Engineering 100-Quadcopter Lab 1: Introduction to Quadcopters

This lab gives you an introduction to the various hardware and software components in a multirotor and explains how to prepare it for flight. We will

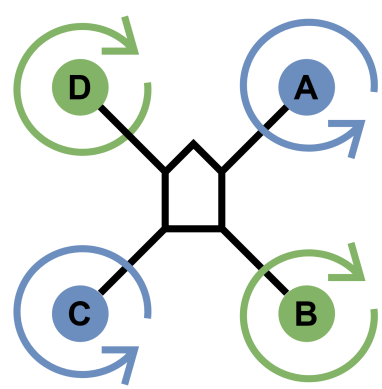
- describe the mechanics of multirotors and quadcopters in particular,
- familiarize you with safety procedures,
- introduce the ArduPilot open-source autopilot software,
- show you how to connect the quadcopter using Mission Planner, the ground station software, and
- explain how to set up and test the quadcopter and its components to avoid crashes and fires due to faulty parts.

1 Background Material

We will first introduce multirotors and their principal axes: pitch, roll, and yaw. We will also explain the hardware and software used for flight control.

1.1 Quadcopter

A multirotor, or a multicopter, is a rotorcraft that uses more than two rotors. An advantage of multirotors is that they require simpler mechanics for flight control, compared to helicopters. Multirotors with four rotors are called quadcopters (QCs). The diagram on the right illustrates the configuration of the QC that you will be using in this lab. The frame of the vehicle forms an X, with electronic components mounted near the center. Motors A and C spin counter-clockwise, while B and D rotate clockwise. Note that clockwise and counter-clockwise motors also use different types of propellers, as each propeller should push the air downward when rotating in its intended direction.



Rotors in a quadcopter.

The orientation of a QC is described in three dimensions: pitch, roll, and yaw. These axes are shown in Figure 1.

- **Pitch**: Rotating around this axis enables forward-backward movement, by tilting the nose up or down. A 0-degree pitch coincides with a level QC.
- **Roll**: This allows sideways movement by rolling and keeping the quadcopter at a fixed angle. Like pitch, a 0-degree roll also results in a level aircraft.

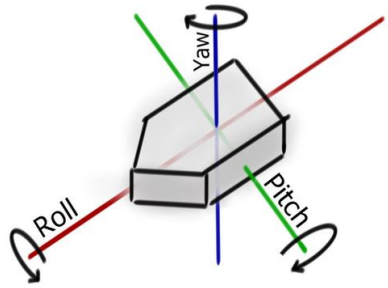


Figure 1: Aircraft principal axes. The triangle marks the nose of the aircraft. Image taken from Instructables.

- **Yaw**: This denotes rotation around a third orthogonal axis, allowing the QC to turn its nose left or right.
- **Throttle**: In addition to the three principal axes, the absolute speed of the motors, i.e., throttle, allows the QC to ascend or descend.

The pitch, roll and yaw of a QC are controlled by changing the relative speeds of the four motors, while the throttle is controlled by changing their speeds in unison. The main board, or the flight controller, is responsible for setting the speeds of the motors, according to control algorithms, sensor readings, and input from the pilot. In this course, you will use a BeagleBone Blue board for flight control. This board has a built-in Inertial Measurement Unit (IMU) and a barometer, which are used for keeping the QC level and controlling the throttle. It generates the PWM signals (described below) sent to the ESCs (Electronic Speed Controllers), which in turn control the motor speeds. More on ESCs later.

There is also an external, ultrasound sensor pointing down that we will use in the future. Its main purpose is to provide an accurate estimate altitude.

1.2 Lithium Polymer (LiPo) battery

The QC is powered by two LiPo battery packs. A 3-cell LiPo pack powers the motors, and a 2-cell pack powers the main board, i.e., the BeagleBone Blue. When working with the QC, **you should always disconnect the motor battery unless you are ready to test the motors or fly the QC**. When you do not intend to fly the QC, e.g., for the exercises in this lab, you can power your main board using a DC adapter, which also charges your 2-cell battery.

Each LiPo cell has a nominal voltage of 3.7 V. At full charge, the pack can sustain about 15 minutes of flight. These batteries are damaged when over-discharged, and can be a fire hazard if one later attempts to charge them. There is over-discharging protection built into the BeagleBone Blue board for the 2-cell battery, but not for the 3-cell battery. Therefore, you should periodically check the voltage of your 3-cell battery to make sure it is not over-discharged.

Each team is given a LiPo battery voltage tester. You should use the tester each time you start working on the QC, to avoid over-discharging the battery. At a starting point of full charge the team is advised to test the voltage again after 10 minutes of flying time, to make sure the voltage

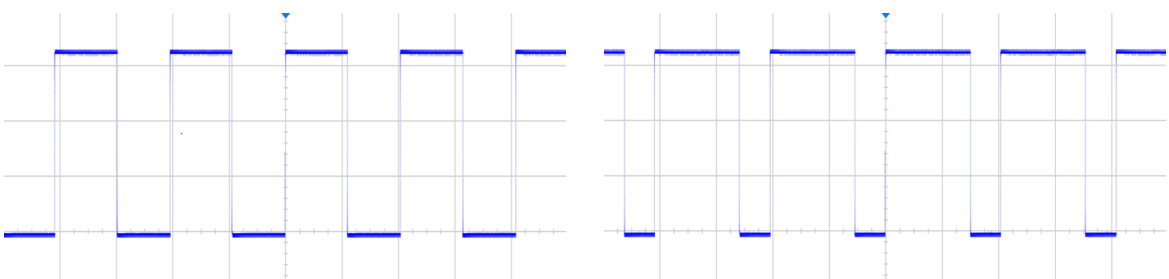
2-cell (left), 3-cell (center) LiPo batteries, and correctly connected battery tester.

stays above 3 V. Once it falls below 3 V, remove the battery pack and hand it to the lab staff for re-charging. As shown in Figure **??**, the black wire should be aligned with the ground (-) pin of the tester.

1.3 DC motors and Electronic Speed Controllers (ESC)

The 3-cell LiPo battery pack powering the motors is connected to a set of Electronic Speed Controllers (ESCs). Each motor receives speed control signals from a dedicated ESC. Each ESC takes Pulse Width Modulated (PWM) signals from the main board, and translates (decodes) that signal to a specific speed value. It then drives the motor at the appropriate speed.

Below are two examples of PWM signals, which you will be asked to observe and control in this lab, to drive the motors in your QC. A PWM signal is a periodic square wave, where the width of the pulse in each cycle can be controlled. They allow the controller to express a certain percentage value using a digital signal, which in turn controls the speeds of the motors.

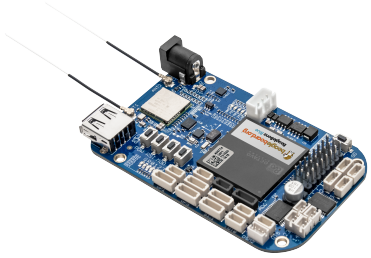


A 50% PWM signal (left) and a 75% PWM signal (right).

The figure on the left has a 50% duty cycle, meaning that it is on half the time and off the other half. This signal typically corresponds to 0% throttle (with respect to the maximum speed of the motor, i.e., full throttle). The throttle increases linearly, as we increase the on-time proportion of the PWM signal. A 100% PWM signal (i.e., a signal that is always on) usually indicates full throttle (max speed). The figure on the right is a 75% PWM, which reduces the off-time of the 50% PWM to 25%. Due to the linear relationship between the PWM percentage and throttle, this signal indicates a 50% throttle. Note that one generally has to calibrate an ESC, so that it knows the percentage values corresponding to 0% and 100% throttle. The ESCs used in this lab have been pre-calibrated.

1.4 Flight controller (BeagleBone Blue)

This robotics computer board includes an ARM Cortex-A8 microprocessor and several sensors, including a barometer and a 9-axis Inertial Measurement Unit (IMU), that combines an accelerometer, a gyroscope, and a magnetometer. It can also output PWM signals for controlling ESCs/motors. It is a full-featured computer running Linux that is accessible via secure shell and runs a web server. It can also interface with additional sensors and actuators. Something like this used to cover a desk a few years ago, it is now the size of your palm.



The BeagleBone Blue

You can connect to and control this board using a micro-USB cable (connector is mounted below the board, just under the DC barrel connector at the top of the figure). A USB connection can both power the board and allow you to control it from your computer. Alternatively, you can also power the board using a 2-cell LiPo battery, or a 12V DC adapter (which can also charge a battery if it is present). When powering the board using a battery or from a power outlet, you can also connect to it using its built-in WiFi network.

The BeagleBone has 4 GB of on-board flash storage for storing its firmware (i.e., a Linux image). Alternatively it can also boot from an external microSD card. In this course you will use a microSD card for storing the firmware, so you can easily transfer your software to a new quadcopter.

Note that on your quadcopter, the board is mounted such that the USB and DC connectors are facing forward.

1.5 Flight control software (ArduPilot and Mission Planner)

The software we use to control flight are (1) ArduPilot[1], which has been pre-installed on the microSD cards that we will provide to you in the lab, and (2) a ground control station such as Mission Planner, which must be installed on your laptop. You will need to connect to ArduPilot from Mission Planner using a WiFi connection to the BeagleBone. From Mission Planner you can then visualize and configure various parameters of the QC. During flight, ArduPilot directly controls the motors from the QC, using input from you, and readings from its sensors.

1.6 Summary of safety measures

There are a number of safety measures that you must follow when working with quadcopters and other accessories:

- Always wear safety goggles when the propellers are attached and the motor battery is attached.
- Disconnect the LiPo battery pack when its voltage falls below 3 V and hand it to the lab staff.
- Do not charge LiPo battery packs on your own.
- Propellers will not be used for this lab.
- Do not touch the main board when power is on. This destroyed several boards during lab development. It is a major problem.
- Never lift the QC by the circuit board. Only lift it by the frame.

[1] `http://ardupilot.org/ardupilot`

- When motors are spinning, even without propellers on, there is always the chance of something loose being caught in it, such like a strand of hair, loose strings from clothing, etc. Be sure that any such thing is securely tucked away before you turn the motors on.

2 Pre-Lab (30 Points)

Please complete this part individually on a separate sheet of paper, and submit it online on Canvas. Pre-labs are due the day before the lab for your lab section.

Question 2.1. (6 points) In three sentences or fewer, explain why two of the motors spin in the opposite direction of the other two?

Question 2.2. (6 points) In three sentences or fewer, explain what a PWM signal is and why it is useful.

Question 2.3. (8 points) How can the quadcopter control motor outputs to allow the vehicle to spin (i.e., change yaw), while keeping it level and hovering? Note that, if all the rotors spun in the same direction, this would result in a net non-zero angular momentum for the quadcopter: it would continuously yaw. Provide an example of this using throttle percentages for different motors that would result in clockwise rotation. Assume that a throttle of 50% on all motors keeps the vehicle level and hovering at a constant altitude, and use the first picture in the lab with motors labelled A, B, C, and D to answer this question.

Question 2.4. (10 points) Assume that an ESC is calibrated to map 55% and 95% PWM to 0% and 100% throttle, and that PWM signal sent to it has a period of 2 ms. What is the duty cycle of a PWM signal corresponding to 75% throttle? What is the duration (in milliseconds) of the on-portion of this signal in each period?

3 In-Lab (40 Points)

The lab consists of four parts. Before starting, make sure you have received every item in the list below. Before starting, besure to disconnect the motor battery.

- A quadcopter, including the frame, ESCs, motors, the main board, etc.
- A 2-cell (for the main board), and a 3-cell (for the motors) battery pack.
- A DC adapter for powering the main board, and charging its battery.
- A voltage tester, for measuring the voltage of LiPo batteries.
- A microSD card containing Linux and ArduPilot for the BeagleBone Blue.

For this lab you need to form groups of four. Once you reach a checkpoint, ask your instructor to sign it off for you before continuing. The in-lab must be completed as a group, and submitted by one team member online on Canvas. Note the due date on Canvas; you have one week following the lab to complete the in-lab and post-lab assignments.

3.1 Install a ground control station

In this and subsequent labs, you will be using a program known as a ground control station in order to control your quadcopter. The most well-supported ground control station program is Mission

Planner. Mission Planner is only available on Windows computers, although you can run Mission Planner from a virtual Windows environment on a Mac or Linux machine. There is also the option of dual-booting your computer with Windows. The lab staff can assist you with this as well, although it may be more trouble than it is worth.

If you are in a group of students that only have computers running Mac or Linux, then ask the lab staff to see if you can use one of the loaner laptops for this lab. If even one student has a Windows computer in your group, then please use that machine and install Mission Planner for this lab and for future labs. The download for Mission Planner[2] is available at the link in the footnote or from the files on Canvas. NOTE: Make sure you download version 1.3.58 (MissionPlanner-1.3.58.msi file). More recent versions are incompatible with our current implementation.

3.2 Safety instructions

Make sure you have gone through the list of safety measures. Use the voltage tester for the 3-cell battery pack. When using the voltage tester, align the (negative sign) ground pin with the black ground wire.

Note that the BeagleBone uses LEDs to show the voltage level of your battery, so you can monitor the board battery without having to disconnect the main board. Look on your board for the LEDs next to the battery connector to determine the voltage.

Question 3.1. (2 points) What is the reading on the tester at the beginning of the lab for the 3-cell battery?

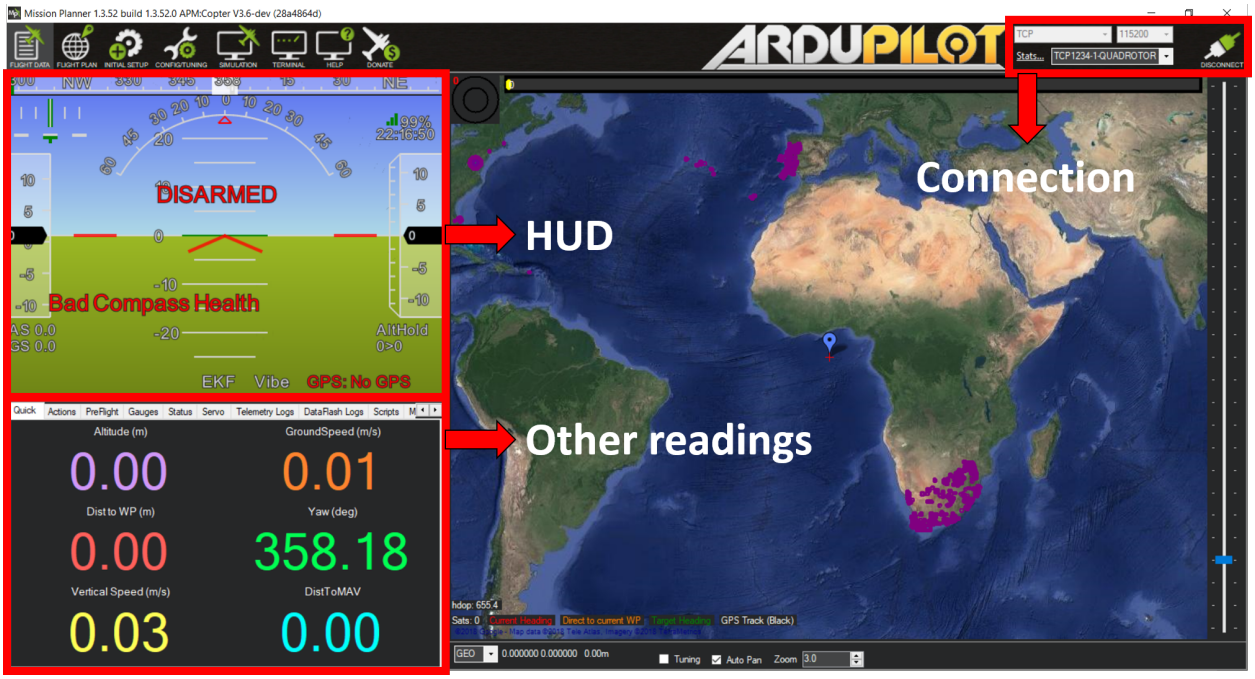
Question 3.2. (2 points) At what voltage should you replace the battery pack?

Question 3.3. (2 points) What is the reading on the tester at the end of the lab for the 3-cell battery?

3.3 Connecting to the quadcopter

Insert the microSD card into the BeagleBone, the microSD slot is located under the board near the USB and DC connectors. Then power on the main board by connecting the smaller 2-cell battery pack. The barrel connector should plug directly into the board.

[2] `http://firmware.ardupilot.org/Tools/MissionPlanner/archive/`



Mission Planner ground control station.

Once the board is on, connect your computer to it by connecting to the board's WiFi. The name and password for the WiFi network are written on a sticker on your microSD adapter.

The board has been pre-configured to run the ArduPilot flight control software once it is powered on. Run Mission Planner (the ground controller) from your computer. On the top-right corner of the window, select `TCP` from the drop-down menu and click `Connect`. The software asks you for a host name (enter `192.168.7.2`) and a port number (enter `1234`). Wait for the software to connect to the main board, and read its parameters. Ignore any warnings that state `Bad Compass Health` (the on-board compass has been disable since it cannot work reliably indoors), or `GPS: No GPS` (the BeagleBone does not have a GPS chip).

Make sure you are on the `Flight Data` tab in the top menu. You can now check connectivity by lifting, moving, and tilting the QC; if properly connected you should see the readings change on Mission Planner's display panel. Use the Heads Up Area (HUD) to find the nose of the aircraft, and the pitch, roll, and yaw axes. Find the corresponding readings for these axes in Mission Planner. Make sure you have selected the `Quick` tab under the flight data screen. You can also see other readings from the QC on your screen.

Checkpoint 1. (2 points) Ask your instructor to double check your connection to the quadcopter.

Question 3.4. (3 points) Sit the QC level on the lab bench. What are the readings you see in Mission Planner? Please record the pitch, roll, and yaw.

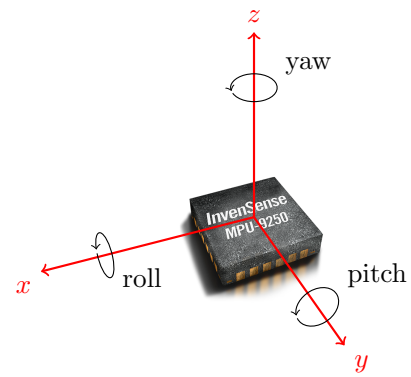
Question 3.5. (2 points) Record the measured ranges (minimum and maximum) for pitch and roll in degrees.

Question 3.6. (4 points) Now try to change the pitch, roll, and yaw of the QC one at a time (e.g., put the QC nose down, or on its left side). Record what you did and write the parameter values on Mission Planner for two different orientations below.

3.4 Setting up and calibrating the quadcopter

In order for the ArduPilot software to know how to control the quadcopter, it first needs to know the specifications of the aircraft, for instance the number of rotors, and the presence of optional accessories such as a downward facing ultrasonic sensor. The parameters have been provided to you in the `QC.param` file in the lab 2 files from Canvas. In order to load these parameters, go to `Config/Tuning` → `Full Parameter List` and click `Load from file`. After selecting and loading the parameter file, click `Write Params` for them to be written on the main board.

The QC uses a 9-axis Inertial Measurement Unit (IMU), consisting of a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. The magnetometer (i.e., compass) is not reliable indoors, and is therefore disabled. The accelerometer is used determine the aircraft's acceleration along the three major axes. The gyroscope measures the angular velocity of the vehicle. Note that in free-fall, the accelerometer would read zero; when stationary, it registers $9.8\,\mathrm{m/s^2}$ directly down due to gravity. By looking at how the gyroscope and accelerometer readings change during flight, the software can maintain a real-time estimate of pitch, roll, yaw and the acceleration of the aircraft along the X, Y, and Z axes.



You will need to calibrate the accelerometer before flying the quadcopter. Note, the accelerometer includes the gravitational acceleration in its reading. The reason for this is that the flight control software tries to keep the QC level according to the IMU readings. If there is a slight difference between the pitch and roll of the board and that of the body, this results in the QC drifting, even when the software thinks that it is level. You can prevent this by letting the software know what orientation corresponds to a *level* body, hence the need for calibration.

Go to `Initial Setup` → `Mandatory Hardware` → `Accel calibration`, and click `Calibrate Accel` to start calibrating the accelerometer.

- Place the vehicle on a level surface and press any key to continue.
- Now follow the instructions on your screen and press any key to move on to the next step. You will need to place the vehicle on each edge, namely, on its left side, right side, nose down, nose up, and on its back. Make sure that the QC is kept still immediately after pressing the key for each step.

You should receive a `Calibration successful` message after the final step. If not, repeat the calibration procedure. Otherwise, restart the software by pressing the MOD button on the BeagleBone (alternatively you can restart the board by pressing the RST button, but then you would have to wait for the board to reboot and reconnect to WiFi) and reconnect Mission Planner to ArduPilot. Go back to the `Flight Data` screen, you should see a level HUD horizon. If you lose calibration in the future (i.e., if you don't see a level horizon), you can re-calibrate the accelerometer, or simply use the `Calibrate Level` option to level the HUD horizon.

Checkpoint 2. (2 points) Ask your lab instructor to double check that you have successfully calibrated the accelerometer.

We will skip calibration for other components, such as the ESCs (which have been pre-calibrated), as well as compass and radio calibration (since we won't be using these components).

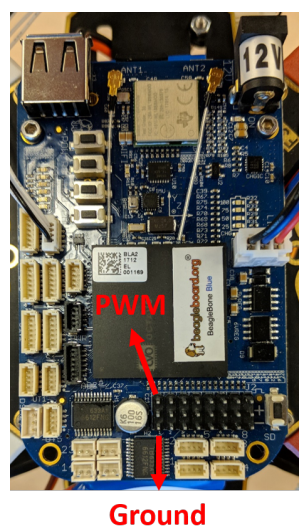
Question 3.7. (4 points) How can IMU readings be used to estimate roll, pitch, and yaw? How would you use the accelerometer or gyroscope for this estimation? Briefly explain the drawbacks of using each sensor. When using the gyroscope, assume that the initial orientation of the vehicle corresponds to 0-degree roll, pitch, and yaw.

Question 3.8. (3 points) Assume that we rotate the vehicle 90 degrees, such that the nose of the vehicle is facing downward. What are the pitch and roll values for this orientation? What would the accelerometer register for its internal X, Y, and Z axes in this position? Assume that other than gravity, there are no external forces being applied to the quadcopter.

3.5 Testing the motors

In this part we will first observe PWM signal outputs from the BeagleBone, and translate them into throttle values. We will then rotate the motors at various throttle levels and determine their directions.

Make sure the motor battery is disconnected, and that the main board is not powered on by disconnecting both the smaller battery and the DC adapter. Now find motor A (see Figure 1) on the QC, and disconnect its ESC from the main board. You can do this by following the 3-wire cable from the ESC to the BeagleBone. This should expose a 3-pin header on the main board. The pin closer to the outside of the board is ground. The middle pin is not used, and the remaining pin corresponds to the PWM signal supplied to motor A. Connect the PWM signal to the scope; there should be two wires: signal and ground. This will allow you to observe the PWM signal for controlling motor A on the oscilloscope.

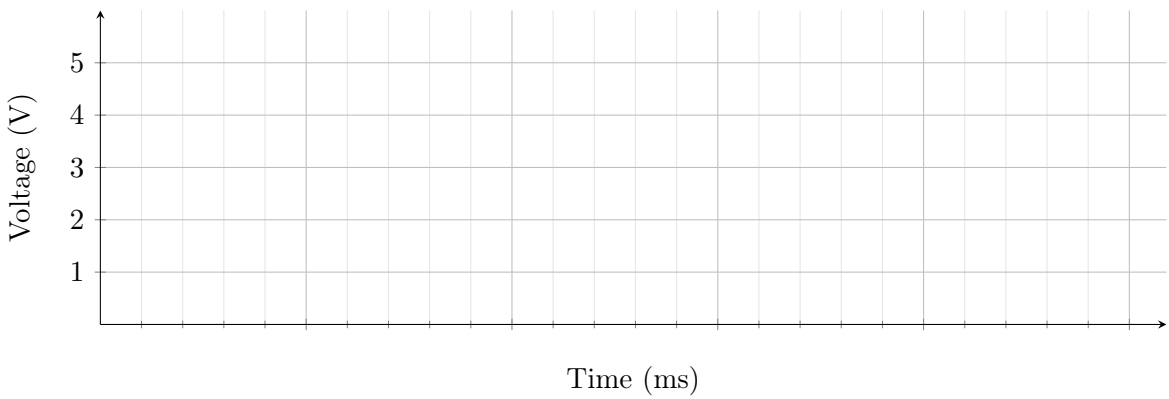


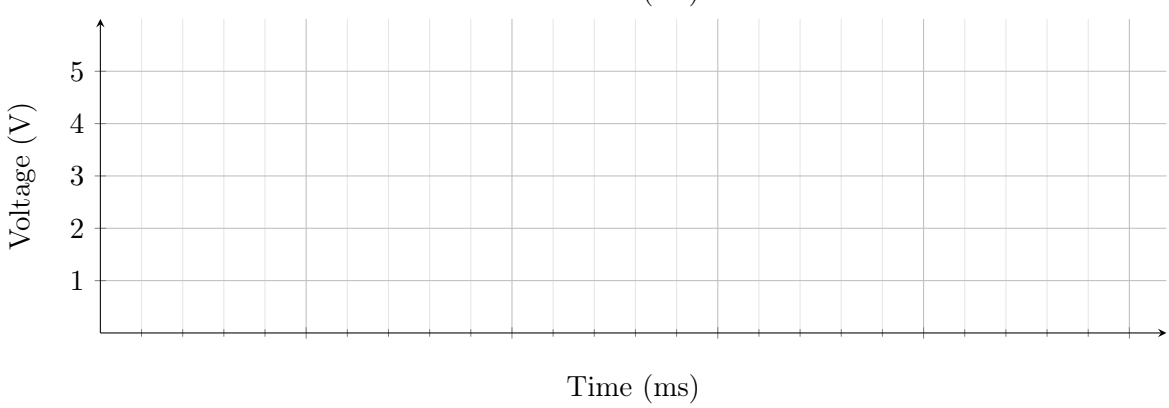
Checkpoint 3. (2 points) Ask your lab instructor to double check your setup before continuing.

Now turn on the main board, and connect Mission Planner to the QC to generate motor outputs. Look at the signal displayed on the scope, and estimate its PWM percentage. This signal should correspond to 0% throttle.

Go to `Initial Setup` → `Optional Hardware` → `Motor Test`. Set the throttle to 50% for 5 seconds, and click `Test motor A`. You see a change on the generated PWM signal. If you need more time, feel free to set a longer duration and repeat, or just freeze the scope.

Question 3.9. (4 points) Plot the PWM signals for both 0% and 50% throttle. Be sure to clearly indicate the width of the square wave with respect to the grid points on the screen. What duty cycles do you observe for each signal?





Once you are done observing the PWM signals, **power off the BeagleBone (always power off the board before altering its connections or you might damage it)** and reconnect motor A's ESC. **Double check the orientation of the connector** by comparing it with the other ESC connections (the orange wire should be located toward the inside of the board).

Power the board back on, and repeat the previous steps to get back to the `Motor Test` window. Now connect the 3-cell battery pack to the motors. You should hear a tone consisting of four beeps, the last of which indicates that the motors have been *armed*. Note that the motors do not arm unless the ESC observes a 0% throttle when powered on; this is intended as a safety measure in case someone has accidentally set a higher default value for motor outputs.

Checkpoint 4. (4 points) With a lab instructor present, repeat motor control for all motors at 10%, 50%, and full (100%) throttle for 2 seconds. Identify any faulty motors. If they all work, your quadcopter is ready for flight in the next lab!

Question 3.10. (4 points) Report the direction of rotation for each motor below. Verify that they are spinning in the correct direction, according to the background material.

Take the microSD card with you; you will keep this card for the remainder of the semester, and use it for completing the labs and your final project. You can plug this card into any of the quadcopters, in order to load your tuned QC parameters (which you will do in future labs) and customized flight software (which you will do for your final project).

We also highly recommend that you export the parameters of your QC before leaving. You can do this by navigating to `Config/Tuning` → `Full Parameter List` and clicking `Save to file`. This would allow you to revert to a working set of parameters at any point in upcoming labs.

3.6 Recommendations for Remote Participation

All team members, includuding remote participants, should actively participate in the lab. This section recommends tasks that might be most practical for remote participants to lead. These are recommendations: you are permitted to divide work in other ways.

- subsection 3.1: This task can be done by the remote student. It should be noted, however, that a team will need at least one computer in the lab to have Mission Planner set up. However, if the remote student works through this on their own, they will be able to quickly guide in-lab students through the process. The remote team member's familiarity with the user interface will also speed up tasks by the in-lab team members. Screen sharing may be helpful here.
- subsection 3.2: Remote students should be familiar with all safety precautions.
- subsection 3.3, subsection 3.4, and subsection 3.5: These sections include working on the quadcopter and using Mission Planner to interact with it. It is recommended to have the computer connected to the quadcopter also be sharing the screen to the remote student so that student can see how Mission Planner and the quadcopter interface with each other.
- This lab (necessarily) focuses on interaction between the quadcopter and a local computer. Later labs will focus on filtering, feedback–control systems, and algorithm design, many of which will be practical for remote participants.

4 Post-Lab (30 Points)

In this lab you were introduced to different components (hardware) in a quadcopter, as well as flight control and ground control station software. In the next lab, you will use a default flying mode included with ArduPilot to fly your QC.

The post-lab must be completed as a group, and submitted by one team member online on Canvas. The deadline for the post-lab is the same as the in-lab. You can complete this part on a separate sheet of paper.

Question 4.1. (9 points) Read the descriptions of all the recommended flight modes from ArduPilot's documentation.[3] Briefly explain three of the flight modes listed. Use three sentences or fewer for each flight mode.

Question 4.2. (6 points) Indicate whether GPS can be reliably used indoors. Why or why not?

Question 4.3. (6 points) Which flight mode(s) would be feasible and safest for indoor flying with the lab QC? Stabilize, AltHold, or Loiter? Why?

Question 4.4. (9 points) Consider a PWM signal that is high (at 3.3 V) for 3 seconds and then low for 9 seconds. Calculate the duty cycle of the wave and plot one of its cycles.

[3] `http://ardupilot.org/copter/docs/flight-modes.html`