QUADCOPTER

Albert Jose Pottams[1], V. Harikrishnan[1], R. Sankar[1], Balu Raveendran[2], Sukanya R. Warier[2]

Student, Department of Applied Electronics & Instrumentation, Rajagiri School of Engineering & Technology, kakkanad[1]

Faculty, Department of Applied Electronics & Instrumentation, Rajagiri School of Engineering & Technology, kakkanad[2]

Abstract: This paper focuses on the design and implementation of Xbee controlled Quadcopter system hardware along with the development of the Flight controller system. Wireless communication between the user (PC in this case) and the Quadcopter is established using the 802.15 architecture XBEE. The Quadcopter balancing condition is sensed by the MPU-6050 gyro-accelerometer sensor. The signals from the sensor are processed by the Arduino Uno on board the Quadcopter. The PID flight control system is implemented in the Arduino board with options to adjust the P, I and D values through Xbee communication. It has 2 process loops- a PID loop and a PID rate loop. The XCTU software is used to transmit user input through the XBEE communication channel. P, I and D values are adjusted in the PID rate loop for accuracy and experimental testing reveals that the Quadcopter can hover with sufficient stability.

Index Terms— Arduino, Flight controller program, Multirotor,Multirotor Wireless communication, PID control, Quadcopter user interface, Xbee, .

1. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAV) has seen a significant rise in recent years. The flexibility in their use in practical day to day applications including fields such as military, agriculture, real time surveillance or even as a hobby has been a major factor.

In recent times development of UAV's with more responsive user interfaces through pre-programmed or constantly reactive set-ups as well as autonomous drones has been given a lot of importance. The Quadcopter stands out as one of the most dynamically stable UAV's in flight. An attempt to develop a more programmable user interface with the flight controller on board the Quadcopter has been made as well as the results have been documented in this paper.

The Quadcopter has 4 motors whose speed of rotation and direction of rotation change according to the user's desire to move the device in a particular direction. The rotation of motors change as per the transmitted signal sent from the transmitter for which we have used the Xbee module. The basic idea and motivation behind an Xbee controlled Quadcopter system is to interface the control system and the sensors to the base station/computer. The onboard flight controller allows the end device Xbee to communicate with the base station through the coordinator Xbee.

An Arduino Uno is used as the hardware for the flight controller of the Quadcopter; onto which the flight controller software/code is run. The Xbee receiver module is plugged into the same arduino board using an Zigbee shield and communication is established with the coordinator Xbee which is connected to the base station/computer through a Zigbee explorer board.

One of the significant advantages with the Xbee communication mode of control is the possibility of developing a mesh network with multiple Xbee's. This allows for Quadcopters at various locations to be simultaneously controlled. This provides a platform for the control and monitoring of Quadcopters performing similar tasks through a single central server. The X-CTU software was installed in the base station PC for an easy interface with the Xbee modules.

Quadcopters do not require mechanical linkages to vary rotor blade pitch as they spin and the use of 4 rotors will allow them to have less kinetic energy during flight.

2. RELATED WORK

Recent years have seen a marked increase in the usage of Quadcopters as a surveillance device in diverse fields. A lot of the recent work involves developing a flight controller system and establishing a communication system interfaced with a computer system for further in-flight changes and control.

Work by L. Salih et al.[1] focuses on developing a PID flight controller code for UAVs. The various types of response provided by the P, I and D parts of a PID controller have been analysed.

The paper by Y.Li et al.[2] gives a detailed insight into the design of such a Quadcopter system. It takes into consideration the various dynamic factors to be considered in designing a stable system. Johnson et al.[3] have focused on a GUI interface that includes an Xbee communication system for transmission and reception of data. The GUI interface in this case helps in continuous monitoring and control during flight. We got a clear dynamic analysis of the structure of a Quadcopter system due A.K Cooke et al.[4] and C Balas et al.[5]. The role and importance of the translational and rotational axes of a Quadcopter frame are explained. The architectural design and dynamic implementation focuses on the Euler angle notation along the different axes.

Present work involves development of a Quadcopter system with an interface for monitoring sensors and other systems in a number of modern day technological devices. Rescue systems and atmospheric as well as agricultural monitoring systems are also largely working on incorporating Quadcopter systems.

3. METHODOLOGY

The project involved working on 4 different phases.

The first phase involved designing the Quadcopter structure for stable flight with the various components attached to it. The weight has to be carefully and equally distributed to ensure overall balance in the system during flight.
The second phase involved the establishment of a communication channel between the Quadcopter and the PC using the Xbee transmitter-receiver set up.
The third phase involved developing a PID and a PID rate control loop whereby information regarding the orientation of the Quadcopter from the MPU-6050 gyro-accelerometer sensor could be used by the flight control software for stabilizing the system when subject to disturbances or to respond to user commands.
The final phase involved testing the system response based on user input of upward, downward and sideward thrust. The control commands were provided through the X-CTU software installed on the PC. This stage was essentially an integration of the previous 3 stages and their corresponding tests.

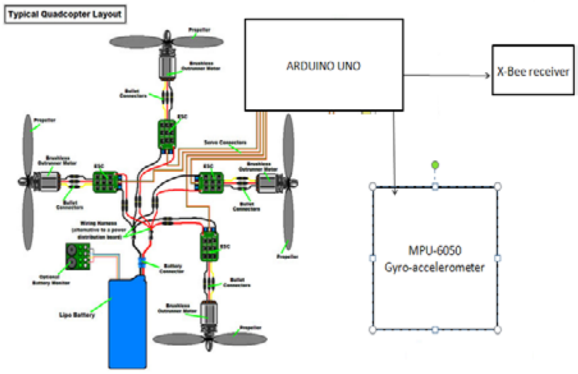

Fig 1: TRANSMITTER SIDE

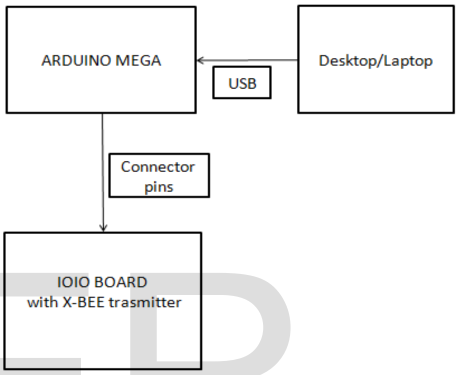

Fig 2:RECEIVER SIDE

4. DESIGN

The design of the Quadcopter involved selecting the appropriate hardware components which would work together to sustain stable flight of the Quadcopter; and on the software side involved designing an algorithm which would implement all the features to allow:
1. The hardware to run properly and
2. The proper execution of the control loop.
This has been achieved as follows

4.1 Hardware

The hardware design of the quadcopter required us to select the appropriate components with the following considerations:-

1.Sensors to be integrated/features

2.Overall weight of the Quadcopter

3.The dimensions of the propellers

4.The KV rating of the BLDC motor

5.Electronic Speed Control

6.The dimensions of the propellers

7.Battery Rating

8.Frame Structure

9.Wireless Communication

As our build was a frame-build with everything else built from scratch, we fixed our frame the first and chose a popular frame: the DJI f450 frame, as we wanted a sub 1kg build with the frame weighing in at 282 grams. We didn't want the battery to add to the weight, and thus compromising on the manoeuvrability, and therefore we chose a 2000MAh 3S 30C Lithium-Polymer(11.1V) battery as our power source to power both, the Arduino Uno and the BLDC motors (through the ESCs). The Battery weighed in at 180g and the Motors each at 53g and each ESC at 23g.

Weight Distribution

282 g	Frame
45 g	Arduino Uno + Zigbee Shield
4 x 38 g	Motors
4 x 23 g	ESC's
180 g	3S Battery
20 g	Sensors, Xbee, wires, tape, etc
771 g	**Total**

Table 4.1

Motors and Propellers

We define the position and velocity of the quadcopter in the inertial frame as:

$$x = (x, y, z) \quad \text{and}$$

$$x' = (x', y', z')$$

respectively. Similarly, we define the roll, pitch, and yaw angles in the body frame as:

$$q = (f, q, y)$$

with corresponding angular velocities

$$q' = (f', q', y').$$

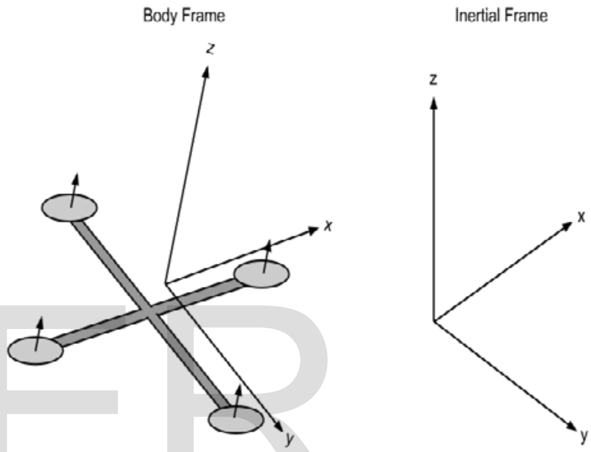

Fig 3: Reference frames

All motors on the quadcopter are identical, so we can analyze a single one without loss of generality. Adjacent propellers, however, are oriented opposite each other; if a propeller is spinning "clockwise", then the two adjacent one's will be spinning "counter-clockwise", so that torques are balanced if all propellers are spinning at the same rate.

Brushless motors are used for all quadcopter applications. The power generated by these motors is used to keep the quadcopter aloft.
Based on the set of equations given (reference: Andrew Gibiansky):
The energy the motor expends in a given time period is equal to the force generated on the propeller times the distance that the air it displaces moves (P.dt = F.dx). Equivalently, the power is equal to the thrust times the air velocity
(P = F.dx/d t).

$$P = T.vh$$

We assume vehicle speeds are low, so vh is the air velocity when hovering.

With the previously calculated weight estimate in mind, we chose the propellers and motors. We wanted a agile quadcopter to test the effect of using different control techniques on the flight and manoeuvrability of the quadcopter; and with the total thrust in rpm calculated to be close to 20000 for each motor, this required a motor of close to 1800K_v rating for each arm (1800 KV corresponds to 1800 rpm/volt giving an rpm of close to 20000 for 11.1V input). The corresponding suggested propellors were a set of 9x4.5 propellers.

4.2 Electrical Considerations

The Quadcopter is powered by four 1800K_v motors and requires at least 50% throttle to maintain a hover. With Each Motor having a maximum current draw of 14A, the total maximum current draw at full throttle would be 56A. A 2000Mah 30C LiPo battery would meet the requirement with 60A maximum current draw.

4.2.1 ESC's

While the output from the battery was DC, the input to a brushless DC motor is AC. This transition of current from DC to AC is carried out through programmable ESC's (Electronic Speed Controllers). These also control the speed of the motors. They consist of a series of switches for the same.

4.2.2 XBEE

The main reason behind choosing the Xbee for the Wireless communication was that unlike a traditional R/C 6/9 channel Transmitter, The Xbee could be programmed to transmit data both ways with equal ease. With this, we programmed the Arduino to receive both control and diagnostic commands from the base station/computer. As an initial stage we have stuck to using the Xbee in the AT mode, choosing to ignore the more advanced features offered by the 802.15 architecture. The AT mode performs parity checking and thus prevents corrupt commands and information from reaching either side, thus forming a major safety net to prevent corrupt information from affecting the quadcopter's flight.

The 802.15 network also allows only signals from the same PAN to be received and thus forming a minimal security feature from hacking, especially if the user randomized the PAN code at each start. Both the coordinator and the end node Xbee were initialized with the same PAN code and the same channel number. They were both preconfigured to communicate at 57000 baud allowing fast transfer of information. We were unable to reliably use the 115200 baud mode (see note in testing).

Among the different configurations of the flight controller software, in on of the diagnostic modes, we attempted to transfer sensor information through the Xbee network for every loop; this was unsuccessful because of the high bandwidth required for transferring all the sensor information with such a high update frequency. We then lowered the frequency to transmit the sensor information every one second and achieved valid data transfer.

4.3 Software Design

The Algorithm for the flight controller consists of a startup block and a control loop that is executed repeatedly to allow the quadcopter to maintain a steady hover.

4.3.1 Startup block

The startup block initializes all the sensors and provides a 10 second delay as a safety feature.

4.3.2 Control Loop

The control loop is divided into many portions

Read Sensor

Read Wireless Data

Calculate Throttle

Send throttle data to the ESC

4.3.3 PID Control Loop

Our algorithm uses two PID Control loops, a main PID loop and a rate PID loop. The input to the PID loop is the sensor data i.e.: fusion of accelerometer and gyroscope data passed; before being given to the PID loop, it is passed through a software implementations of a discrete complementary filter with alpha=0.98. This softens the spikes in the sensor readings. The dT value used for the PID loop is the time taken for each execution of the loop which in this case depends on the maximum frequency of the ESC - 50 hz.

The Rate PID loop uses the output of the main PID loop as its input and thus we have a cascaded PID control system with the inner PID loop's input as the angular velocity of the quadcopter's body in any one axis and the outer PID loop's input as the output of the inner loop. Although theoretically, the PID loops can be integrated into one loops as they follow linear equations, the splitting into two loops is for a practical reason as this allows the quadcopter's PID loop's constants to be easily

ISSN 2229-5518

calibrated using the observed response of the quadcopter's flight.

5. TESTING

A flaw observed after a number of test flights was the accumulation of garbage values as well as slightly adjusted offset values.
This though has pretty much been found to be a result of the slight displacement of the MPU-6050 sensor from its original position. This should be sorted out with a bit more strong and stable attachment of the sensor to the Quadcopter frame.

Another problem that arose while testing was that although the claimed maximum baud rate for the Xbee was 115200, the data would tend to get corrupted and using 57000 as the baud rate yielded stable operation.

6. RESULT

The degree of success of the project was based on:

1. The stability of the Quadcopter at particular heights with the corresponding thrust required.
2. The degree of control obtained based on directional inputs given in the X-CTU platform.
3. The responsiveness of the Quadcopter to changes in P, I and D values of the flight controller based on the type of flight required.

The above mentioned conditions were met satisfactorily within a range of 20 metres. This is keeping in mind the basic series of Xbee used for the project. Communication between an Xbee transmitter and receiver module of a higher series such as the Series 2 modules and the XSC modules have been established over much larger distances.
Furthermore, the quadcopter was found to maintain stable flight at a particular height when outside the range of the Xbee transmitter. As the battery on-board got depleted the Quadcopter was found to descend gradually.

7. CONCLUSION

The Xbee communication setup gives the option of operating multiple drones using a mesh setup. The Xbee modules basically operating at the same baud rate when configured receive the data transmitted to a particular module. Thus in the case of multiple drones performing the same function the required instruction can be given using a single transmitter (the Coordinator).

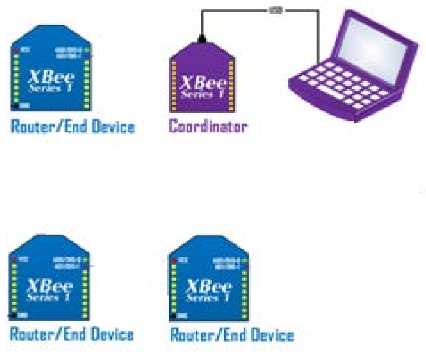

Fig 4: XBee user interface

The above is an illustration of a mesh network. The multiplicity of Xbee's controlled by the Coordinator interfaced with the PC can be increased with routers

8. FUTURE SCOPE

Thus taking into consideration the above mentioned options with the current setup, some of the possibilities include:

1. Links to Power and Data
   As a lot of structures are making use of sensors for timely data on their condition which need to be charged, a set of pre-programmed Quadcopters, when in position, will be able to recharge the sensors as well as derive information from them simultaneously. A regular check helps in the maintenance of these large structures.

2. Disaster Response
   Disaster management involves various phases based on the situation. A change from one phase to another can be incorporated into a whole set of Quadcopters through a mesh network using a signal transmitted from a central coordinator.

3. Monitoring systems
   Changes in the atmospheric parameters to changes in the condition of agricultural specimen can be monitored on a large scale basis through Quadcopters

REFERENCES

[1] L. Salih, M. Moghavvemil, H. A. F. Mohamed and K. S. Gaeid, "Flight PID Controller Design for a UAV Quad-copter," *Scientific Research and Essays*, Vol. 5, No. 23, 2010, pp. 3660-3667.

[2] J. Li and Y. T. Li, "Dynamic Analysis and PID Control for a Quadrotor," *Proceeding of the International Conference on Mechatronics and Automation* (*ICMA*), Bei-jing, 7-10 August 2011, pp.573-578.
[3] M. A. Johnson and M. H. Moradi, "PID Control," Springer- Verlag, London, 2005.
[4] A. K. Cooke, I. D. Cowling, S. D. Erbsloeh and J. F. Whidborne, "Low Cost System Design and Development an Autonomous Rotor Vehicle," *Proceeding of the* 22*nd International Conference on Unmanned Air Vehicle Systems*, Bristol, 16-18 April 2007, pp. 281-289 .
[5] C. Balas, "Modelling and Linear Control of Quadcopter"Thesis, Cranfield University, Cranfield, 2007.
[6] A. L. Salih, M. Moghavvemil, H. A. F. Mohamed and K. S. Gaeid, "Flight PID Controller Design for a UAV Quadcopter," *Scientific Research and Essays*, Vol. 5, No. 23, 2010, pp. 3660-3667.
[7] The MITRE Corporation. (2013). Designing a spatially aware, automated, quadcopter using an android control system. 2013-2014 Capstone Project Description
[8] Arducopter ---- Arduino----Based Autopilot For Mulirotor Craft, From Quadcopters To Traditional Helis ---- Google Project Hosting. Code.Google.Com.
[9] Ashfaq Ahmad Mian, Wang Daobo (2007).― "Nonlinear Flight Control Strategy for an Underactuated Quadrotor Aerial Robot‖ Networking, Sensing and Control", 2008. ICNC 2008. IEEE International Conference Sanya, China.
[10] "Quadcopter Dynamics, Simulation and Control" by Andrew Gibiansky.
andrew.gibiansky.com.
[11] Mechanical Design (2010, April 13) Http://Wyvernupenn.Blogspot.Ca/2010/04/Mechani cal-Design.Html.