DOI: https://doi.org/10.33103/uot.ijccce.20.1.1

Design and Implementation of Autonomous Quadcopter using SITL Simulator

Hasan M. Qays[1], Dr. Bassim A. Jumaa[2], Dr. Ayman D. Salman[3]
Computer Engineering Department, University of Technology, Baghdad, Iraq
[1]120568@student.uotechnology.edu.iq, [2]basim_alani@yahoo.com, [3]120008@uotechnology.edu.iq.

Abstract— *In recent years Quadcopter has been used in many applications such as military, security & surveillance, service delivery, disaster rescue and much more due to its flexibility of flying. In this paper, Quadcopter will be used for mail delivery between many locations that is received from the end user. The Quadcopter will execute an autonomous flight using the concept of companion PC. Raspberry PI 3 (RPI3) will control the Quadcopter by command the controller of the drone (Pixhawk) by using DroneKit-Python API to send MAVLink messages to the Ardupilot. This concept is useful to perform an additional task to the autopilot and provide such a smart capability like image processing and path planning which cannot be done by the flight controller alone. Basically, the idea has been stimulated and the code has been tested by using the SITL Simulator with MAVProxy under Ubuntu environment. The result of controlling the Quadcopter using Python script was excellent and give a motivation to implement the same script on a real Quadcopter. The implementation on real Quadcopter was perfect as it has given the same behavior as the SITL drone in the simulation.*

Index Terms— *Quadcopter, Companion PC, ArduPilot, DroneKit-Python, MAVLink, Autonomous flight, Pixhawk.*

I. INTRODUCTION

The Unmanned Arial Vehicle (UAV) or drone is a vehicle without a pilot on it. There is a system for the UAV called the Unmanned Aerial System (UAS) that allows communication with the physical drone [1].

Drones are often controlled by a human pilot which is a user input by the way of using the remote control that is known as Radio Controller (RC). But also, they can be autonomously controlled by the system integrated on the drones themselves without using the RC input. The UAS is installed on the companion PC which is called onboard computer [2],[3].

The drone was initially developed for a military purpose, had its test flight before the World War II, and then was fully deployed in the Iraq War and the Afghan War from around 2000. [4].

A reliable UAV appeared in the 1990s, and its research and development became active in universities and other academic fields. In Georgia Institute of Technology, the first international flying robot competition was held in 1991 [5]. This was started as a university event but gradually expanded worldwide into the field of small unmanned aerial vehicle among government, industry and academia. Most of the studies focused on issues of higher performance hardware platforms, software systems, aerodynamic models, and autonomous flight control, and the state-of-the-art technologies were competed. In addition, (Defense Advanced Research Project Agency) (DARPA) set up in 1997 a project of MAV (Micro Air Vehicle), which led to the development of small drone with the diameter not exceeding 15 cm at an annual research funding of 35 million USD [4].

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

In 2009, 3D Robotics and the Swiss Federal Institute of Technology in Zurich introduced the open-source Autopilot system that is named (ArduPilot Mega) (APM) and its newer versions named Pixhawk [4]. Many versions were released such as: ArduPilot in 2009, APM1 in 2010, APM2 in 2011, APM 2.5 in 2012 and Pixhawk in 2013.

Depending on the study of the Federal Aviation Administration (FAV) regarding the drone's sales, the marketing of this industry is expected to increase from 2.5 million in 2016 to 7 million in 2020. This increase in sales is due to the new capabilities and the various types of sensors that became supporting with the decrease in their price. The new production of drones support various and important types of sensors such as barometer, air speed, air pollution, Light Detection and Ranging (LiDAR), Ultra-sonic sensors, etc., with support of wireless communication for example Wi-Fi, 4G and so on. [6].

These improvements have made the drones very useful tools to use in different Internet of Things (IoT) applications for examples: collect air quality data from places that are considered hard to reach or dangerous for human and send these data to the server on real time.[6]. Some companies are using drones specifically because of their high flexibilities for example Amazon and DHL which they tested drones in service delivery.[7].

There are three major kinds of drones: fixed wings, single motor and multirotor. The multirotor is the most common among them, they are classified depending on the number of motors they have such as tri-copter (three motors), Quadcopter (QC, four motors), hexacopter (six motors) and octocopter (eight motors). The most common multirotor is quadcopter [8].

The traditional fixed-wing UAVs can fly for long distance but require runways or wide-open spaces for taking -off and landing. On the other hand, the more trending multirotor UAVs are extremely maneuverable but cannot be used for long-distance flights because of their slower speeds and relatively higher consumption of energy. The flight is more stable compared with a single rotor and the fixed wing in terms of taking off, landing and controlling the maneuvering of the multi-rotor [9].

Several studies have been devoted and deployed with various issues towards design and implementation of using quadcopter for many applications such as: delivery service, monitoring, video transfer, military, reconnaissance mission, rescue and product conveyance for instance, deliver medical provides to inaccessible places, film creation and far a lot of, for expamles:

M. Muhammad. et al., 2014 [2], introduced autonomous quadcopter for producing home delivery using android device as on-board processing unit and connecting to APM flight controller. The curren paper used walking mode direction on google map for determining the path. In this paper the path provided by walking mode is not the shortest path since it depends on ground path as it became longer than the straight line, also no multiple destinations are considered. The author uses only SMS notifications to notify the users, no emails notifications are used which provide more detail information to the users. While RPI is used as on-board processing uint which can be used for running many applications such as image processing , object tracking and path planning. These applications need higher processing and libraries that can not be installed on mobile device only.

Alberto Lisanti and Giorgio Venezia 2015 [3], provided a system to quadcopter for drug shipments. They introduce a very useful android application to be used by the client and pharmacies to request such a medicine. In this paper there is only one request per time which means only one destination for the quadcopter and only one path then return to the home location, also no-load calculations are considered. The quadcopter is not fully commanded by the companion PC, as its only taking user's location and then flying to that location by using AUTO mode. While the current work considered multiple destination points receviced from registers users in the system, load calculation, and fully autonoums flight using GUIDED mode.

Cameron Roberts [2016] [10], the author designed quadcopter to be guided autonomously using GPS module. Navio flight controller was used to be connected with RPI. The implementation

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

only receives the waypoints from the user using direct connection with the RPI through Secure Shell (SSH) protocol to access the terminal of the RPI. No path planning is taking into consideration. No user remote access to the system and no graphic user interface (GUI) to let users send their requests. Also, power estimation and path planning are not considered by the author.

Majida S. et al., [2018] [11], introduced a monitoring system using quadcopter with an attached camera to the Raspberry PI to evaluate the video transfer using different protocols. This study introduced an interesting evaluation for video transfer using Ad-hoc network depending on quadcopter. The draw back in this study is that they used quadcopter with manual control using radio transmitter at the required location to capture the video and with other person at receiver side synchronized with the operator to control the movement of the quadcopter. This is not a practical solution for moving and controlling the quadcopter and there is no user interaction with the system; also no path planning is used, moreover no notification system is used.

Taking into consideration all the above works which have used the drones, they all use the autonomous flight but the implementation, hardware, and software are different according to the application requirements. In this paper, the QC is used to implement the autonomous flight using the concept of onboard computer which is, in this case, the Raspberry PI 3 B+ (RPI3) to delivers the mail among the departments of the University of Technology-Iraq.

The RPI3 is used to control the quadcopter by commanding the controller of the drone (Pixhawk) using Drone-Kit API to send a MAVLink message (Micro Air Vehicle Link protocol). This concept is very useful to perform additional smart tasks to the autopilot software for example image processing and path planning which cannot be done by the flight controller alone.

Actual testing using real drones are considered costly. Therefore, the drone's systems must be tested and checked prior to deploying it. In order to avoid the damage on costly devices and sensors and to develop reliable system, many researchers use some simulation environments. The Software In The Loop (SITL) is one of these simulators which gives the ability to run various vehicles such as Plane, Copter and Rover, without a need to any microcontroller or hardware. [12].

Since the Ardupilot is a cross-platform, it can be used on different operating systems. The SITL has this feature to run on many environments such as Linux, Windows or MAC OS . The autopilot that is used in this paper is the open source code software named ArduPilot. ArduPilot is a free software package and it contains all the software needed to monitor the drone, reading sensors inputs and control the drone according to the data received from the sensors. This software is available for programmers and researchers. It's been developed over 5+ years by a team of various skilled engineers and computer scientists. It is the only autopilot computer code capable of controlling any vehicle system possible, from standard airplanes, multirotor, and helicopters, to boats and submarines [13].

The result of controlling the quadcopter in tearm of takeoff, fly to specific location, and landing was done successfully and give us the motivation to implement the same algorithm on a real drone. The implementation on real quadcopter was done successfully and it gives the same behavior as the SITL drone in the simulation since it uses the same autopilot software which is Ardupilot.

II. PRINCIPLES OF QUADCOPTER OPERATION & PROTOTYPE COMPONENTS

Quadcopter consists of four motors connected to the frame body at equal distances from each another. Two of motors rotated in clockwise, the other two motors rotated in counterclockwise in order to provide the specified thrust to lift the Quadcopter in the air as described in Fig.1.

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

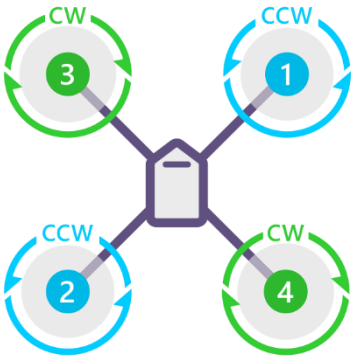


FIG. 1. QUADCOPTER MOTOR DIRECTION.

Table1 below shows the most components which are used in the design of the prototype of this study with its weight. The weight is too important in the next sections as it is needed to calculate the thrust, power, and flight time

TABLE 1. QUADCOPTER COMPONENTS WEIGHT

Parameter	specifications
Frame	330 grams
Motors X4	212 grams
Flight controller	40 grams
Electronic Speed Controller ESC X4	128 grams
Ublox GPS Neo-M8N module	15 grams
LiPo battery Blackmagic 5200 mAh	400 grams
Raspberry PI3 B+ and other connectors and cables	265 grams
Total weight	**1390**

III. LOAD CALCULATION

The whole weight of the quadcopter was estimated in table 1. The Motors and propellers together produce the required thrust to lift the quadcopter in the air, the estimated AWU All Weight Up is 1.39 kg, the thrust requirements from the all motors must be at least double of 1.39 kg. The suggested payload to be used is 110 grams ($1390 + 110 = 1500$ grams), the thrust value is $1500 \times 2 = 3000$ grams, therefore each motor must produce 750 g of thrust force. Motors are chosen depending on the value of KV. It is calculated by the formula in equation1:

$$RPM = KV\ rating \times Voltage\ input \quad (1)$$

Substituting the values of RPM and Voltage input:

KV rating = 11100/11.1 = 1000 Kv.

Propellers are kind of fan which convert the rotational motion into thrust. The propeller is specified on the basis of its pitch and diameter in inches.

$$Power\ (watts) = PC \times D^4 \times P \times RPM^3 \quad (2)$$

Where:

PC: is the propellers constant (1.11 for APC propellers).

D: is the diameter of propellers in feet.

P: is the pitch of the propellers in feet.

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

RPM: is the rotations per minutes.

Substituting the values of the prototype:

$$Power(watts) = 1.11 \times (0.83333)^4 \times (0.375) \times (10)^3 \ = 200.734 \ \approx 200$$

To calculate the mass that the quadcopter could lift, the following formula must be used:

$$Mass\ (m) = \frac{thrust}{acceleration\ due\ to\ gravity} = \frac{T}{g} \quad (3)$$

Where g = 9.81, from equation 3 it's clear that the calculation of the thrust of the quadcopter should be done by the way of using the formula in equation (4):

$$Thrust\ (T) = \left[\frac{3.14}{2} * D^2 * roh * power^2\right]^{\frac{1}{3}} \quad (4)$$

Where roh in Greek is the air density. Air density is affected by changes in humidity and temperature. It's close to equal to 1.225 $kg/m\text{^}3$ at 15 Celsius with sea level in line with International standard pressure (ISA). And D: is the diameter of propellers in meters. Therefore, the thrust (T) = 17.05288499 then the total mass lifted by quadcopter is:

$$Mass\ (M) = \frac{17.05288499}{9.81} \approx 1.739\ kg$$

Empty mass of quadcopter = 1390 gram + payload 110 = 1500 grams, then the quadcopter can fly safely with this payload. The result of the calculation shows that it would capable of flying with 110 grams of the load.

IV. FLIGHT TIME TESTING

To estimate the flight time of the QC with various loads, the actual flight test has been done with loads from 110 grams to 200 grams. The flight time formula described in equation 5:

$$Time = capacity * \frac{discharge}{ADD} \quad (5)$$

Where:

Time: is the flight time of the QC in hours.

Capacity: is the capacity of the battery, and it's calculated in either in (mAh) milliAmpere hour or (Ah) Ampere hour.

Discharge is that the battery discharge that you simply allow throughout the flight. As LiPo batteries are often damaged if totally discharged, it is common practice never to discharge them by over 80%

AAD: The average ampere draw (AAD) of the QC, calculated in amperes.

To calculate the amp draw (AAD), we use the below formula of equation 6:

$$AAD = AUW * \frac{P}{V} \quad (6)$$

Where:

AAD: The average ampere draws in amperes.

AUW: The All Up Weight of the QC - the whole weight of all devices that goes up in the air, including the battery and it's usually calculated in the unit of kilograms

P: is the power needed to raise one kg of the devices in the air, expressed in watts per kg. Moreover,there is a tend to assume a conservative estimate of 170 W/kg. Some a lot of efficient systems will take less, for instance, 120 W/kg.

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

V is the voltage of the battery and it's written on the battery itself and it's calculated in the unit of Volte. Therefore, the calculations to our prototype are:

$$AAD = AUW * \frac{P}{V} = 1.5 * \frac{170}{11.1} = 22.97A$$

Then the flying time is:

$$time = 5.2 * \frac{0.8}{22.97} = 10.86 \text{ Minutes}$$

This flight time includes payload 110 grams calculations, the Table 2 shows the flight time of QC with various loads: note that the discharge battery is 0.8 and capacity is 5.2 A

TABLE 2. THE ESTIMATED FLIGHT TIME WITH DIFFERENT PAYLOAD

weight	Payload (grams)	AAD	Total weight (kg)	Time (min)
1390	110	22.97	1.5	10.86
1390	150	23.59	1.54	10.58
1390	200	22.35	1.59	10.25

V. SYSTEM ARCHITECTURE

The proposed system use: Quadcopter, RPI3, Database server, TCP/IP connection, as shown in Fig. 2

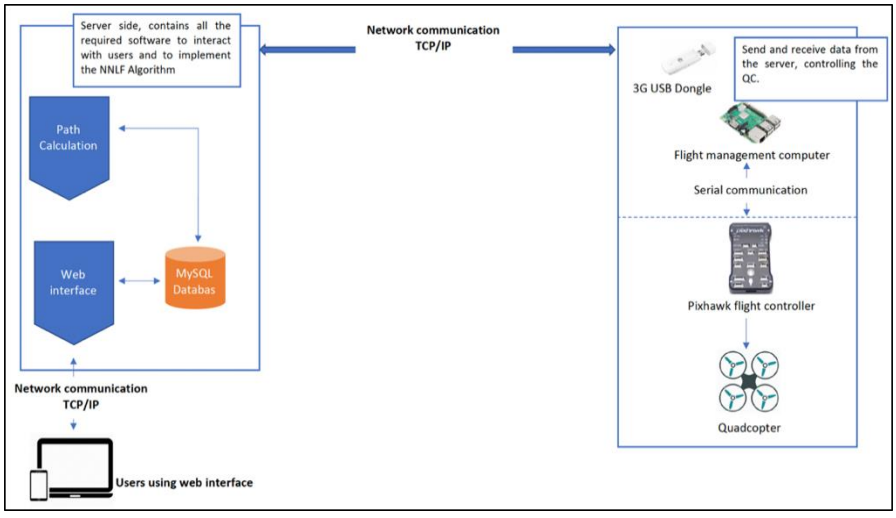


FIG. 2. PROPOSED SYSTEM.

The users will connect to the server by using the web interface to register in the system and then send the required data which is the user GPS location (longitude & latitude) and the destination GPS location, the GPS location will be referred to later on as the waypoint. The system has two privileges at the software level: administrator and the users. The admin has full privileges and he should check all the waypoints before submitted to the RPI3, after checking all the data he will approve the mission and

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

send all the waypoints to the RPI3; also he checks all the data to increase the flight safety to ensure that the drone will fly only above the allowed regions.

A. Control unit

It consists of two units; the first one is the ***onboard control unit*** which is the RPI3 in this case. Connect the Pixhawk TELEM2 port with the RPI pins (+5V, Ground, TX, and RX) as shown in Fig.3, note that the RPI can be powered using the red +V cable to the +5V pin of RPI3 or using the USB port. We use the power as in the Fig.3

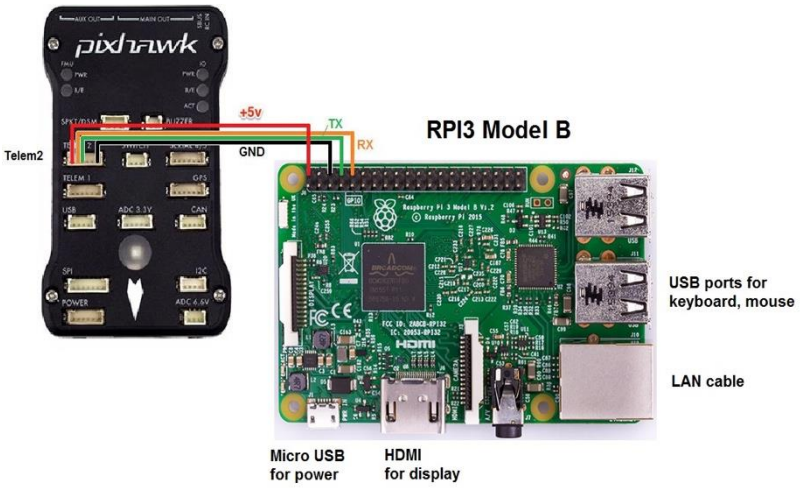


FIG. 3. CONNECTING PIXHAWK WITH RPI3.

The onboard unit is used to implement the autonomous flight and to control the quadcopter movement by changing the flight mode to GUIDED mode. Using ground station software such as mission planner or QGround control station to set the following parameters of the Pixhawk:

1. `SERIAL2_PROTOCOL <copter: SERIAL2_PROTOCOL>` = 1 to activate MAVLink protocol on serial port.
2. `SERIAL2_BAUD <copter: SERIAL2_BAUD>`= 921, then the Pixhawk can communicate with the RPI3 at baud rate equal to 921600.

The second control uni is the ***off-board control unit*** which is the radio control or the joystick with manual control of quadcopter.

B. DroneKit-Python application programming interface

Ubuntu 16.4 was used as development environment with the DroneKit-Python API. DroneKit-Python, it's a cross-platform application programming interface that can be used on many different operating systems, it helps to produce applications for Drones. These applications run on a drone's companion PC and they increase the capabilities of the autopilot by performing tasks that are computationally intensive and need a low-latency link (for example computer vision & Path planning). This API has many versions for each developing environment, its available for Android OS as DroneKit-Android, for Python programming language as DroneKit-Python and for IOS devices it's still under development, in [14] and [15] include more detailed about this API is found.

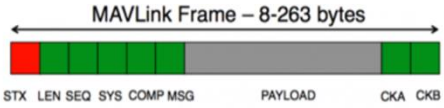
C. MAVLink Protocol

MAVLink Protocol is a serial protocol commonly used in sending data and commands between vehicles such as Drones or Rovers and ground control stations. It is a very light-weight messaging protocol for communication with drones and between onboard drone equipment's. MAVLink is a binary

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

telemetry protocol designed for resource-constrained systems and bandwidth-constrained links [17]. The protocol defines a set of messages which will be found in [16] with all details of each message. The packet size of MAVLink version 1 is 8 bytes including the header (STX) and the checksum. Because MAVLink doesn't need any additional framing it is well suited for applications with very limited communication bandwidth. It is Very reliable. MAVLink had been used from 2009 to communicate between a Ground control station and drone, and between drone autopilot with MAVLink enabled drone camera. It is supporting various programming languages such as C++ and Python, running on different microcontrollers/operating systems (including ARM7, ATMega, dsPic, STM32 and Windows, Linux, MacOS, Android and iOS). Since the size of the packet is 8 bytes there will be 255 concurrent systems on the network. its support offboards and onboard communications. Fig. 4 describes the frame structure of this protocol [17].



Byte Index	Content	Value	Explanation
0	Packet start sign	v1.0: 0xFE (v0.9: 0x55)	Indicates the start of a new packet.
1	Payload length	0 - 255	Indicates length of the following payload.
2	Packet sequence	0 - 255	Each component counts up his send sequence. Allows to detect packet loss
3	System ID	1 - 255	ID of the SENDING system. Allows to differentiate different MAVs on the same network.
4	Component ID	0 - 255	ID of the SENDING component. Allows to differentiate different components of the same system, e.g. the IMU and the autopilot.
5	Message ID	0 - 255	ID of the message - the id defines what the payload "means" and how it should be correctly decoded.
6 to (n+6)	Data	(0 - 255) bytes	Data of the message, depends on the message id.
(n+7) to (n+8)	Checksum (low byte, high byte)	ITU X.25/SAE AS-4 hash, **excluding packet start sign, so bytes 1..(n+6)** Note: The checksum also includes MAVLINK_CRC_EXTRA (Number computed from message fields. Protects the packet from decoding a different version of the same packet but with different variables).	

FIG. 4. FRAME STRUCTURE OF THE MAVLINK PROTOCOL. NOTE THAT "FIG." IS ABBREVIATED [17]

The sender always fills in the System ID and Component ID fields so that the receiver knows where the packet came from. System identification is a unique ID for each vehicle or ground control station. Ground stations by default use a system id 255 and vehicles use default value equal to 1 (it can be modified by setting the SYSID_THISMAV parameter). The Component ID for the GCS or flight controller equal to "1". Another MAVLink device on the vehicle (i.e. companion computer, GPS) must use the same System ID as the flight controller (Pixhawk) but uses a different Component ID. Fig 6 demonstrates the communication messages between the drone and the QGC

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

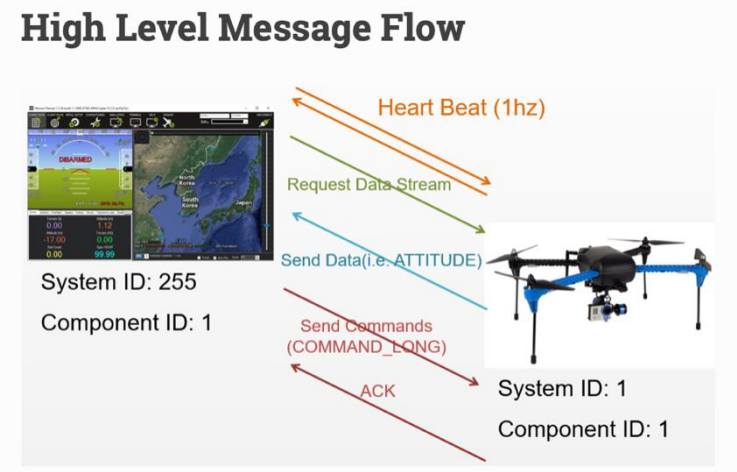


FIG. 5. COMMUNICATION BETWEEN DRONE AND QGC. NOTE THAT "FIG." IS ABBREVIATED [13]

D. MAVProxy

A UAV ground station software package for MAVLink protocol-based systems. It's a fully-functioning GCS for UAVs. It's a command line GCS that is lightweight (requires very little resources) and it's an open source GCS written in python. It can be used to share Drone-GCS connections with other ports/GCS. The drone will offer to open port that a GCS can connect to, For SITL drone, the port is TCP: 127.0.0.1:5760. Note that the MAVProxy can be used to open port number to connect the drone with other software which is a very useful feature that is used in this project. The full list of commands line and more details are found in [17].

E. ArduPilot

ArduPilot is a free software package and it contains all the software needed to monitor the drone, reading sensors inputs and control the drone according to the data received from the sensors. This software is available for programmers and researchers. It's been developed over 5+ years by a team of various skilled engineers and computer scientists. It's the only autopilot computer code capable of controlling any vehicle system possible, from standard airplanes, multirotor, and helicopters, to boats and submarines [9]. ArduPilot firmware works on various boards (hardware such as Pixhawk and Navio) to control UAVs of all types. More detailed for this open source software found in [13].

F. Software In The Loop (SITL) Simulation

The SITL simulator contains three main different versions of vehicles such as Plane, Copter and Rover that can be run without any hardware. SITL allows testing the behavior of the code without hardware. SITL enables us to run ArduPilot on the PC directly, without any microcontroller or hardware. Since the ArduPilot is cross-platform which can be used on different operating systems SITL is also used this feature to run on many environments such as Linux, Windows or MAC OS. For example, SITL can simulate: (multi-rotor aircraft, fixed-wing aircraft, ground vehicle, underwater vehicles, and camera gimbals). Fig. 6 shows the port numbers that can be varied. For instance, the ports between ArduPilot and the SITL on the Fig.6 are 5501/5502 but they can be different to be 5504/5505 or other port numbers according to the environment [13].

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

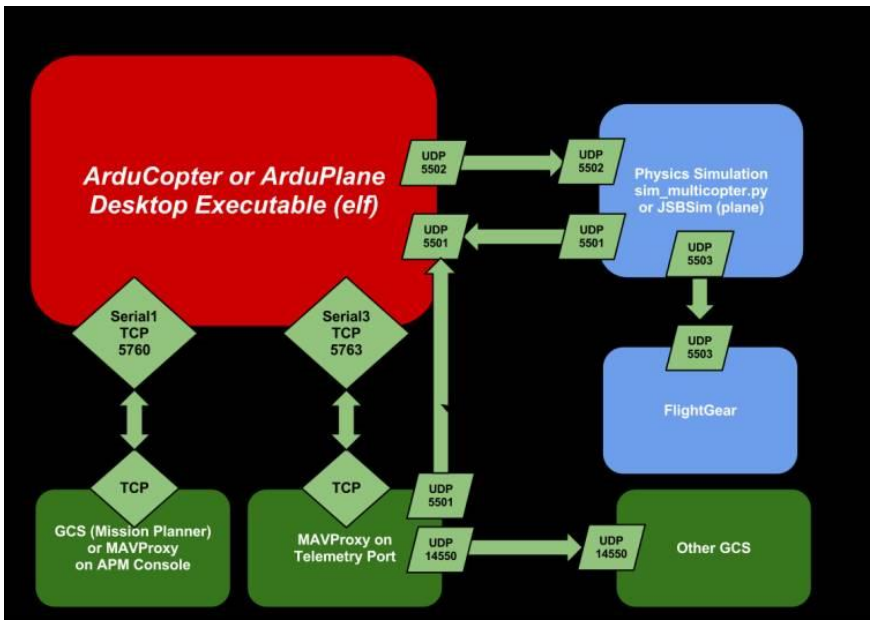


FIG. 6. SITL STRUCTURE AND PORT NUMBER. NOTE THAT "FIG." IS ABBREVIATED [13]

By replacing the drone hardware and flight controller with the SITL software, the general system block diagram is shown in Fig. 7

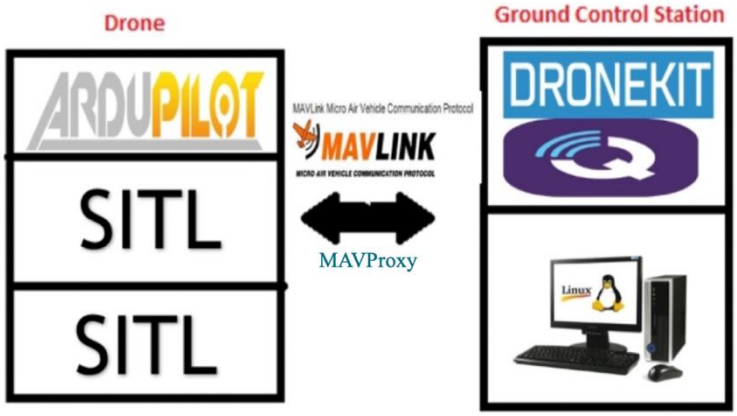


FIG. 7. SITL SIMULATION BLOCK DIAGRAM

Starting the mission for the QC to visit two waypoints inside the University of Technology, as presented in Fig.8 below , three states of the system, initial state which include parameters initialization and sensors checking, if the initialization passed without any error then the second state of arming the motors and taking off to specified altitude which is in this case 10 meters are executed, and then enters the third state of flying to the waypoints received from the end users. The QC arming and taking off to the 10 meters, and the system will give an indication that the specified altitude is reached as described in Fig. 8-A. After these states, the QC should fly to the waypoints and then go back to the home location again and enter to land mode to be ready for a new mission, as shown in Fig. 8-B

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

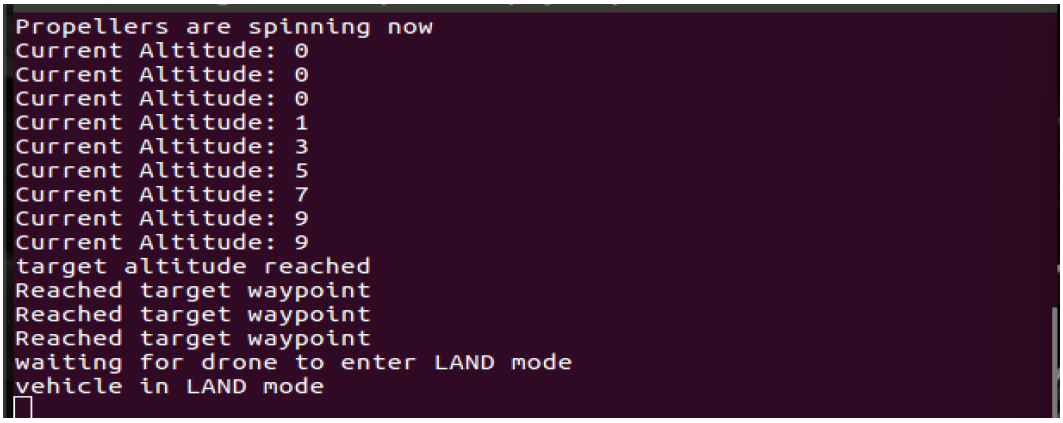


FIG. 8-A SYSTEM STATUS

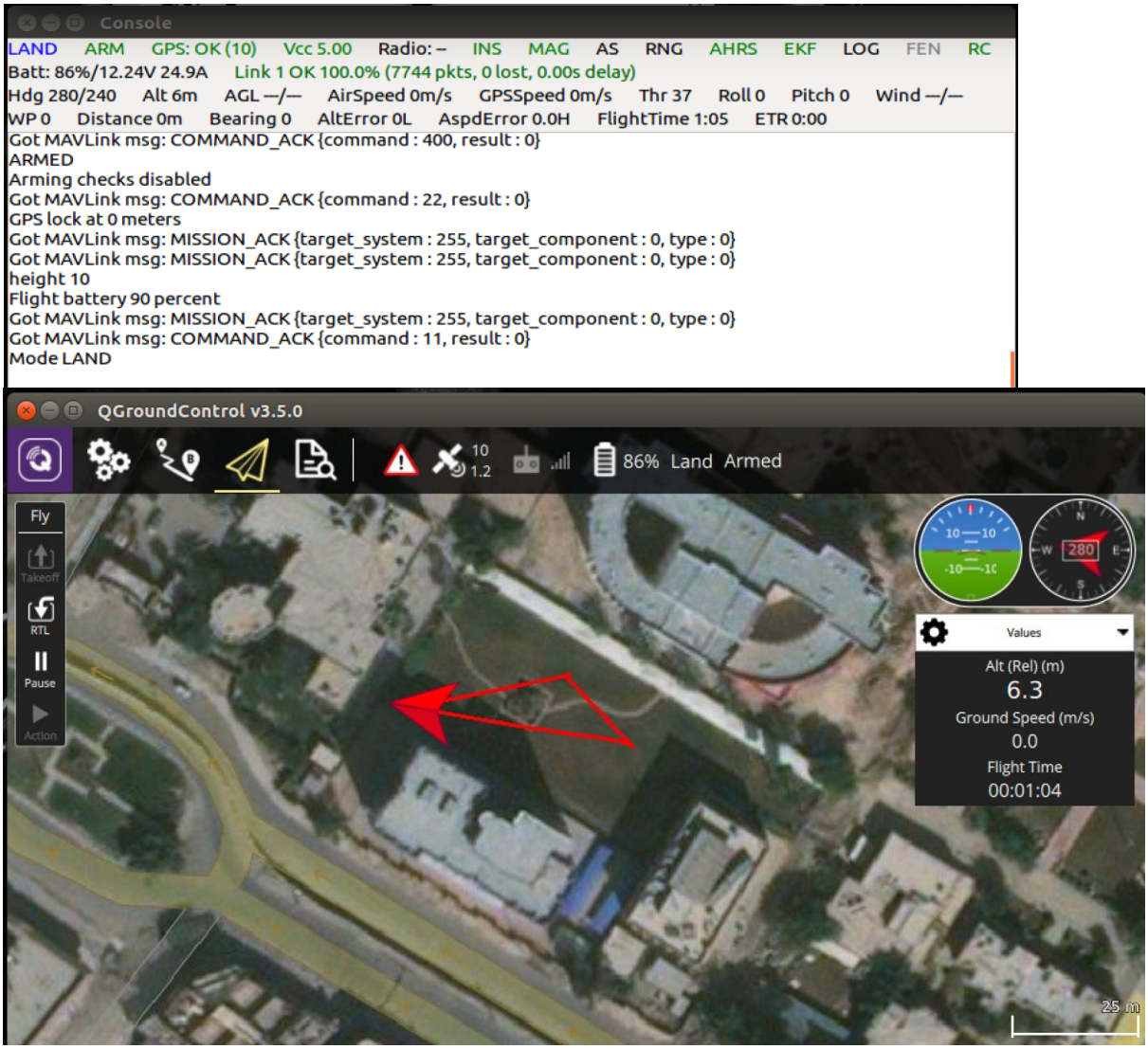


FIG. 8-B DRONE IMPLEMNT AUTONOMOUS FLIGHT WITH SITL SIMULATOR

G. Autonomous flight algorithm

The autonomous flight is implemented using Python programming language with DroneKit API and MAVProxy. The script is divided into many functions, each one has a specific task. The function is listed as follows:

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

1- **ConnectMyCopter():** this function is used to establish the communication between the onboard unit and the quadcopter. In the case of real drone, it will use the serial port ttys0, and for SITL drone it will use the local host IP address which is 127.0.0.1 with port number 5760. The flow chart in Fig. 7 describe the process of communication.

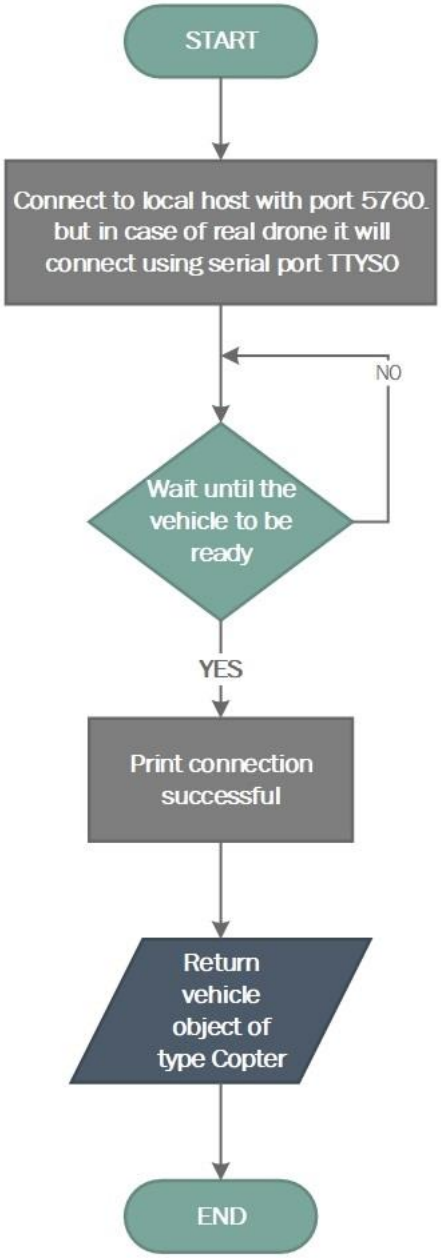


FIG. 6. CONNECTION TO QUADCOPTER FLOWCHART

2- **arm_and_takeoff(targetHeight):** this function is used to arm the motors of the quadcopter and to change the flight mode to be GUIDED mode to take the control f quadcopter using the RPI3, then take off to a specified height, the function checks each state to ensure the right operations for the quadcopter. The flow chart for this function is shown in the Fig. 7

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

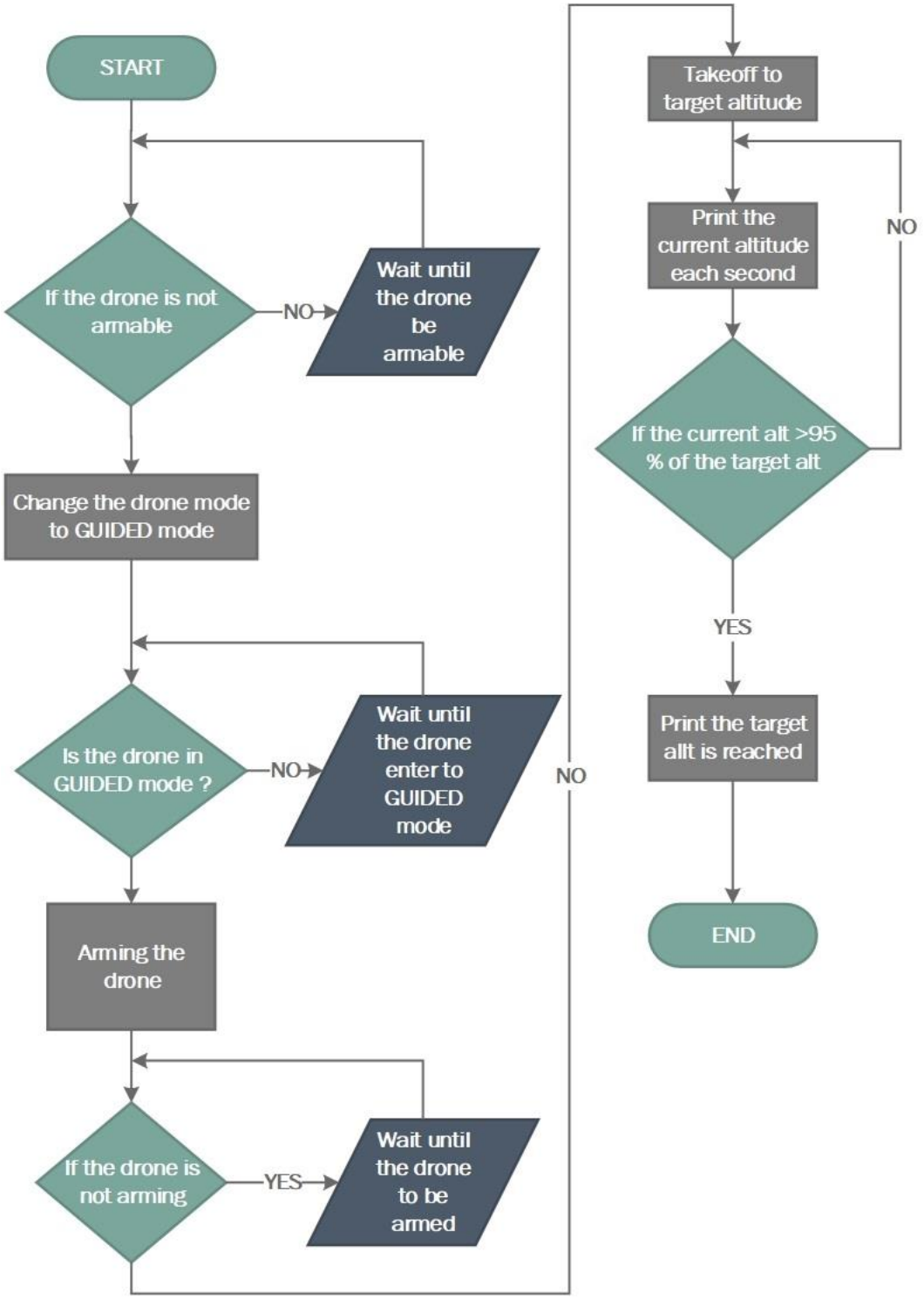


FIG. 7. ARM AND TAKEOFF FUNCTION FLOWCHART

3- **get_distance_meters(target location, current location):** this function, takes two variables the target location and the current location and returns the distance between these two locations in meters.
4- **goto (target location):** it commands the drone to fly to the target location and it will calculate the remaining distance to the target location each second, and if

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

the remaining distance is 30 meters then the drone will reach to target location. The flow chart in Fig. 8 describes this function.

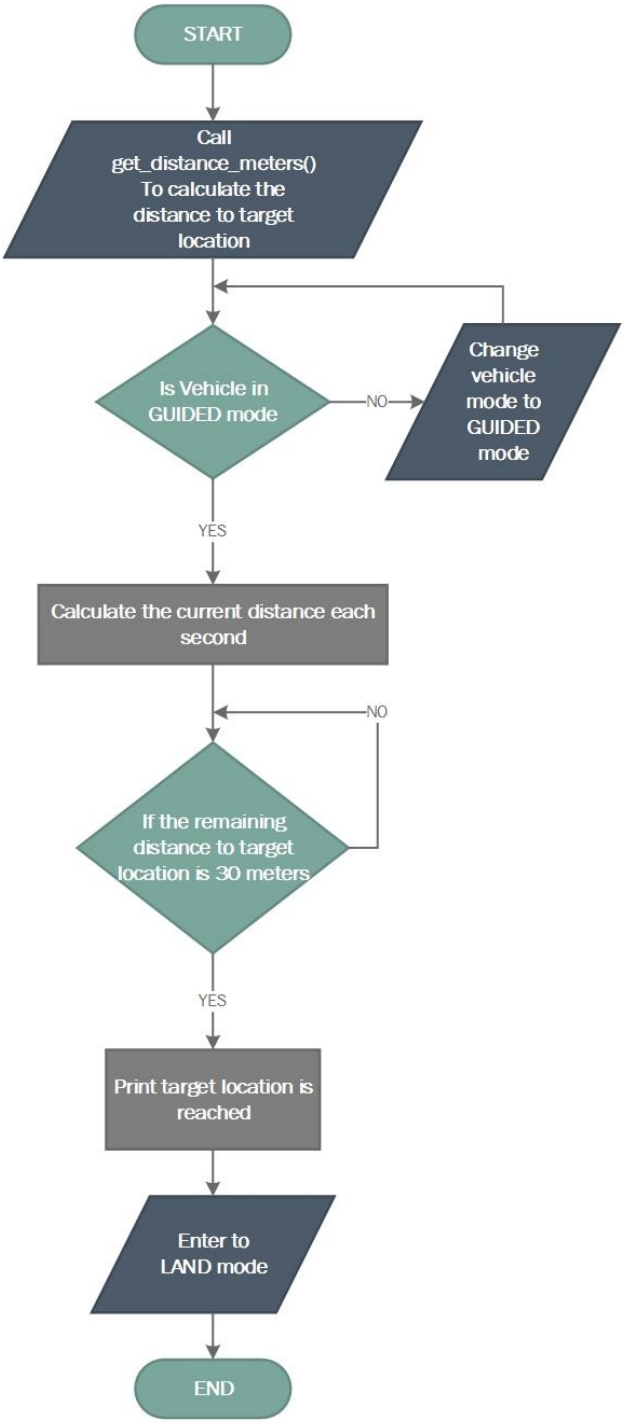


FIG. 8. GO TO TARGET LOCATION FUNCTION FLOWCHART

VI. CONCLUSION

The implementation of the autonomous flight on SITL simulation was successful and excellent. The Quadcopter will execute an autonomous flight using the concept of companion PC. Raspberry PI 3 will control the Quadcopter by commanding the controller of the drone (Pixhawk) by using DroneKit-Python API to send MAVLink messages to the Ardupilot. This concept is useful to perform an

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

additional task to the autopilot and provides such a smart capability like image processing and path planning which cannot be done by the flight controller alone. The same algorithm of the SITL simulator was used in the real Quadcopter and the result was very good with three to five meters difference when landing to home location or to another waypoint due to the GPS signal error. This error can be eliminated by applying some enhancement to the Quadcopter landing which is called precision landing using computer vision technique which represents a future work for this project. Also, in case there is multiple destination; the path planning is required to select the shortest path to visit all the location once that can be done in Travel Sales man Problem (TSP) which is the future work for this project.

VII. Discussion

Controlling the Quadcopter was done using RPI3, as a result the most important feature which is autonomous flight was perfectly implemented on both SITL and real Quadcopter without using RC receiver or manual controlling. The results in this Project is the behavior of the Quadcopter, which means how we can do the takeoff, landing, and fly to specific location automatically according to the user input from the system interface. To enhance the flight of Quadcopter especially for obstacle avoidance and precision landing, techniques such as ultra sonic sensors to implement obstacle avoidance and computer vision to detect marker location and land over it to enhance the landing accuracy. Depending only on the GPS signal alone will give varying error in landing between three and five meters according to the GPS signal.

REFERENCES

[1] Syed Omar Faruk Towaha, Clerk Maxwell "Building Smart Drones with ESP8266 and Arduino", A Treatise on Electricity and Magnetism, BIRMINGHAM – MUMBAI, 2018.

[2] M Muhammad, D Swarnaker, and M Arifuzzaman, "Autonomous Quadcopter for Product Home Delivery", International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) 2014.

[3] Alberto Lisanti and Giorgio Venezia , "New Frontiers of Delivery Services Using Drones: a Prototype System Exploiting a Quadcopter for Autonomous Drug Shipments", IEEE 39th Annual International Computers, Software & Applications Conference 2015.

[4] Kenzo Nonami ,"Research and Development of Drone and Roadmap to Evolution", Autonomous Control Systems Laboratory Ltd, WBG Marive West 32F, 2-6-1 Nakase, Mihama-ku, Chiba-city, Chiba 261-7132, Japan 2018.

[5] Eric N. Johnson and and Daniel P. Schrage, "The Georgia tech unmanned aerial research vehicle: GTMax," School of Aerospace Engineering, Georgia Institute of Technology, Atlanta 2003.

[6] Azade Fotouhi, Ming Ding and Mahbub Hassan, "Understanding Autonomous Drone Maneuverability for Internet of Things Applications", University of New South Wales (UNSW), Sydney, Australia 2017.

[7] Matthias Heutger, "UNMANNED AERIAL VEHICLE IN LOGISTICS", Marketing & Development, DHL CSI 53844 Troisdorf, Germany 2014.

[8] Terry Kilby and Belinda Kilby, "Make: Getting Started with Drones", Elevated Element, Printed in the United States of America 2016.

[9] Yasir Ashraf Abd Rahman1, Mohammad Taghi Hajibeigy1, Abdulkareem Shafiq Mahdi Al-Obaidi1, and Kean, "Design and Fabrication of Small Vertical-Take- Off-Landing Unmanned Aerial Vehicle ", MATEC Web of Conferences 152, 02023 (2018).

[10] Cameron Roberts, "GPS Guided Autonomous Drone", University of Evansville, College of Engineering and Computer Science, April 25, 2016.

[11] Prof. Dr. Salih Mahdi Al-Qaraawi, Dr. Aymen Dawood Salman and Majida Saud Ibrahim, "Performance evaluation of Ad-hoc quadcopter monitoring system", a thesis submitted to the computer engineering department, university of technology 2018.

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

[12] Aicha Idriss Hentati, Lobna Krichen, Mohamed Fourati and Lamia Chaari Fourati , "Simulation Tools, Environments and Frameworks for UAV Systems Performance Analysis", Laboratory of Technology and Smart Systems (LT2S), Digital Research Center of SFAX (CRNS), University of Sfax, Tunisia 2018.

[13] ArduPilot software documentation [Online]. Available on: http://ardupilot.org/about. [Accessed 2 April 2019].

[14] DroneKit by 3DR Robotic [Online]. Available: http://dronekit.io [Accessed 5 April 2019].

[15] DroneKit-Python documentation [Online]. Available: http://python.dronekit.io/1.5.0/about/overview.html [Accessed 7 April 2019].

[16] MAVLink common messages set and documentation [Online]. Available: https://mavlink.io/en/messages/common.html [Accessed 3 April 2019].

[17] MAVProxy documentation [Online]. Available: http://ardupilot.github.io/MAVProxy/html/index.html [Accessed 15 March 2019].

Received 24 April 2019; Accepted 8 October 2019

 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352