RESEARCH ARTICLE OPEN ACCESS

Designing and implementation of quadcopter drone for aerial surveillance and other applications

Mrs. Smruti A. Pore, Dr. Mrs. L.S. Admuthe
Assistant Professor, Professor,Department of Electronics Engineering, DKTES' TEI, Ichalkaranji.
Corresponding Author: Mrs. Smruti A. Pore

ABSTRACT: Now a days, aerial photography is used for many applications like monitoring, surveillance for military purpose etc. Quadcopter is a cost effective, safe and reliable solution for aerial photography. In this paperworking and controlling of quadcopter is described.Quadcopter can lift vertically and maintain stable position also it can change its horizontal position. It is controlled remotely usinglaptop, Arduinoand ESP8266 wife device. The position of quadcopter is sensed using accelerometer and gyroscope. For controlling quadcopter PID programming is done in Java NetBeans. ESP 8266 device is connected to the laptop using UDP communication and Arduino controller is used to give commands to the quadcopter. Lifting mechanism of the quadcopter consists of BLDC motors with Electronics Control Unit. Thus this system controls the drone assembly remotely using laptop

Keywords: BLDCMotor, Electronic Speed Controller (ESC),PID Controller, Pitch, Roll, Quadcopter

---

Date Of Submission: 30-06-2019 Date Of Acceptance: 19-07-2019

I. INTRODUCTION

In surveillance system quadcopter plays important role. It can work in remote areas as well as in hazardous conditions. It is useful in unmanned area. Quadcopter operation, movement is a difficult and challenging problem. Its working includes uplifting of quadcopter assemblyand six degrees of freedom. Stabilization of quadcopter is equally important as it freely moves in air without any friction. In [1] communication with quadcopter is done using ZigBee. It restrict area of surveillance. This paper describes basic quadcopter system and its controlling using closed loop system that includes Proportional-Integral-Derivative (PID) controller using Wi-Fi connection.

II. BLOCK DIAGRAM

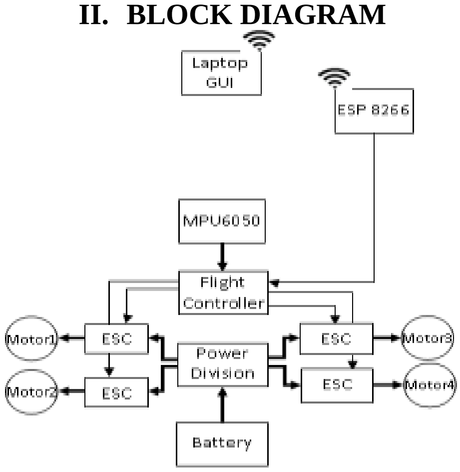

Fig. 1 Quadcopter Block Diagram

III. BLOCK DIAGRAM DESCRIPTION

- Quadcopter assembly consists of flight controller, battery, power division block, Electronic Speed Controller (ESC) and BLDC motors. This mechanism is used to provide thrust and movement to the quadcopter.
- Quadcopter consists of group of four propellers. Each propeller is connected to the motor through the reduction gears. All the propellers axes of rotation are fixed and parallel [6].
- Laptop computer is wirelessly connected to Flight Controller (Arduino) via ESP8266 Wi-Fidevice by using UDP protocol. GUI is developed in laptop computer to provide access to quadcopter. It displays control signals which include upward, downward, left, right keys as well as start and exit button and labels to display current speed of 4 motors, accelerometer, and gyroscope and distance readings.
- The GUI programming and PID programming is done in JAVA NetBeans. ESP8266 chip is used for data transferring and receiving. UDP programming is used for the wireless connection. Arduino board is used for receiving data from ESP8266 and generating control signals for BLDC motors.
- First control signals are given by Laptop computer to ESP8266. Arduino receives control signals from ESP8266 through serial data communication that means serial transmitter, receiver pins.

- The accelerometer and distance sensor are connected Arduino board. It sends sensor readings to laptop through ESP8266 by UDP communication.
- In laptop computer depending on reading of sensor, roll, pitch, yaw angle is calculated from that angle, PID is implemented. Thus closed loop control system formed which controls position of quadcopter and stabilizes it.

IV. HARDWARE DESCRIPTION

a. Flight Controller:

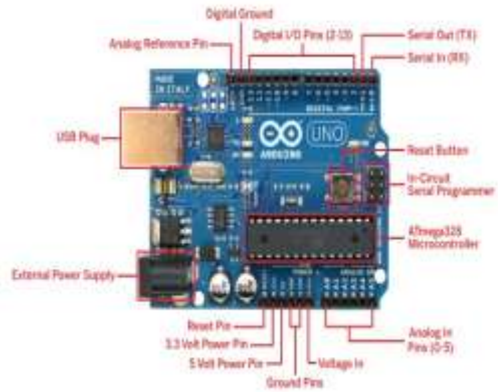

Fig. 2 Arduino Board

Flight controller consists of ATmega32 controller. It receives commands from laptop and accordingly sets speed and direction of BLDC motors of quadcopter. It has 32 KB (with 0.5 KB used for the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

It has 14 digital pins which can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kilo Ohms.

In addition, some pins have specialized functions: Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

b. ESP8266 Wi-Fi Module

Fig. 3 ESP 8266 Wi Fi Module

ESP-12E Wi-Fi module is developed by Ai-thinker Team. Core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi MAC/BB/RF/PA/LNA, on-board antenna. The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. Users can use the add modules to an existing device networking, or building a separate network controller. ESP8266 is high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or I2C/UART interface). ESP8266EX is among the most integrated Wi-Fi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

c. ESC(Electronic Speed Controller)

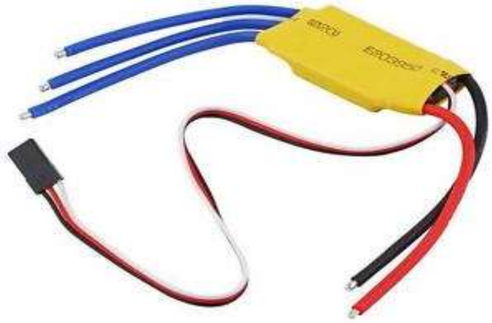

Fig. 4 Electronic Speed Controller (ESC)

This is fully programmable 30A BLDC ESC with 5V, 3A BEC. This module is basically used to drive BLDC motors with very high speed from 1000 rpm to 2000 rpm.It can drive motors with continuous 30Amp load current. It has sturdy construction with 2 separate PCBs for Controller and ESC power MOSFETs. It can be powered with 2-4 lithium Polymer batteries or 5-12 NiMH / NiCd batteries. It has separate voltage regulator for the microcontroller for providing good anti-jamming capability.

V. CONNECTIONS:

BLDC ESC has three Blue wires coming out from the one end which are to be connected to the BLDC motor. On the other end, it has red and black wires which are to be connected to the battery. From the 3pin servo cable coming out of the ESC, connect the Brown cable to the 'GND' pin on FRDM kl25z. Connect the Yellow cable to any digital pin.

Control Signal:

30A BLDC ESC requires standard 50-60Hz PWM signal from any remote control as throttle input. You can also generate similar input signal from the microcontroller for making your own customized flying platform. Throttle speed is proportional to the width of the pulse. Maximum throttle position is user programmable. In general throttle is set at zero for 1mS pulse width and full at the 2mS pulse width.

a. BLDC Motor:

Fig. 5 Brushless DC Motor

Brushless DC Motors- We require high quality reliable motors with rapid response in order to control the quadcopter. If one or several of the motors at some point during a flight experience any problems it would be devastating for the quadcopter, and can at worst endanger the quadcopter itself, property and people. Furthermore it is important that the motors are powerful enough to be able to lift the quadcopter and perform various aerial movements. We also require the motors to have a fast response in order to ensure a more stable flight. Finally we require that the motors are close to vibration free, as any vibration will cause noise in our IMU measurements. Based on these criteria we decided to acquire the SunnySky Angel A2212 KV1400 Brushless Motor. It is a brushless motor designed for remote controlled airplanes as well as quadcopters. Each motor can give a thrust of 820 grams at 136 watt, based on ESC and propellers, which means that quadcopter could theoretically fly at approximately 35 percent capacity.

b. MPU6050:

Fig. 6 MPU 6050 Accelerometer and Gyroscope Sensor

Accelerometer:

- It provides output in terms of three axis coordinates. mass deflection is measured as a change in capacitance between the proof mass and sensing plates
- Internal circuitry converts the tiny capacitance to a voltage signal which is digitized and output

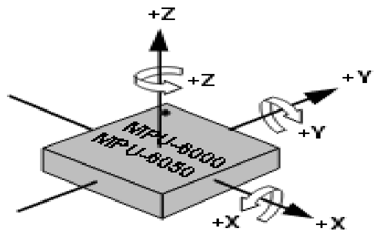

Fig. 7 Working of Accelerometer

- Accelerometers also have a sensitivity, usually expressed in mV/g

- Each accelerometer has a zero-g voltage level, you can find it in spec
- Divide the zero-g level corrected reading by the sensitivity to produce the final reading
- Computing orientation from an accelerometer relies on a constant gravitational pull of 1g (9.8 m/s^2) downwards
- Gyroscope
- A gyroscope measures angular velocity (the rate of change in orientation angle), not angular orientation itself
- Must first initialize the sensor position with a known value (possibly from the accelerometer), then measure the angular velocity (ω) around the X, Y and Z axes at measured intervals (Δt)
- ω × Δt = change in angle
- The new orientation angle is the original angle plus this change
- Sensor Fusion:

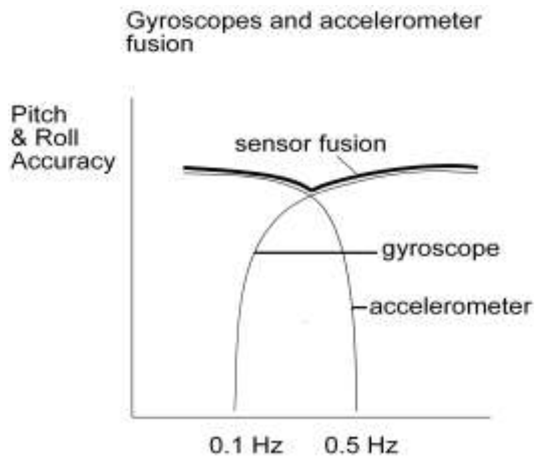

Fig. 8 Graph for output of MPU 6050

c. Battery

FIG. 9 High Power battery used for BLDC motor

Lipo Battery 2200mAh 11.1V Specifications:
- Brand: HPB
- Product Type : Lithium Polymer Battery Pack
- Capacity : 2200mAh
- Voltage : 11.1V
- Max Continuous Discharge : 30C(66.0A)
- Balance Plug : JST-XH
- Max Burst Discharge : 60C(132.0A)
- Discharge Plug : XT-60
- Charge Rate : 1-3C Recommended, 5C Max
- Higher charge voltage than standard lipos.
- Perfect for high performance models.
- Higher capacity than comparable size packs.
- Heavy duty silicone wire discharge leads.Rechargeable battery.

VI. SYSTEM FLOWCHART

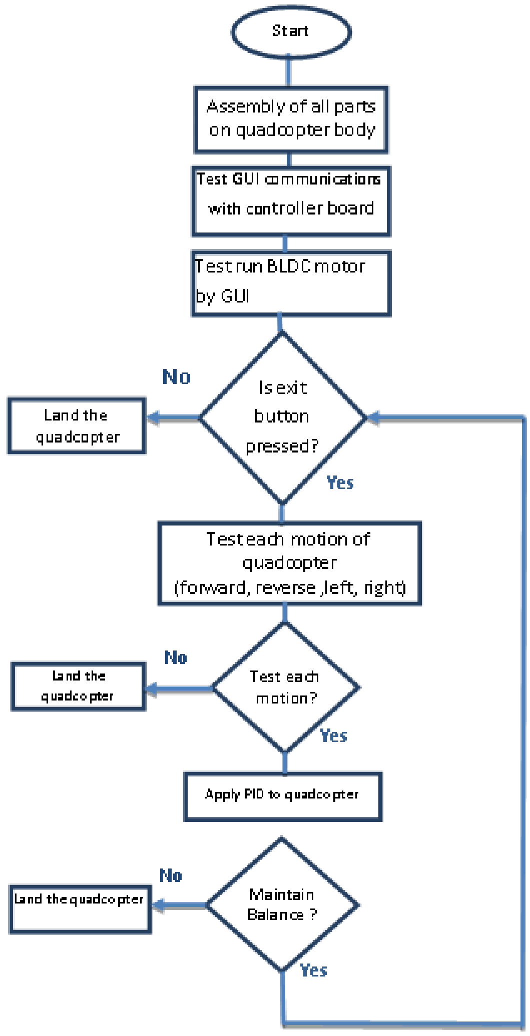

Fig. 10 Quadcopter Working flowchart

Working of quadcopter is described using figure 10. Steps for implementation of quadcopter are as given below. According to flowchart all possible conditions of quadcopter are considered and corrective action is taken.

VII. SOFTWARE IMPLEMENTATION

Software implementation of quadcopter consists of following modules:

a. GUI implementation for Laptop (Java Programming):

Graphical User Interface provides interactive mode for user to control position, direction and movement of quadcopter. It displays accelerometer and gyroscope readings from sensor. Through GUI user can input required values of pitch and roll to quadcopter.

b. PID controller program in Laptop (Java Programming):

This module actually implements PID control for quadcopter using sensor values as input and implements closed loop control system.

Algorithm for PID controller: It performs calculation for pitch and roll of quadcopter motors.

1. Set initial motor speed values for quadcopter motors.

m1=m2=m3=m4=1000

These are the initial speeds of 4 quadcopter motors required to provide thrust to the quadcopter.

2. Initialize following parameters for PID controller

Proportional constant for pitch: Pitch_Kp = 1
Integral constant for pitch: Pitch_Ki = 0.25
Proportional constant for roll: roll_Kp = 1
Integral constant for roll: roll _Ki = 0.25

3. Read sensor values i.e. accelerometer and gyroscope values from quadcopter which provides current position of quadcopter given as following

ax=Accelerometer X coordinate
ay=Accelerometer Y coordinate
az=Accelerometer Z coordinate
gx= Gyroscope X coordinate
gy= Gyroscope Y coordinate
gz=Gyroscope Z coordinate

4. Calibrate sensor readings with 2000 iterations with formulae as given in equations 1 to 3.

$$cal_{gx} = cal_{gx} + gx \quad (1)$$

$$cal_{gy} = cal_{gy} + gy \quad (2)$$

$$cal_{gz} = cal_{gz} + gz \quad (3)$$

5. Normalize values by dividing by 2000
6. Find pitch angle using equation 4 and 5

$$angle_{pitch} = angle_{pitch} + (gx - cal_{gx}) + 0.0003053435 \quad (4)$$

$$angle_{roll} = angle_{roll} + (gy - cal_{gy}) + 0.0003053435 \quad (5)$$

7. Calculate Accuracy factor

$$Acc_{vector} = \sqrt{\left(\frac{ax}{16384}\right)^2 + \left(\frac{ay}{16384}\right)^2 + \left(\frac{az}{16384}\right)^2} \quad (6)$$

8. Calculate correction factor

For pitch:

$$angle_{pitch_{acc}} = sin\left(\frac{\left(ay/16384\right)}{Acc_{vector}}\right) * 57.296 \quad (7)$$

9. Calculate correction factor

For roll:

$$angle_{roll_{acc}} = sin\left(\frac{\left(ax/16384\right)}{Acc_{vector}}\right) * (-57.296) \quad (8)$$

10. Calculate process variable for pitch and roll factors:

$$angle_{pitch} = angle_{pitch} * 0.9996 + angle_{pitch_{acc}} * 0.0004 \quad (9)$$

$$angle_{roll} = angle_{roll} * 0.9996 + angle_{roll_{acc}} * 0.0004 \quad (10)$$

11. Find error using set point and process variable

$$error_{pitch_i} = (angle_{pitch} * pitch_p) + error_{pitch_i} \quad (11)$$

$$error_{roll_i} = (angle_{roll} * roll_p) + error_{roll_i} \quad (12)$$

12. Calculate coefficients using error calculated:

$$pitch_{pi} = (error_{pitch} * pitch_p) + error_{pitch_i} \quad (13)$$

$$roll_{pi} = (error_{roll} * roll_p) + error_{roll_i} \quad (14)$$

13. Finally new values for motor speed are calculated.

$$m1 = m1 + roll_{pi} \quad (15)$$

$$m2 = m2 - roll_{pi} \quad (16)$$

c. UDP socket programming for wireless connection for laptop (Java Programming)

d. UDP socket programming for data transmission in ESP8266.

Modules c and d establish wireless connection between laptop and quadcopter. It implements UDP socket program for transfer of data with high speed.

e. Quadcopter motor interfacing and serial interfacing of ESP8266 (Arduino Programming):

This module provides pulses to ESC (Electronic Speed Control) to control quadcopter motors. It is implemented in Arduino Board.

VIII. RESULTS AND CONCLUSION

Fig. 11 GUI in JAVA Netbeans

GUI of quadcopter is implemented in Java neatbeans. As shown in fig. 11, it displays sensor readings for accelerometer and gyroscope. It provides control panel for direction control of quadcopter and provides display of parameters of quadcopter such as pitch and roll.

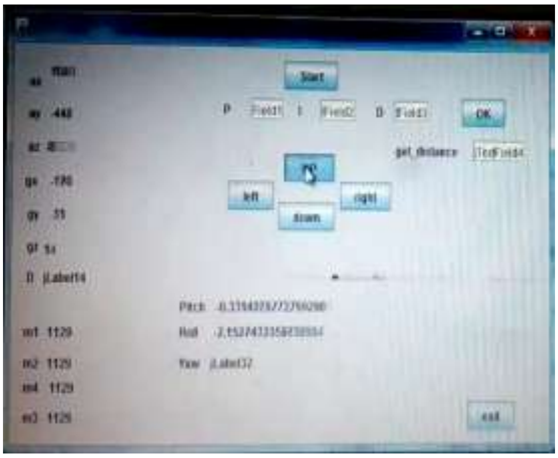

Fig. 12 Output of GUI in Java NetBeans

As shown in fig. 12 various readings of sensor are collected from ESP8266 using UDP port. Pitch and roll of quadcopter are calculated and displayed on laptop. Simultaneously motor speeds are calculated and transmitted to quadcopter using WiFi.

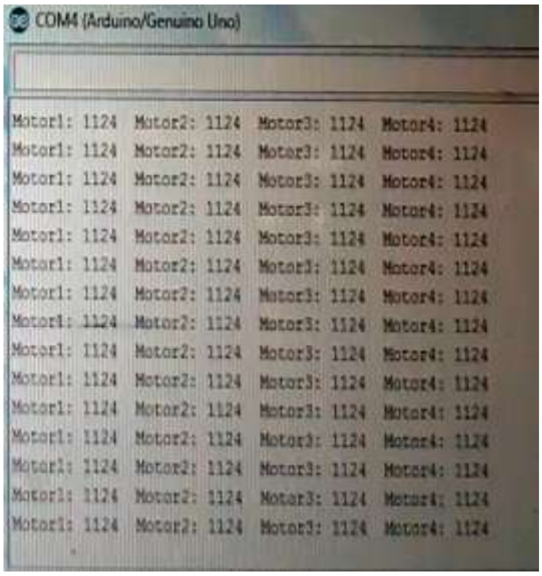

Fig. 13 Output of Arduino serial monitor

As shown in figure 13 speed of motor received from laptop by Aurdino board is displayed in serial window.

Speed of quadcopter motor varies between 1000 rpm to 2000 rpm.

When motor speed reaches between 1000 rpm to 1500 rpm, required thrust of quadcopter is reached and it is lifted above floor level by approx. 2 feet.

PID Controller provides better stability for quadcopter.

REFERENCES

[1]. Atheer L. Salih, M. Moghavvemil, Haider A. F. Mohamed and Khalaf Sallom Gaeid (2010). “Flight PID controller design for a UAV Quadcopter.”

[2]. Allison Ryan and J. Karl Hedrick (2005). “A mode-switching path planner forUAV- assisted search and rescue.” 44th IEEE Conference on Decision andControl, and the European Control Conference 2005.

[3]. Bouabdallah, S.; Noth, A.; Siegwart, R.; , "PID vs LQ control techniques applied to an indoor micro quadrotor," Intelligent Robots and Systems, 2004. (IROS 2004)

[4]. http://arduino-esp8266.readthedocs.io /en/latest/esp8266wifi/readme.html

[5]. https://quadstardrones.com/part-3-reducing-drag/

[6]. Claudia Mary, Luminita Cristiana Totu and Simon Konge Koldbæk,”Modelling and Control of Autonomous Quad-Rotor”, Faculty of Engineering, Science and Medicine University of Aalborg, Denmark (2010)

Mrs. Smruti A. Pore" Designing and implementation of quadcopter drone for aerial surveillance and other applications" International Journal of Engineering Research and Applications (IJERA), Vol. 09, No.07, 2019, pp. 23-28