International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 16th Aug 2017. Link :http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-7

Title: **GESTURE CONTROLLED QUADCOPTER**

Volume 06, Issue 07, Pages: 257 – 264.

Paper Authors [1]**SHAIK MOHAMMAD KHAJA, [2]DR.V.K.SUMAN, [3]DR. SYED AZEEMUDDIN**

[1]Shri Venkateshwara University, Gajraula, Dist. Amroha, U.P. India

[2]SVSU, Meerut, U.P.India

[3]IIITHyderabad, Hyderabad, A.P. India

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

GESTURE CONTROLLED QUADCOPTER

SHAIK MOHAMMAD KHAJA[1], DR.V.K.SUMAN[2], DR. SYED AZEEMUDDIN[3]
[1]Research Scholar, Shri Venkateshwara University, Gajraula, Dist. Amroha, U.P. India
[2]Professor & Dean, SVSU, Meerut, U.P.India
[3]Assistant Professor, IIITHyderabad, Hyderabad, A.P. India
[1]smkhaja@gmail.com, [2]suman.glbitm@yahoo.co.in, [3]azeemiiit@gmail.com

Abstract: In this paper we present an approach to control a Quad copter by merely using hand gestures. It helps us to propose a way to accomplish Human Machine Interface. The idea is to extract an old techniques of controlling a Quad copter using joysticks, computer controlled etc. with more intuitive technique, that is, by controlling the Quad copter with the help of hand gestures, which will make it more user friendly, as the user would not have to study or learn about different instructions to control different parameters of a Quad copter, because these parameters will be controlled and adjusted with the help of codes interfaced to the hand gesture inputs. Gesture control devices can be made with the help of Accelerometer Gyro combination (for knowing the gesture) and a microcontroller.We have to make a glove with accelerometer (Small IC chip) and gyroscope (again Chip) .Then read their data as we move our hands. Accelerometer would give us hand motion data and gyro gives angular data. After reading this data we have to calibrate it. Then altering gestures according to the required motion, we can control equivalent Left, Right, UP and Down command for Quad copter. This is done through Microcontroller program using Arduino.

Keywords: Quadcopter, Hand Gestures, Accelerometer, Gyroscope, Arduino, UAV.

INTRODUCTION: A quadcopter is an aerial vehicle that uses four rotors for lift, steering, and stabilization. Unlike other aerial vehicles, the quadcopter can achieve vertical flight in a more stable condition. The quadcopter is not affected by the torque issues that a helicopter experiences due to the main rotor. Furthermore, due to the quadcopter's cyclic design, it is easier to construct and maintain. As the technology becomes more advanced and more accessible to the public, many engineers and researchers have started designing and implementing quadcopters for different uses. Various groups such as the military, engineers, researchers, and hobbyists have been developing quadcopters to understand different technical areas. For example, quadcopters can be used for land survey and collecting data. This could range from searching for survival victims in a disaster area to checking the state of electrical power lines. Some quadcopters in production today can hold light payloads, such as food and medical supplies, and deliver them to areas where normal planes cannot reach. Many amateur radio operators have designed and built their own multi-copters. Universities, such as MIT, have been studying and doing research for the quadcopter over the past couple of years. The aim of this project was to build and program a quadcopter that can be controlled using Hand Gestures. A UAV quadcopter is an unmanned aerial vehicle with four rotating rotors used for lift and movement. It uses an electronic control system and electronic sensors to help stabilize itself. Quadcopter parts have been decreasing in price over the past couple of years due to technological advances. As a result more hobbyists, universities, and industries are taking advantage of this opportunity to design and develop applications for the quadcopter

Flight Control:

A quadcopter consists of four motors evenly distributed along the quadcopter frame as can be seen in figure 1 below. The circles represent the spinning rotors of the quadcopter and the arrows represent the rotation direction. Motors one and three rotate in a clockwise direction using pusher rotors. Motor two and four rotate in a counter-clockwise direction using puller rotors. Each motor produces a thrust and torque about the center of the quadcopter. Due to the opposite spinning directions of the motors, the net torque about the center of the quadcopter is ideally zero, producing zero angular acceleration. This eliminates the need for yaw stabilization. A vertical force is created by increasing the speed of all the motors by the same amount of throttle. As the vertical forces overcome the gravitational forces of the earth, the quadcopter begins to rise in altitude. Figure 2 shows the vertical movement of the quadcopter. As above, the circles represent the spinning rotors, the larger arrows represent the direction the rotors are spinning, and the black arrows represent the forces caused by the spinning rotors. Pitch is provided by increasing (or decreasing) the speed of the front or rear motors. This causes the quadcopter to turn along the x axis. The overall vertical thrust is the same as hovering due to the left and right motors; hence only pitch angle acceleration is changed. Figure 3 shows an example of pitch movement of a quadcopter. As the front motor slows down, the forces created by the corresponding rotor are less than the forces created by the back rotor. These forces are represented by the blue arrows. These forces cause the quadcopter to tip forward and this movement is represented by the red arrow.

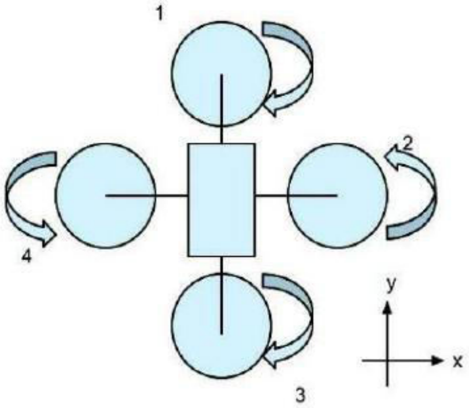


Fig 1 Quadcopter: Motor rotation directions

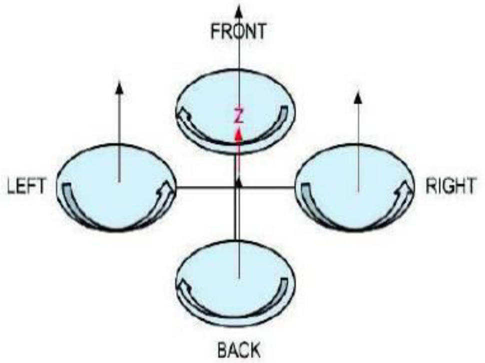


Fig 2 Quadcopter: vertical thrust movement

Roll is provided by increasing (or decreasing) the speed of the left rotor speed and right motors. This causes the quadcopter to turn along the y axis. The overall vertical thrust is the same as hovering due to the front and back motors; hence only roll angle acceleration is changed. Figure 4 shows an example of roll movement of a quadcopter. As the right motor slows down, the forces created by the corresponding rotor are less than the forces created by the left rotor. These forces are represented by the blue arrows. This causes the quadcopter to tip to the right and this movement is represented by the red arrow. Yaw is provided by increasing (or decreasing) the speed of the front and rear motors or by increasing (or decreasing) the speed of the left and right motors. This causes the quadcopter to turn along its vertical axis in the direction of the stronger spinning rotors. Figure 5 shows an example of yaw movement of a quadcopter. As the front and back motor slows down, the forces created by the corresponding rotors are less than the forces created by the left and right rotors. The quadcopter will begin to rotate in the same direction as the faster spinning rotors due to the difference in torque forces. This movement is represented by the red arrow.

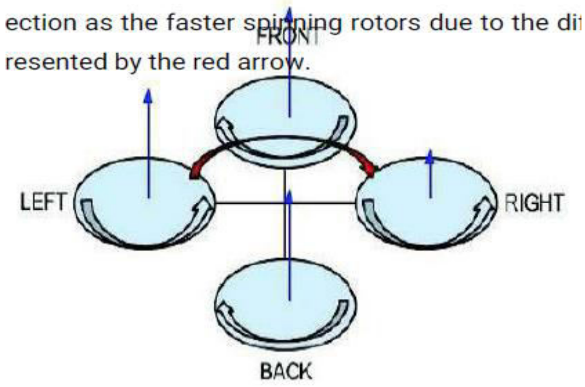

ection as the faster spinning rotors due to the di
resented by the red arrow.

Fig 3 Quadcopter: Pitch Movement

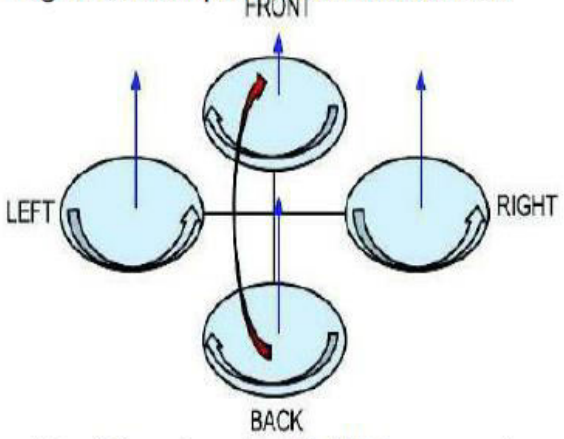


Fig 4 Quadcopter: Roll Movement

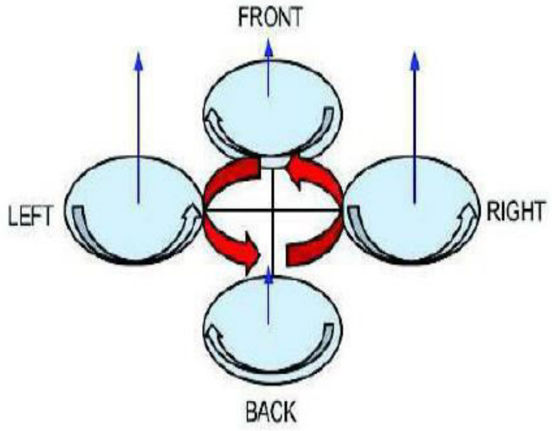


Fig 5 Quadcopter: Yaw Movement

MOTIVATION: The goal of our project is to design, implement, and test a stable flying UAV quadcopter controlled using hand gestures. Our plan was to choose an existing quadcopter kit and add the required components to give the quadcopter the capabilities that can be operated using hand movements. An IMU module will be used to determine the position of the hand and all these values are stored in EEPROM which can be sent using Bluetooth modules. The final quadcopter design had to meet the following specifications

1. The quadcopter must be capable of flying and landing in stable manner.
2. The quadcopter must be capable to act according to user's requirement.

The ultimate goal of this project is to design a UAV quadcopter that is able to operate by the hand movements of the user. This project could be used in many applications, ranging from checking the condition of power lines to searching for survival victims in a disaster area in a simple user-friendly way. The end result of this project could be used for these applications in future developments in this field. Due to the time constraints of our project, it was not possible to design and build a quadcopter from scratch. Therefore, it was decided that our group would purchase an existing quadcopter kit and interface the necessary components to fly, stabilize and log data. This project was split into three main stages; the research stage, the building and interfacing stage, and the programming stage. Each stage will be explained more thoroughly in the following sections of this document. Implementation: The research stage was a critical stage that provided our team with the knowledge necessary to complete the other stages of our project. This stage was an ongoing process that our team had to return to many times during the development process to gain the knowledge needed to continue on with the project. Our research encompassed a wide range of sources, which included studies done at different universities and hobby enthusiast sources. Our research included the aerodynamics of the quadcopter, theory and principle of each quadcopter component, and how the Arduino Software should be modified to fit the purposes of our project. This stage started when the ordered parts started arriving. During this stage we focused on verifying and testing each component thoroughly. The testing process will be explained in greater detail in a later section in this document. After each part was verified to be working correctly, we combined the components together. The frame and control board were assembled, the motors and electronic speed controllers (ESC) were mounted, the Arduino Mega board, and the transmitter and receiver were interfaced.

The Programming Stage:

In this stage of our project we were to design four different indications for our quadcopter. The preliminary designs for two gestures have been drawn out and will be explained in greater detail in a later section of this document. Unfortunately we did not get very far in this stage and programming for the other two gestures has yet to be started.

Arduino UNO:

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC -to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-toserial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. “Uno” means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform.

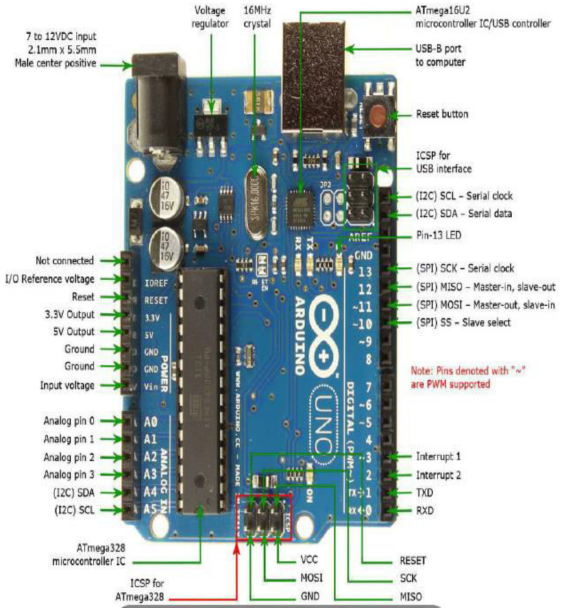


Fig 6 Arduino Uno

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a ACto- DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

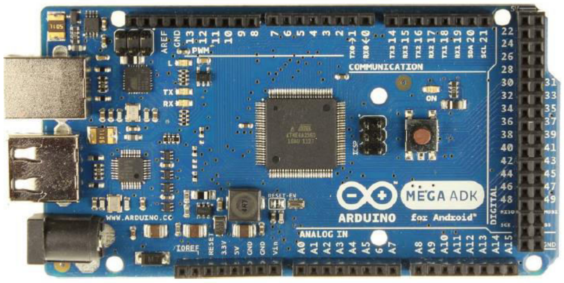
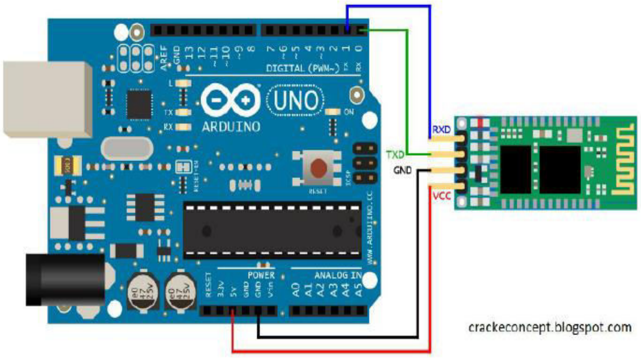


Fig 7 Arduino 2560

Communication

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Mega 2560's digital pins.The Mega 2560 also supports TWI and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library. Physical Characteristics and Shield Compatibility The maximum length and width of the Mega 2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins. The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Furthermore, the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega 2560 and Duemilanove / Diecimila boards. Please note that I2C is not located on the same pins on the Mega 2560 board (20 and 21) as the Duemilanove / Diecimila boards (analog inputs 4 and 5). Sending data from the Bluetooth module (via the module's TX pin) will not be an issue, as the Arduino's RX line will interpret the 3.3V signal from the Bluetooth module correctly. Receiving data from the Arduino is where we need to do more work. The RX line of the Arduino Uno is running at 5V, so we will be sending a higher voltage on the Bluetooth's module RX line, than suggested by the label. This may damage the Bluetooth module permanently!



crackeconcept.blogspot.com

Fig 8 Interfacing Bluetooth to Arduino

Setting And Pairing Master And Slave Modules

For this, we need to configure both modules. In order to do that we need to switch to AT Command Mode and here's how we will do that. First we need connect the Bluetooth module to the Arduino as shown in the below circuit schematics. What we need to do additionally is to connect the "EN" pin of the Bluetooth module to 5 volts and also switch the TX and RX pins at the Arduino Board. So the RX pin of the Arduino needs to be connected to the RX pin of the Bluetooth module, through the voltage divider, and the TX pin of the Arduino to the TX pin of the Bluetooth module. Now while holding the small button over the "EN" pin we need to power the module and that's how we will enter the command mode. If the Bluetooth module led is flashing every 2 seconds that means that we have successfully entered in the AT command mode. After this we need to upload an empty sketch to the Arduino but don't forget to disconnect the RX and TX lines while uploading. Then we need to run the Serial Monitor and there select "Both NL and CR", as well as, "38400 baud" rate which is the default baud rate of the Bluetooth module.

INTERFACING IMU TO ARDUINO

We'll be using an Arduino UNO here, but the code will work on a Mega or Leonardo just fine. Most other Arduino compatibles should have no problems either but we only support official Arduinos for code.

1. Connect the SCL pin on the breakout to the SCL pin on your Arduino. On an UNO & '328 based Arduino, this is also known as A5, on a Mega it is also known asdigital 21 and on a Leonardo/Micro, digital 3

2. Connect the SDA pin on the breakout to the SDA pin on your Arduino. On an UNO & '328 based Arduino, this is also known as A4, on a Mega it is also known asdigital 20 and on a Leonardo/Micro, digital 2

3. Connect the VIN pin on the breakout to 3.3V or 5V on your Uno (5V is preferred but if we have a 3V logic Arduino 3V is best)

4.Connect the GND pin on the breakout to the GND pin on your Uno

That's it! With those four wires, we should be able to talk to any of the I2C chips on the board and run any of the example sketches.

Interfacing MPU 6050 to Arduino Uno/Mega The MPU 6050 communicates with the Arduino through the I2C protocol. The MPU 6050 is connected to Arduino as shown in the following diagram. If your MPU 6050 module has a 5V pin, then we can connect it to your Arduino's 5V pin. If not, we will have to connect it to the 3.3V pin. Next, the GND of the Arduino is connected to the GND of the MPU 6050.

The program we will be running here, also takes advantage of the Arduino's interrupt pin. Connect your Arduino's digital pin 2 (interrupt pin 0) to the pin labeled as INT on the MPU 6050. Next, we need to set up the I2C lines. To do this, connect the pin labeled SDA on The MPU 6050 to the Arduino's analog pin 4 (SDA) and the pin labeled as SCL on the MPU 6050 to the Arduino's analog pin 5 (SCL). That's it; we have finished wiring up the
Arduino MPU 6050!

CONCLUSION:

In variety of computer application hand gestures provides an indigenous interaction between human interface and computer system. Vision based is the second way of collecting data. This paper presents the most advance technology to control the quad copter wirelessly. In the experiment done in the movement was controlled by hand as the hand was moved up the quadcopter gained height in proper relation. Using gesture controlled sensor and the modern quad copter technology coupled together the quad copter controlling technology changed its shape.

REFERENCES:

[1] McGowen, Stanley S. (2005):. Helicopters: an illustrated history of their impact. ABCCLIO,
Inc. Mellinger, D.

[2] Michael, N. and Kumar, V. (2010): Trajectory generation and control for precise aggressive maneuvers with quad rotors in Proceedings of the International Symposium

on Experimental Robotics Oka, K.

[3] Sato, Y. and Koike, H. (2002): Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems, In IEEE International Conference on
Automatic Face and Gesture Recognition.

[4] Magnussen Kjell E. Skjønhaug (2010): The quadcopter investigation. Technical report,
University of Agder.

[5] Rakshit, P.C.; Chattopadhyay, D. (2004): Elements of Physics, Vol. 1. New Age International Publisher. Sidenbladh, M.

[6] Black and D. Fleet (2000): Stochastic tracking of 3Dhuman figures using 2D image
motion. In Sixth European Conference on Computer Vision, pages II: 702–718, Dublin,
Ireland.

Mr. Shaik Mohammad Khaja has obtained his Bachelor’s degree in Electronics and Communication Engineering from JNTUHyderabad with distinction in the year 2001 and Master’s Degree in VLSI systems design in the year 2012 with distinction as well. He is a research scholar of Shri Venkateshwara University. He taught various subjects of ECE and CSE departments and his primary research areas are VLSI, Communications, Photonics and Nano-Electronics. He is actively involved in latest technologies like Hadoop, Cloud computing, TIBCO and SAP like technologies. He is a member of several bodies like ISTE, IETE, IEEE and VSI. He is state 5th rank holder in entrance for his Master’sdegree.