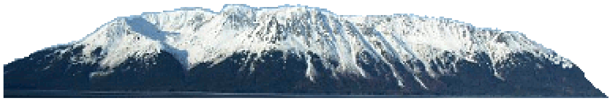
Terms of service



Brokking .net

Let's keep it simple

Home

TL-blog

Projects

Articles

Donation

YouTube

LinkedIn

Project YMFC-AL - The Arduino auto-level quadcopter

Home Q & A Downloads Media

The YMFC-AL is an auto-leveling Arduino Uno based quadcopter that is easy to build and fun to fly. Auto leveling means that, when you release the sticks the quadcopter levels itself.

The total cost to build a quadcopter like this is approximately $150 if you use the hardware list below. This includes a battery, transmitter, charger, etc.

The YMFC-AL is not a high level flight controller. The purpose of the YMFC-AL is to provide a simple and understandable code that is needed to build an Arduino based auto-level quadcopter flight controller.

The code is well commented and clearly explained in the ↗ YMFC video tutorials. This makes it possible to further develop the YMFC-AL flight controller code for your own purpose.

If you encounter any problems during the build or setup please check the ↗ Q&A page first. Most questions are already answered in detail.

Step 1 - Software

First download the complete YMFC-AL software package:

↗ YMFC-AL.zip (version 1.4 April 30, 2017)

And read the README.txt file that comes with the software package.

Step 2 - Hardware

I often get the question what hardware is needed to build the YMFC-AL quadcopter. I made the following hardware list for your convenience. This list is a suggestion and it's your own responsibility to ensure that the products meet your specific requirements. But this list should be sufficient to build the YMFC-AL quadcopter.

- ↗ 1 x 450 size frame with integrated power distribution board
- ↗ 4 x 1000kV motor / 10x4.5 props / ESC combo
- ↗ 1 x 3S / 2200mAh / 30C lipo
- ↗ 1 x Arduino Uno or Funduino Uno
- ↗ 1 x MPU-6050 gyro / accelerometer
- ↗ 1 x Flysky FS-T6 6-CH TX Transmitter
- ↗ 1 x 2S/3S lipo battery charger

And some small parts like three resistors (1.5kΩ & 1kΩ & 330Ω), a 1A diode (1N4001 or similar), LED, some wire, a connector for the flight battery, etc.

I personally prefer (and use) 8x4.5 inch props instead of the 10x4.5 inch props. This to offload the motors and ESC's and to get a better response.

Step 3 - The build

Watch the ↗ YMFC-3D part 1 (hardware) video and build the quadcopter according to the video and the schematic that is included in the complete software package.

▶ Building the YMFC quadcopter.

Detailed pictures of my own YMFC-AL quad can be found in the ↗ media section of this project page.

3.1 The diode D1 and resistors R2 / R3

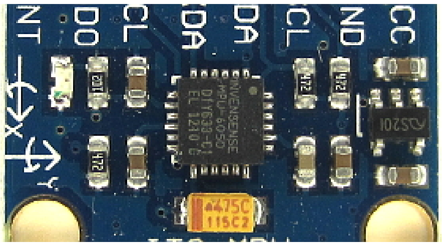
Click to see the full image

The diode D1 protects the USB port of the computer when the Arduino is connected to the computer. This diode has an important function and cannot be excluded.

The resistors divide the flight battery voltage by 2.5. This way it is possible to measure the battery voltage during flight. The LED will light up when the battery voltage gets to low and the motor rpm automatically increase to compensate the dropping battery voltage during flight.

The 1kΩ and 1.5kΩ resistors need to be installed correctly otherwise the quadcopter will not fly perfect.

3.2 The MPU-6050 gyro/accelerometer



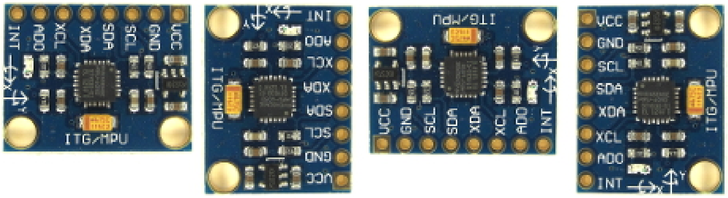
Click to see the full image

The only gyro / accelerometer that is supported by the YMFC-AL software is the MPU-6050. This is because the auto-level feature requires an accelerometer and a gyro as I explain in these two videos:

▶ MPU-6050 6dof IMU tutorial for auto-leveling quadcopters - Part 1

▶ MPU-6050 6dof IMU tutorial for auto-leveling quadcopters - Part 2

The orientation of the gyro is not important as long as the Z-axis is vertical (perpendicular to the surface) and the edges of the gyro are aligned with the edges of the quadcopter. Click on the image to see all the possible gyro orientations.



Click to see the full image

The setup software will detect the gyro's orientation and invert the gyro and accelerometer axis when necessary. Mount the gyro with thin double side tape. Don't use foam tape or other dampening material. This will decrease the performance.

3.3 The tranmitter and receiver

Click to see the full image

Almost every 4 channel RC transmitter can be used for the YMFC-AL. The most important feature is the used receiver output pulse. The range should be approximately 1000 till 2000 with a 1500 center position.

Check the manual of the specific transmitter / receiver for details. For example, the following transmitter should work fine with the YMFC-AL:

↗ 1 x Flysky FS-T6 6-CH TX Transmitter

Connect the roll (aileron), pitch (elevator), yaw (rudder) and throttle output of the receiver to the Arduino Uno ports 8, 9, 10 and 11. The order is not important as the setup software will recognize each separate channel. Check the manual of the transmitter / receiver to see which receiver port is connected to the specific function.

The receiver is powered by the +5V output of the Arduino. The connection can be found on the schematic (top left corner 'Receiver power').

If you want to know how the Arduino reads the receiver signals, watch this video:

▶ YMFC-3D part 2 - Connect RC transmitter and receiver - Arduino quadcopter.

3.4 The ESC's

Click to see the full image

On the schematic only the ground and the signal wires of the ESC's are connected. This is correct. The +5V from the ESC is not connected because the Arduino gets its power directly from the flight battery via the diode D1.

In some cases the ground of the ESC doesn't have to be connected. Check with a multimeter if the ground of the battery connection is connected to the ground / - of the esc connection wire. If these are connected the ground of the ESC does not need to be connected to the Arduino because they share the same battery ground.

The signal wire of the ESC's are connected to the digital outputs 4, 5, 6 and 7 of the Arduino as shown in the table below. Also check the direction of rotation.

Arduino	Location	Direction of rotation
D4	right front	counter clockwise
D5	right rear	clockwise
D6	left rear	counter clockwise
D7	left front	clockwise

If you want to know more about the workings of an ESC and how it is controlled by the Arduino, watch this video:

▶ YMFC-3D part 4 - Arduino quadcopter - Electronic Speed Controller (ESC).

Step 4 - Run the setup software

Remove the props, don't connect the flight battery and upload the setup program to the Arduino Uno. Open the serial monitor at 57600baud and complete the setup by executing the requested actions.

After the setup is completed all the settings are stored in the EEPROM of the Arduino.

Check question 2 on the ↗ Q&A page if you encounter any problems / errors during the setup.

Step 5 - Receiver and gyro check

To make sure that everything is working correct it's necessary to run some basic checks. Remove the props, disconnect the flight battery and upload the ESC calibration program to the Arduino. Open the serial monitor at 57600baud.

5.1 Receiver input check

Send the letter 'r' to start the receiver monitor. Now move the sticks and see if the values on the screen correspond with the movements of the sticks.

All the channels should read 1000us till 2000us with a center position of 1500 (+/-8).

5.2 Gyro / accelerometer angle check

After the receiver check is completed send the letter 'a' to start the angle check.

Don't move the quadcopter because the gyro needs to calibrate itself. After the calibration the roll and pitch angles are shown. The yaw value is the output of the gyro and will go back to zero if the yaw rotation stops.

Check if the angles correspond with the movement of the quadcopter:

- Nose up is positive pitch and nose down is negative pitch.
- Left wing up is positive roll and left wing down is negative roll.
- Nose right is positive yaw and nose left is negative yaw.

Step 6 - Calibrate the ESC's

Electronic speed controllers or ESC's for short are controlled with a 1000us till 2000us pulse. 1000us means off and 2000us means full throttle. To make sure that all the ESC's react the same way it's important to calibrate the 1000us and 2000us point. Without calibration the motors will perform different and the quadcopter doesnt fly well or might even crash.

Remove the props and upload the ESC calibration program to the Arduino. Disconnect the USB cable and follow the instructions in the manual to calibrate the ESC's.

In most cases this is done with the following steps:

1. Place the throttle stick in the upper position (full throttle)

2. Connect the flight battery
3. After some beeps place the throttle stick in the lowest position
4. Disconnect the flight battery

But again, check the manual of your specific ESC for the correct calibration procedure.

Step 7 - balance the motors and props

SAFETY NOTE:
This is the first time that the motors will run with the mounted props. Make sure that a save operation is possible. YOU ARE SOLELY RESPONSIBLE FOR YOUR OWN SAFETY.

7.1 Why is balancing the props important?

Balancing the props is incredibly important! Without well balanced props and motors the gyro and accelerometer will produce noise that makes the motors react jerky. There is minimal stability and the quadcopter can't level itself.

To get the best performance the props and motors need to be balanced perfectly. Putting the gyro / accelerometer on vibration dampeners does not help and can only make things worse.

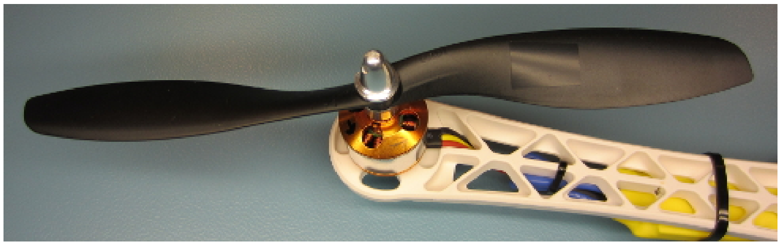
7.2 How to balance the props by using the YMFC-AL software

Mount the props on the motors and check if the counter clock wise and clock wise props are in the right position. Upload the ESC calibration program and open the Arduino serial monitor at 57600baud. Send '1' via the serial monitor and wait for the response "Test motor 1 (right front CCW.)".

The numbers that are printed on the screen represents the amount of vibration measured by the accelerometer. This is not a standardized value and should only be used to minimize the amount of vibration of your YMFC-AL quadcopter.

Hold the quadcopter firmly down, place the throttle in the lowest position and connect the flight battery. Now slowly increase the throttle until motor 1 starts to spin. Check the direction of rotation and that the prop produces upward thrust. If the motor rotates in the wrong direction you need to switch two of the three motor wires. Put the throttle in the lowest position to stop the motor.

Now hold the motor frame firmly in your hand and increase the throttle to half throttle. Check the numbers on the screen and also memorize the vibrations that you feel with you hand that is holding the motor frame.


Click to see the full image

Stop the motor and put a small piece of tape on one of the blades and run the test again. Check if the vibrations reduce. If nor try a piece of tape on the other blade. Keep doing this until the motor and prop run as smooth as possible. This can sometimes be a daunting task but the reward is a very stable flying quadcopter. So take your time and get it perfect!

When done with motor 1, send a '2' via the Arduino IDE and start the process again for motor / prop 2. And after that, send a '3' for motor number 3 and a '4' for motor 4.

By sending a '5' all the motors will run together as a final test.

Step 8 - Upload the flight controller software

Disconnect the flight battery and upload the flight controller software to the Arduino. Disconnect the USB cable and connect the flight battery.

Hold the quadcopter firmly in your hand and start the motors with the following sequence:

Start = throttle down and yaw left
Stop = throttle down and yaw right

Increase the throttle up to the point when it almost starts to become weightless. The quadcopter should now try to level itself. If you move the quad it should start to counteract the movement until it is level again.

When the roll or pitch stick of the transmitter is moved the quadcopter should move in the same direction. If this is not the case redo the setup procedure and double check all the points on this page.

When all is good it is time for a careful first test flight. Always fly the quadcopter outdoor and over grass for safety. Grass will minimize the damage during a crash.

Step 7

If necessary improve the standard PID settings. Watch the following video if you want to learn more about PID controllers:

▶ YMFC-3D part 5 - Quadcopter PID controller and PID tuning.

Step 8

If you would like to learn more about the technical background of the software you might want to watch the other ↗ YMFC video tutorials.