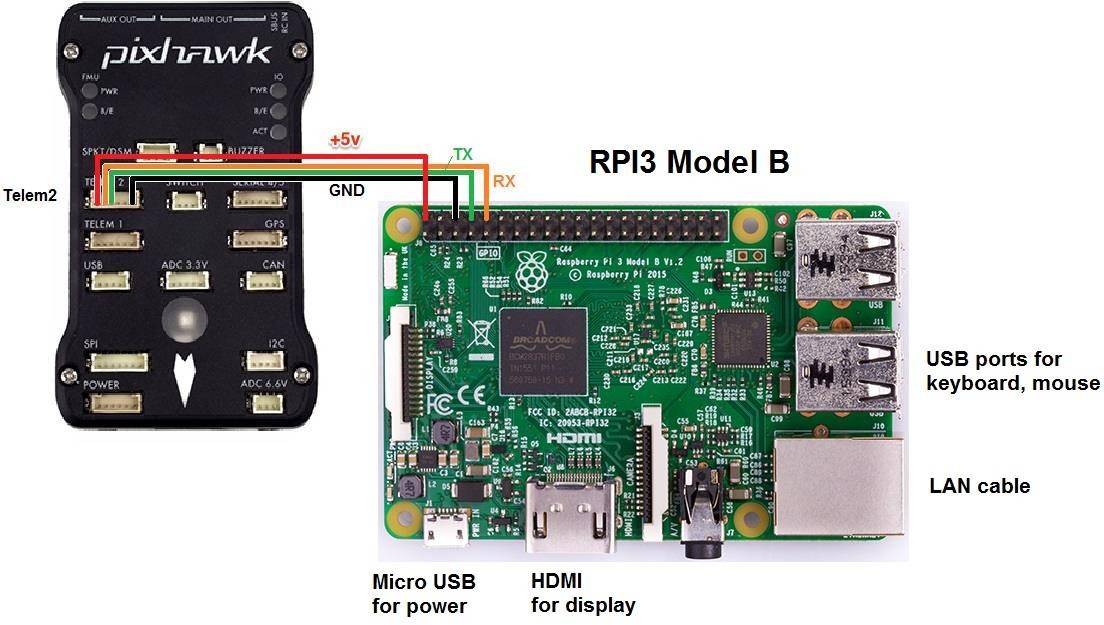
# Coding and implementation

After finding out the logical solution for the requirement enlisted, the end product was assembled accordingly. The steps followed for the coding and implementation of the product to fulfill the requirements are:

## Communicating with Raspberry Pi via MAVLink

Connecting and configuring a Raspberry Pi (RPi) so that it is able to communicate with a Pixhawk flight controller using the MAVLink protocol over a serial connection. This can be used to perform additional tasks such as image recognition which simply cannot be done by the Pixhawk due to the memory requirements for storing images:

## Connecting the Pixhawk and Rpi



**FIGURE 9: CONNECTION OF PI WITH PIXHAWK**

Connect the Pixhawk’s TELEM2 port to the RPi’s Ground, TX and RX pins as shown in the image above.

The RPi can be powered by connecting the red V+ cable to the +5V pin (as shown above) or from USB in (for example, using a separate 5V BEC hooked up to the USB power)

## Setup the RPi

The easiest way to setup the RPi is to flash one of the existing APSync images:

* Purchasing a formatted 8GB or 16GB SD card (16GB is better because some 8GB cards will not be quite large enough to fit the image) and insert into your laptop/desktop computer’s SD card slot
* Downloading the latest image from firmware.ardupilot.org. Look for the file starting with “apsync-rpi”.
* Extracting the image

## SETTING UP PIXHAWX

Connecting the Pixhawk with a ground station (i.e. Mission Planner) and set the following parameters:

* SERIAL2\_PROTOCOL = 1 (the default) to enable MAVLink on the serial port.
* SERIAL2\_BAUD = 921 so the Pixhawk can communicate with the RPi at 921600 baud.
* LOG\_BACKEND\_TYPE = 3 if you are using APSync to stream the dataflash log files to the RPi

## Install the required packages on the Raspberry Pi

After the internet connection is confirmed to be working install these packages

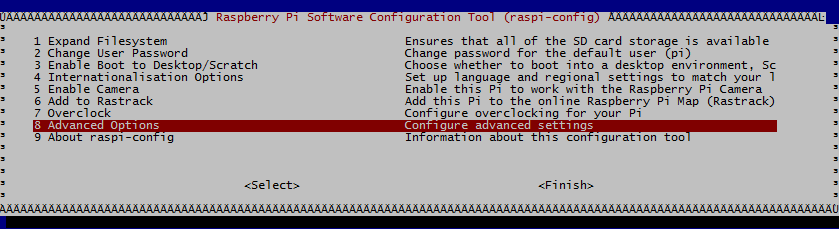
* sudo apt-get update #Update the list of packages in the software center
* sudo apt-get install screen python-wxgtk2.8 python-matplotlib python-opencv python- pip python-numpy python-dev libxml2-dev libxslt-dev python-lxml
* sudo pip install future
* sudo pip install pymavlink
* sudo pip install mavproxy

## Disable the OS control of the serial port

Use the Raspberry Pi configuration utility for this: Type:

* + - * sudo raspi-config

And in the utility, select “Advanced Options”:



**FIGURE 10:RASPICONFIG:ADVANCEDOPTIONS**

And then “Serial” to disable OS use of the serial connection

## Testing the connection

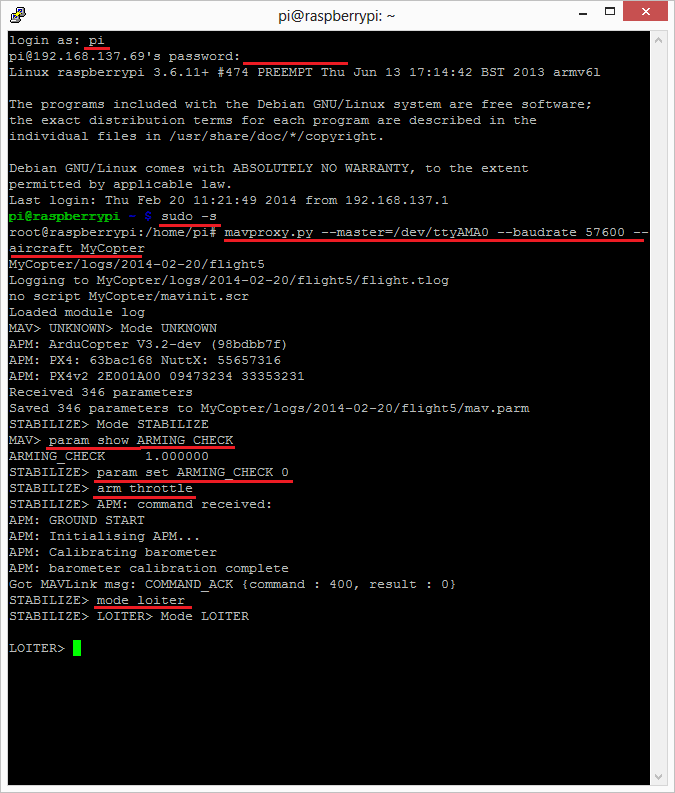
To test the RPi and Pixhawk are able to communicate with each other first ensure the RPi and Pixhawk are powered, then in a console on the RPi type:

* + - * sudo -s
      * mavproxy.py --master=/dev/ttyACM0 --baudrate 921600 --aircraft MyCopter

ACMO for USB port access

Once MAVProxy has started you should be able to type in the following command to display the ARMING\_CHECK parameters value.

* + - * param show ARMING\_CHECK
      * param set ARMING\_CHECK 0
      * arm throttle



**FIGURE 11: COMMAND SCREEN**

# RESULTS

After the assembly of all the components, the quadcopter weight was found to be nearly 800gm.

After the setting of everything as per instructed in the procedure using the technologies as per instructed in the methodology section, quadcopter was completed. Firstly, quadcopter was operated in the manual mode in which it was operated with the help of IoT through the laptop and quadcopter was flown near about the height of 50m.

After the manual test was succeed then the quadcopter was operated in the automatic mode. In this mode only the GPS coordinates were given and then quadcopter was powered on and when coordinates were received , quadcopter was flown using the dornekit library and reached the location and land was successful.

From the test, it was found that Panchhi drone has a flight time of 45 minutes when the battery of 3000mah was used.

# CONCLUSION AND FURTHER WORK

Quadcopter was completed and successfully tested. Using the IoT and cloud with the help of dronekit library, it was operated manually as well as autonomously. It is concluded that quadcopter can also be operated using IoT and it can fly even without the intervention of human to reach the given location as long as battery power hold.

Following are the further work we would like to do in future to make our quadcopter to be more efficient and effective and commercialization of it.

* Installment of wireless charging so that quadcopter can charge itself in different wireless charging stations and hence we can increase its flight time.
* Autonomous features where it will learn itself to fly in the free environment without any human interference.
* Increasing its capacity to carry payloads upto 2kg.
* Use of camera of high quality for surveillance and monitoring of agricultural farm

# REFERENCES

* <http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html>
* <http://ardupilot.org/planner/docs/common-connect-mission-planner-autopilot.html>
* <http://ardupilot.org/copter/index.html>
* [https://www.instructables.com/id/How-to-Make-a-Drone-at-Home-DIY- Quadcopter](https://www.instructables.com/id/How-to-Make-a-Drone-at-Home-DIY-Quadcopter)
* <https://aws.amazon.com/iot>
* <https://aws.amazon.com/ec2>
* <http://ardupilot.org/planner/docs/common-loading-firmware-onto-pixhawk.html>
* <http://ardupilot.org/dev/docs/mavlink-commands.html>
* [http://ardupilot.org](http://ardupilot.org/)