Outline

• quad-copter
• Abstract
• Quad-Copter Movement
• Hand movement
• Hardware Implementation
  • Quad-copter components
  • Wireless hand gesture components:

• Software Implementation
  • Quad-copter
  • Wireless hand gesture
• Testing
• Budget
• demo
The quad-copter is one of the most complex flying machines due to its versatility to perform many types of tasks. Classical quad-copters are usually equipped with four rotors. Quad-copters are symmetrical vehicles with four equally sized rotors at the end of four equal length rods.
Abstract

The objective of this project is to build a quad-copter that can be controlled by hand gesture wirelessly. User is able to control motions of the quad-copter in three dimension.
Quad-Copter Movement

Yaw Rotation

Each of the rotors on the quad-copter produces both thrust and torque. Given that the front-left and rear-right motors both rotate counter-clockwise and the other two rotate clockwise, the net aerodynamic torque will be zero.

Figure 1: Torque patterns and related motion.
Hovering

For hovering a balance of forces is needed. If we want the quad-copter to hover, \( \text{SUM}(F_i) \) must be equal \( m \cdot g \). To move the quad-copter climb/decline the speed of every motor is increased/decreased.

\[
\text{SUM}(F_i) > m \cdot g \iff \text{climb} \\
\text{SUM}(F_i) = m \cdot g \iff \text{hover} \\
\text{SUM}(F_i) < m \cdot g \iff \text{decline}
\]

Figure 8: Balance of power while hovering.
Tilting

Now let us take a look on what is happening when we tilt the quad-copter. For simplification only two of the four rotors are shown. We see that the force is divided in two different parts. FL1 and FL2 are the part of the force used to lift the quad-copter. FT1 and FT2 represents the part used for the translation. It is obvious that the lift part becomes smaller with increasing $\phi$.

Figure 9: Force distribution for tilting.
Hand movement

* control the roll of quad-copter: rotate the hand left and right

  • control the pitch: rotate the hand up and down
  • control the speed: fingers motion change the speed (throttle).

Figure 5: Movement of quad-copter and the way of control.
Hardware Implementation

- Accelerometer
- MCU (Arduino)
  - Analog: Flex Sensor 1, Flex Sensor 2
- RF Transmitter (433 MHz)
- RF Receiver (433 MHz)
- MCU (Arduino)
  - PWM (300 Hz)
    - ESC 4
    - ESC 3
    - ESC 2
    - ESC 1 electronic speed controller
- IMU Gyro & Acc
  - SCL, SDA
Hardware Implementation

Quad-copter components:
1- Frame.
2- Microcontroller (Arduino Uno).
3- Motors (A2217-9 Brushless Outrunner Motor).
4- Electronic Speed Controller (ESC).
5- Lithium Polymer Battery.
6- Propeller.
7- Inertial Measurement Unit (IMU Digital Combo Board).
8- RF receiver.

Wireless hand gesture components:
1- Microcontroller (Arduino Uno).
2- Accelerometer (ADXL 335).
3- Flex sensors.
4- RF transmitter.
Quad-copter components

Frame

12 cm

27 cm
Quad-copter components

Frame

* The first consideration is the material to be used. It must be lightweight, sturdy, and affordable. The forces which act on the quad-copter primarily will be gravity and air pressure.

* We chose plastic which is less weight from the other material.

* We designed a prototype frame with a 12cm X 12cm square plastic central plate with four rods 27cm.
Quad-copter components
Microcontroller
Quad-copter components

Microcontroller

* Collects sensor data
  ● Receives control commands
  ● Calculates orientation.
  ● Control motor speed.

We use Arduino which have the following specifications:

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Value</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>ATmega328</td>
</tr>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
<tr>
<td>Input Voltage (limits)</td>
<td>6-20V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>14 (of which 6 provide PWM output)</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>6</td>
</tr>
<tr>
<td>DC Current per I/O Pin</td>
<td>40 mA</td>
</tr>
<tr>
<td>DC Current for 3.3V Pin</td>
<td>50 mA</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>32 KB (ATmega328) of which 0.5 KB used by bootloader</td>
</tr>
<tr>
<td>SRAM</td>
<td>2 KB (ATmega328)</td>
</tr>
<tr>
<td>EEPROM</td>
<td>1 KB (ATmega328)</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>16 MHz</td>
</tr>
</tbody>
</table>
Quad-copter components

Motor
The motors chosen should meet the following specifications:

- Lightweight.
- High speed and torque.
- PWM speed controlled.

We chose BL-2217/9 brushless Outrunner motor

* Brushless
  - Outrunner
  - Requires special controller
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Kv</strong></td>
<td>950 RPM/V</td>
</tr>
<tr>
<td>Max Efficiency</td>
<td>80%</td>
</tr>
<tr>
<td>Max Efficiency Current</td>
<td>5 - 15A (&gt;75%)</td>
</tr>
<tr>
<td>No Load Current</td>
<td>0.9A @10V</td>
</tr>
<tr>
<td>Resistance</td>
<td>0.095 ohms</td>
</tr>
<tr>
<td>Max Current</td>
<td>18A for 60S</td>
</tr>
<tr>
<td>Max Watts</td>
<td>200W</td>
</tr>
<tr>
<td>Weight</td>
<td>73.4 g / 2.59 oz</td>
</tr>
<tr>
<td>Size</td>
<td>27.8 mm x 34 mm</td>
</tr>
<tr>
<td>Shaft Diameter</td>
<td>4mm</td>
</tr>
<tr>
<td>Poles</td>
<td>14</td>
</tr>
</tbody>
</table>
Quad-copter components

Electronic Speed Controller (ESC)
*Converts the battery pack DC voltage to a three phase alternating signal which is synchronized to the rotation of the rotor and applied to the armature windings.

*The motor speed is set by the ESC in response to a pulse width modulated control signal.

- The motor speed is then proportional to the root-mean-square (RMS) value of the armature voltage.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output</strong></td>
<td>Continuous 18A, Burst 22A up to 10 Secs</td>
</tr>
<tr>
<td><strong>Input Voltage</strong></td>
<td>2-4 cells lithium battery or 5-12 cells NiCd/NiMh battery</td>
</tr>
<tr>
<td><strong>BEC</strong></td>
<td>2A / 5V (Linear mode)</td>
</tr>
<tr>
<td><strong>Max Speed</strong></td>
<td>210,000rpm for 2 Poles BLM, 70,000rpm for 6 poles BLM, 35,000rpm for 12 poles Brushless Motors</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>45mm (L) * 24mm (W) * 11mm (H)</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>18g</td>
</tr>
</tbody>
</table>
Quad-copter components

Electronic Speed Controller (ESC)

**Signal output from MCU to ESC**
ESC handle (1-2 ms) pulse width but we use output signal frequency 300Hz not 500Hz.

**Signal output from ESC to motor**
The frequency of output signal from ESC to motors 10-30KHz.
Quad-copter components

Battery
Quad-copter components

Battery

- Lightweight
- High discharging current and capacity
- Low internal resistance.
  * Long working time.

We select Lithium Polymer (LiPo) to achieve these characteristics.

Max current can be calculated by using the following equation:
Max current = $Ah \times C = 5A \times 30 = 150A$

In average, all four motors consume 40A. We can calculate the flight period using the following equation:
Flight period = $Ah / Acc = 5A / 40A \times 60 = 7.5$ minutes

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>11.1V</td>
</tr>
<tr>
<td>Nominal capacity</td>
<td>5000 mAh</td>
</tr>
<tr>
<td>Continuous discharge current</td>
<td>30 C</td>
</tr>
<tr>
<td>Dimension</td>
<td>131<em>41</em>21 mm</td>
</tr>
</tbody>
</table>
Power Distribution
Quad-copter components

Propeller
Quad-copter components

Propeller

* Dimension: 10X4.7 inch
  ● 2 blades
  ● Directly attached to motor
  ● 2 each rotating CW and CCW (a "pusher" and a "puller").
  ● Propeller balance reduces vibrations.
Quad-copter components

Inertial Measurement Unit (IMU Digital Combo Board)
This is a simple breakout board for the ADXL345 accelerometer and the ITG-3200 gyro.

With this board, we get a full 6 degrees of freedom

* combination of accelerometers and gyroscopes is a common approach used to measure the stability of quad-copters.
The sensors communicate over I2C

**Gyro**

0x0D0 for write, 0x0D1 for read

**Accelerometer**

0x0A6 for write, 0x0A7 for read
Quad-copter components

Radio Frequency Receiver – 434 MHz
Quad-copter components

Radio Frequency Receiver - 434MHz

This wireless receiver provides a simple, straight-forward receiver for all of low-cost wireless project.

Features:
• 434 MHz.
• 150m range.
• 4800bps data rate.
Wireless hand Gesture Components
Wireless hand Gesture Components

- ADXL 345 X-Axis
- ADXL 345 Y-Axis
- Flex Sensor
- Flex Sensor

MCU

transmitter

gesture wireless
Triple Axis Accelerometer Breakout (ADXL345)
Flex sensor
When the sensor is bent, the conductive particles move farther apart, increasing this resistance.
RF Link Transmitter - 434MHz
RF Link Transmitter - 434MHz

This wireless transmitter, provides a simple, straight-forward transmitter for all of low-cost wireless project and work with the 434MHz receivers

Features:
• 434 MHz.
• 150m range.
• 4800bps data rate.
Quad-copter processes

To maintain the stability and response to control command from transmitter,
Flight control process:

1. Start
2. Read all current parameters
3. Calculate the output from PID controller
4. Change motors speed
5. End
Flight control process:

Our challenge is the combination both of the gyro and accelerometer values.

We used Kalman filter which is the most commonly approach to make combining of these sensors by filtering out noise from both sensors and derived angles for both in a range between -90 and 90 degrees.
Flight control process:

To keep quad-copter self-stable automatically it should use specific algorithm, the best algorithm for this task is PID controller.
Flight control process:

PID controller

The quad rotor will use a Proportional-Integral-Derivative control which a closed-loop feedback system, it will be tuned to determine the optimum response and settling time.

The controller calculated the difference between the desired orientation and the current orientation and adjusts output value(U) accordingly.

The equations for a PID controller is as follows:

\[ u = P + I + D \]

\[ e(t) = ed(t) - ea(t) \]

\[ P = Kp * e(t) \quad I = Ki \int_0^t e(t) \, dt \quad D = Kd \frac{de}{dt} \]
Change motor speed control

We use the correction value from PID, to change the motor speed through changing the duty cycle of PWM to each motor.

The next equations show how to change the motors speed.

\[
\begin{align*}
\text{motor[right]} &= \text{throttle} + \text{turn_roll} + \text{turn_pitch} \\
\text{motor[left]} &= \text{throttle} - \text{turn_roll} + \text{turn_pitch} \\
\text{motor[rear_right]} &= \text{throttle} + \text{turn_roll} - \text{turn_pitch} \\
\text{motor[rear_left]} &= \text{throttle} - \text{turn_roll} - \text{turn_pitch}
\end{align*}
\]
Gesture Wireless Processes

We use accelerometer and flex sensors to detect the correct hand motion. The bellow flowchart shows the sequence of process.
Testing

• Video
# Budget

<table>
<thead>
<tr>
<th>Component</th>
<th>Count</th>
<th>Price</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino</td>
<td>2</td>
<td>180</td>
<td>360</td>
</tr>
<tr>
<td>Motor</td>
<td>4</td>
<td>250</td>
<td>1000</td>
</tr>
<tr>
<td>ESC</td>
<td>4</td>
<td>100</td>
<td>400</td>
</tr>
<tr>
<td>Propeller</td>
<td>4</td>
<td>20</td>
<td>80</td>
</tr>
<tr>
<td>IMU</td>
<td>1</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>1</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>RF pair</td>
<td>1</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>Flex</td>
<td>2</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Battery</td>
<td>1</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Total (NIS)</td>
<td></td>
<td></td>
<td>2830</td>
</tr>
</tbody>
</table>
• DIMO