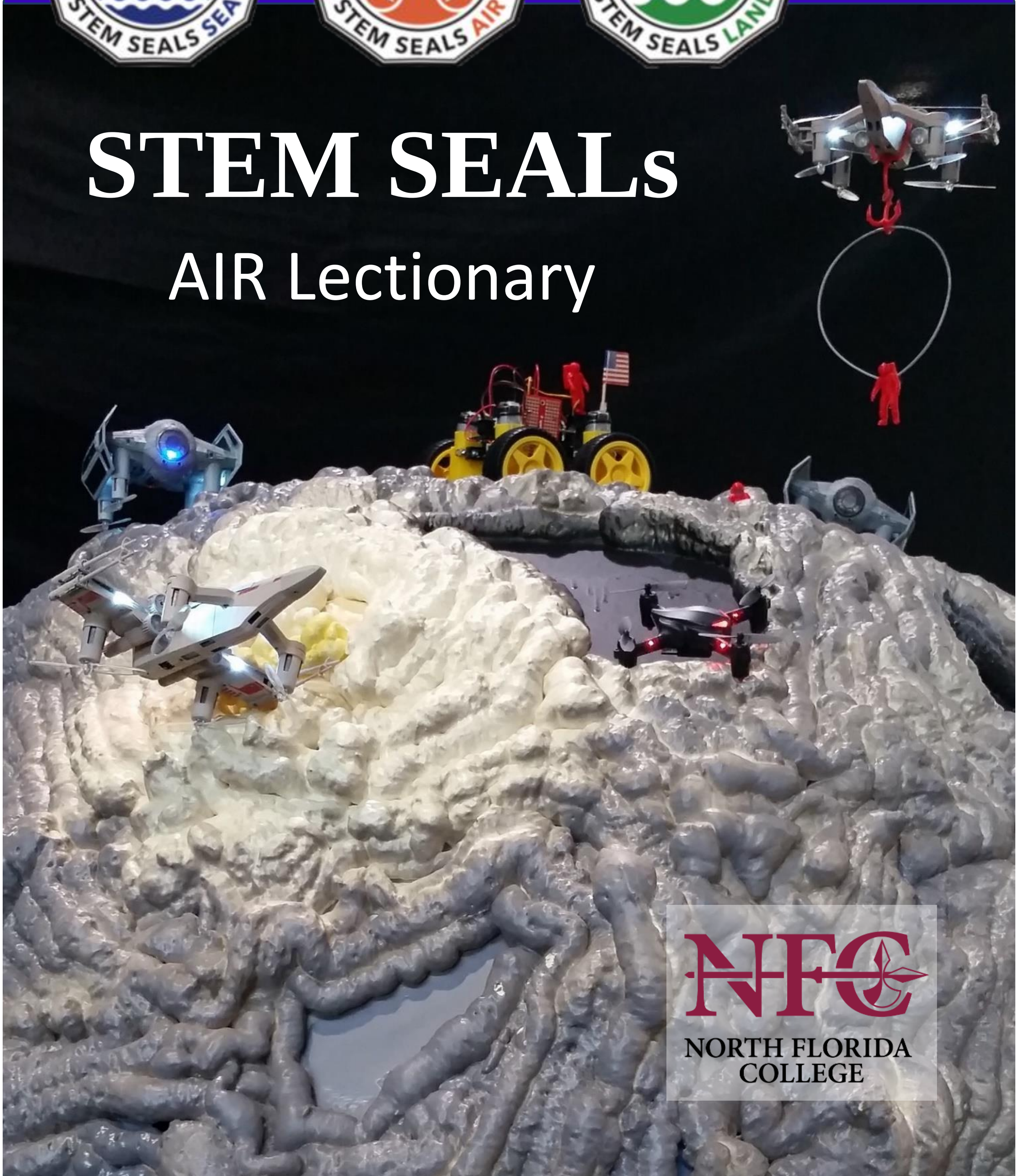
STEM SEALs

AIR Lectionary



NORTH FLORIDA COLLEGE

A Student Guide to the AIR Challenge

STEM SEALS
AIR Lectionary

A Student Guide to the AIR Challenge

Authors: Dr. Guenter G. Maresch and Lura L. Murfee

ISBN: 979-8-9874652-1-9

Printed by North Florida College

North Florida College
325 NW Turner Davis Drive
Madison, Florida 32340
United States of America

Synopsis

This document will guide you through the STEM SEALs AIR challenge. You will build a remote control for an UAV – unmanned aerial vehicle, learn the principles of flight, and learn to fly one or more drones.

Your challenge is to control your drone, test and improve the code you write for it, and finally master a search and rescue mission using your drone.

Table of Contents

Preface – The STEM SEALs Project

The STEM SEALs project began as a three-year program at North Florida College as a research effort for the improvement of learning and teaching in STEM (Science, Technology, Engineering and Mathematics) fields in rural environments. The core motivation was the observation by faculty that the majority but especially students from underprivileged families show a lack of familiarity or even basic exposure to scientific and technological elements, and therefore have major difficulties to adjust to the demands in their secondary educational phase.

The goal was to produce a program exploiting the attractiveness of modern devices such as robots, microcomputers, and unmanned aerial vehicles (UAVs or drones) and involving middle school students in the design of sea, air and land vehicles (hence the "SEALs" term borrowed from the United States Navy SEALs), their construction, testing, improvement, and finally using the vehicles during a competition event. (image)*"SEAL Trident, the special warfare insignia of the U.S. Navy SEALs* "

It was clear from the very beginning that college faculty are not experts in teaching middle school students. The STEM SEALs project involved middle school teachers in all three crucial phases of the program, during the design, review, and activity phase. Groups of teachers formed together with college faculty and held Design Team and Review Team workshops long before the final program for the students' activities was finalized. Then, during the STEM SEALs Summer Institute, students were led through the activities again, together, by groups of middle school teachers and college faculty.

During all phases a direct link was maintained between students, teachers, and faculty to the Research Team, which acquired data about each event. The Research Team analyzed the data and fed the results back to college faculty for further guidance for improving the next steps.

For each of the three disciplines, SEA, AIR and LAND two documents provide instructions. The first, the "Lectionary" is the student guide; it is the textbook needed by each student, containing modules describing each activity but also providing plans and images for clarification and troubleshooting when needed. A second document, the "Dictionary" was used as a guide for facilitators, i.e., teachers and / or parents, who might want to know more details for assisting the students, who might need to know alternatives in case an activity develops in an unexpected direction, or who may want to create their own materials and are looking for purchasing, engineering or manufacturing information. The Dictionary has not been published, yet.

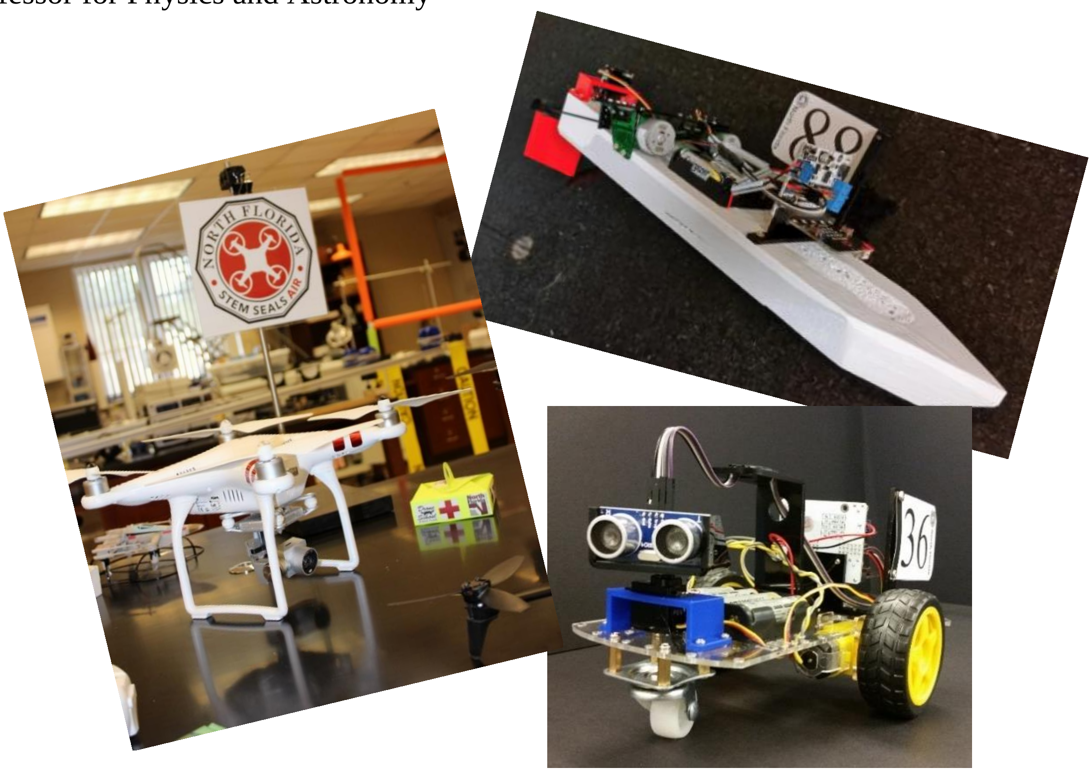
Although the initial three years of the STEM SEALs project were successful, the program itself is not hewn in stone as described in the materials. Due to time or resource limitations, teachers or parents may want to choose just certain modules from the program. This is certainly possible with the appropriate planning.

Also, the market changes rapidly, new products may be preferable due to better performance or price, or items described in the documents may be discontinued. The core of the STEM SEALs idea is not solely based on the devices used during those first three years. An open-minded STEM teacher or parent can adapt the program to different motors, sensors, computers or drones. Minor modifications likely are easier to accomplish than major changes.

All members of the STEM SEALs team, including the researchers at Cynosure Consulting, and the evaluators at Indikus Evaluation and Planning, gratefully acknowledge funding by the National Science Foundation and its Discovery Research PreK-12 (DRK12) program. Also, we thank Tri-County Electric Cooperative of Madison, Florida for their support. We are grateful to the Maintenance Department, Duplication, College Advancement, Computer Services and to the leadership of North Florida College for their help, their insightful contributions, their participation, and for being available and creative when in need. However, the most important people for the successful deployment of the STEM SEALs challenges are the teachers from the schools in our area. This project would not be possible without active participation and the guiding contributions of those educators.

Dr. Guenter Maresch

Professor for Physics and Astronomy



Introduction: The STEM SEALs AIR Challenge

The STEM SEALs project is about remotely controllable vehicles, and the AIR challenge is about aircraft, how they fly, how you can fly them, and what they are useful for.

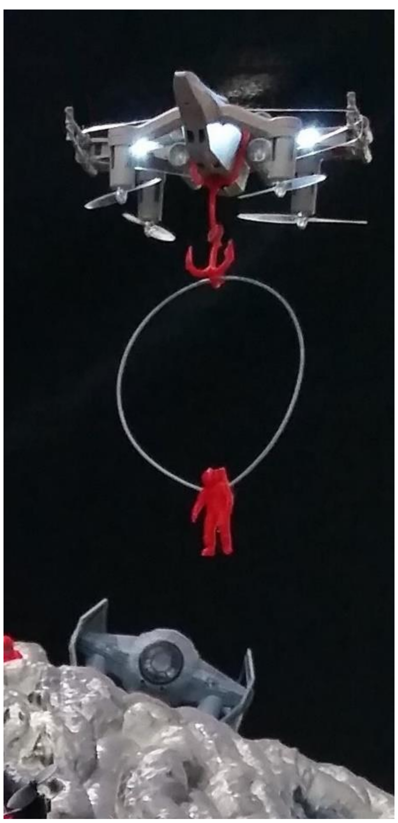
In Module 1 we will be learning about airplanes and drones. You will be flying a small, winged aircraft using a flight simulator. You also will be learning about the physics of flying.

In the following modules you will learn how to fly a drone, how to control it remotely from a computer, explore autonomous flying and later build a hand-held remote control for your drone.

You probably know already it is easy to fly a drone … but it is also easy to crash! You will explore different ways and write code, which flies the drone allowing you to practice and find out what works best for you and for the task at hand.

On the last day you will participate in several competition activities. That is an opportunity for you to demonstrate how well you understood the technology and rules of flying. The highlight is a rescue mission. You must pick up a stranded astronaut from an asteroid and fly him to safety.

When you complete all modules and participate in the competition at the end of the Summer Institute, you not only learned quite a bit about drones and flying and had fun while doing it, you also are to keep your drone and take it home together with the remote control for further explorations on your own.



Module 1: Drones

1.1 What is a Drone?

(This Module is supported by a Power Point Presentation.)

Merriam-Webster Dictionary:

A drone is a stingless male bee which does not gather nectar or pollen.

The verb “drone” means to make a sustained deep murmuring, humming, or buzzing sound, as droning bees do.

In music, a drone is an instrument that can sound a continuous unvarying tone, such as bagpipes do.

In aviation and space, a drone refers to an unpiloted aircraft or spacecraft. UAV.

Image 1

Drones or UAVs can

- Fly – unmanned, i.e., remote-controlled, or autonomous
- Image, taking photographs or recording video
- Transport a load
- Deliver a device or substance
- Support communications

Image 2

Some selected drone applications are:

- Event Photography
- Real-estate Photography
- Structural Inspection
- Utilities Inspection
- Agricultural or Livestock Management
- Environmental Protection Management
- Military and Law Enforcement

Image 3

Image 4

Image 5

Can you think of any other uses?

Image 6

Image 7

In the STEM SEALs AIR challenge, you will be flying some of the following drones:

The CoDrone Pro:

Image 8 & 9

The CoDrone Pro is the first programmable drone made for education. Code in Python or Blockly. The drone comes with a buildable and programmable Arduino-compatible remote as well.

Propel Star Wars collectable drones:

Star Wars X-Wing Fighter

Image 10

Star Wars Tie Fighter

Image 11

Module 2: Aeronautical Knowledge

(This Module is supported by a Power Point Presentation.)

2.1 How a Pilot Thinks?

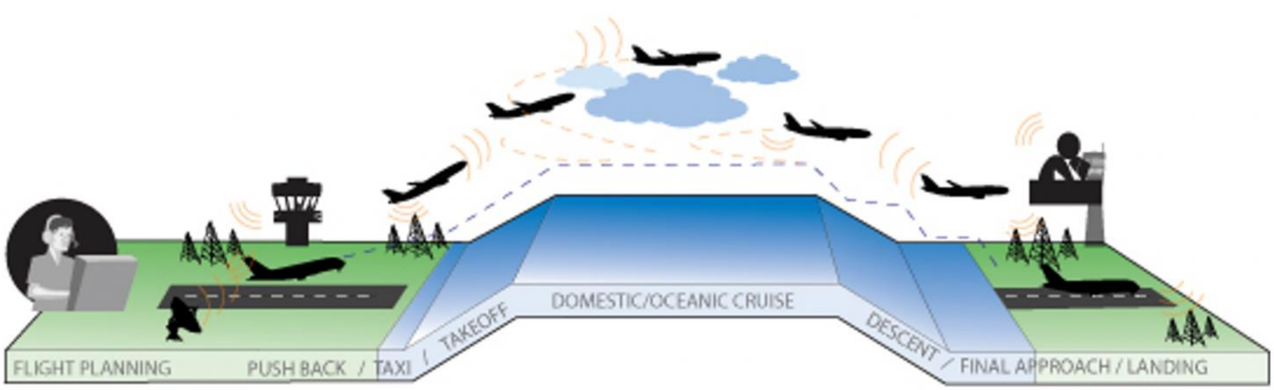
There are many things that go into flying. Pilots go through extensive training before taking off in an aircraft. Even flying small toy or hobby drones takes preparations and considerations that need your attention before flying.

Here are some items that pilots must consider:

- Know all safety precautions.
- Create a flight plan!
- Use a checklist.
- Understand the function, capabilities, and design of the aircraft.
- Understand the dynamics of how things fly.
- Know how to operate the aircraft.

To become a professional drone pilot or to even use drones in a professional manor such as event photography, the pilot in command must have a certificate from the FAA (Federal Aviation Administration) Drones used for recreation do not require certification.

In the STEM SEALs AIR Challenge, safety and operation procedures for each aircraft or drone that you will fly will be discussed right before flying and you will be given a handout for reference so that the information is close at hand. The flight plans will be determined by the goal or objective of each activity in the lectionary. So, let's jump into some basics for navigation and flight.



There are three phases of flight: (image 13)

1. Take Off
2. Cruise
3. Land

Do NOT forget though:
To prepare before take-off
And to know all that is needed to land!

2.2 Aircraft Navigation

You may be curious how pilots navigate. They do what travelers have been doing for centuries – they either navigate by sight or instruments. For some flights, a pilot may simply just look through the window for familiar landmarks to guide them along their course. You will get to practice this type of navigation in a simulated flight of a piper cub plane. Modern large aircraft pilots rely heavily on computerized instruments with the assistance of an autopilot and flight management computers that guide (steer) the plane along the planned route. Aircraft are monitored by air traffic control towers in the flight path to coordinate all aircraft. For the small drones you will fly, you will navigate by sight. The larger drones in the STEM SEALs fleet have cameras on board that send images to a connected tablet or device that can aid in navigation.

One thing that all types of navigation rely on is cardinal directions which are often depicted using a compass rose.

*A **compass rose**, sometimes called a **wind rose**, **rose of the winds** or **compass star**, is a figure on a compass, map, nautical chart, or monument used to display the orientation of the cardinal directions (north, east, south, and west) and their intermediate points. It is also the term for the graduated markings found on the traditional magnetic compass. Today, a form of compass rose is found on, or featured in, almost all navigation systems, including nautical charts, non-directional beacons (NDB), VHF omnidirectional range (VOR) systems, global-positioning systems (GPS), and similar equipment.* *(https://en.wikipedia.org/wiki/Compass_rose)*

Cardinal directions:	Ordinal directions (or intercardinal directions):
North (N)	**Northeast (NE)**
South (S)	**Southeast (SE)**
East (E)	**Northwest (NW)**
West (W)	**Southwest (SW)**

GPS Navigation (Global Positioning System) uses: **Longitude and Latitude**

Internet Resources:

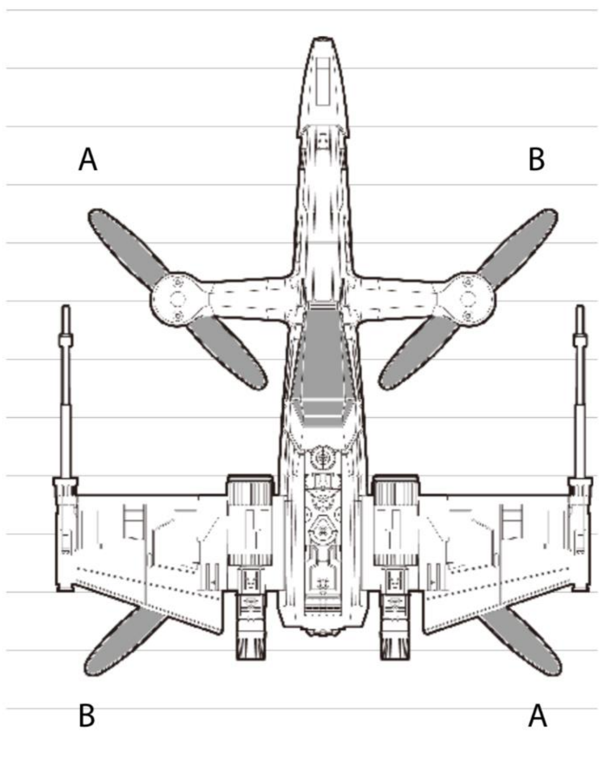
Check out the following to learn more about navigation:

- What are Cardinal Directions and a Compass Rose? (video) https://youtu.be/lkfq37bMVBY

2.3 How Aircraft Fly

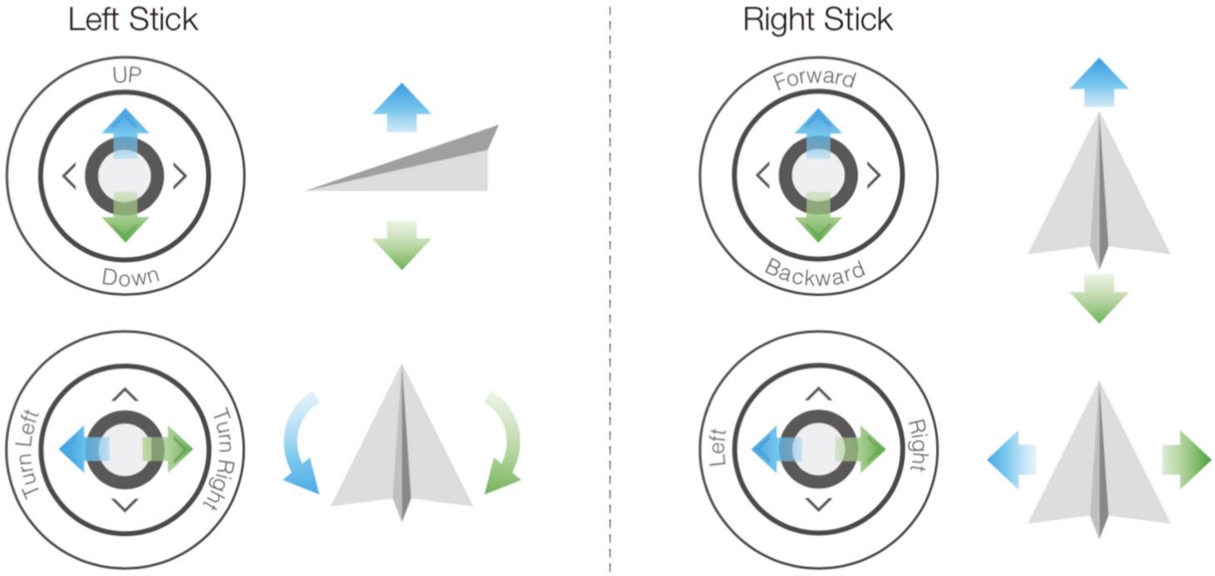
The following are components of flying as discussed in the instructor demonstration and Power Point presentation. The diagrams that follow show the various control mechanism of drones and how to utilize them to navigate the drones.

a. **Lift**
b. **Thrust**
c. **Pitch**
d. **Roll**
e. **Yaw**
f. **Angle of Attack**
g. **Drag**
h. **Controlled Flight: the sticks of the RC**



Propellers provide thrust for lift, propulsion, and navigation.

*Because of the **Conservation of Angular Momentum,** there are always A-B pairs of propellers turning in the opposite direction. [Star Wars X Wing Manual t65-x-wing.pdf]*

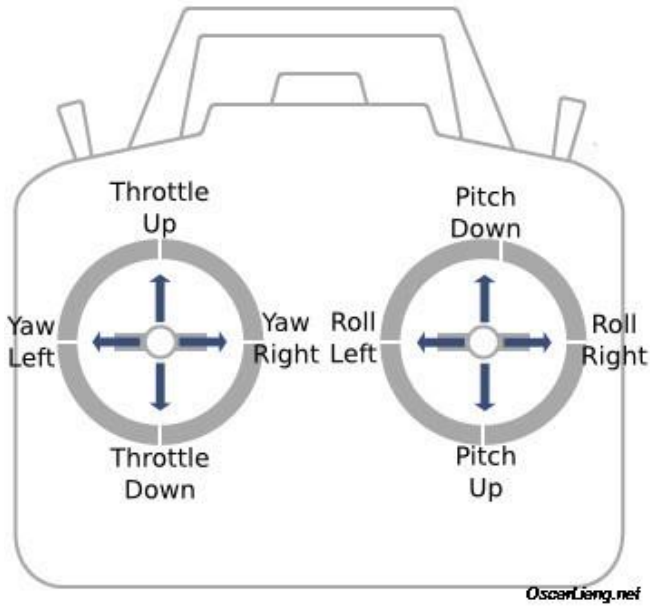


Joysticks on the remote-control enable flight maneuvers.

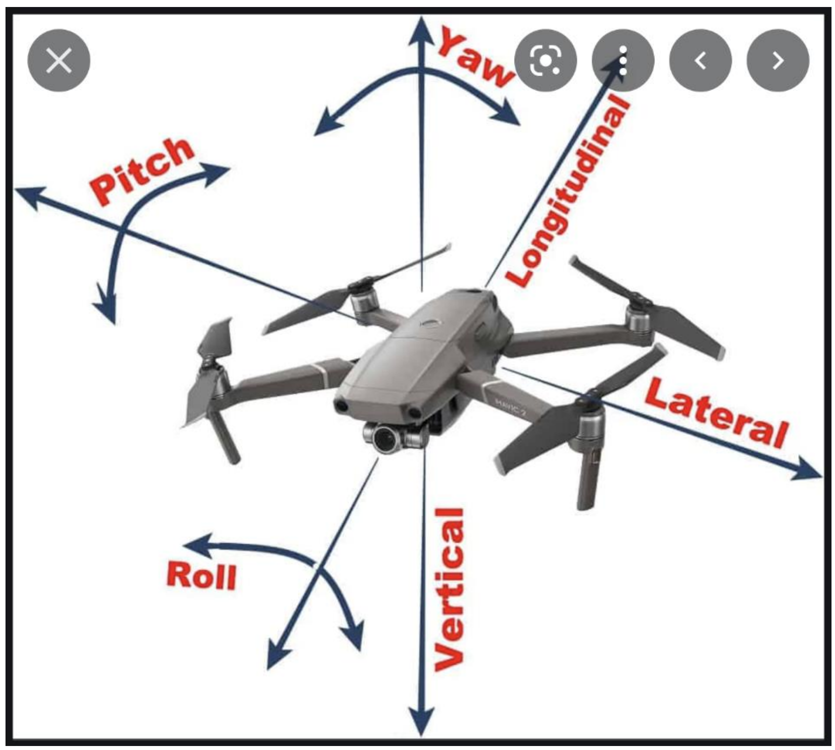
Throttle:	*L*	*up / down*
Pitch:	*R*	*forward / backward*
Roll:	*R*	*left / right*
Yaw:	*L*	*left / right*

[Phantom_4_Quick_Start_Guide_v1.2….pdf]

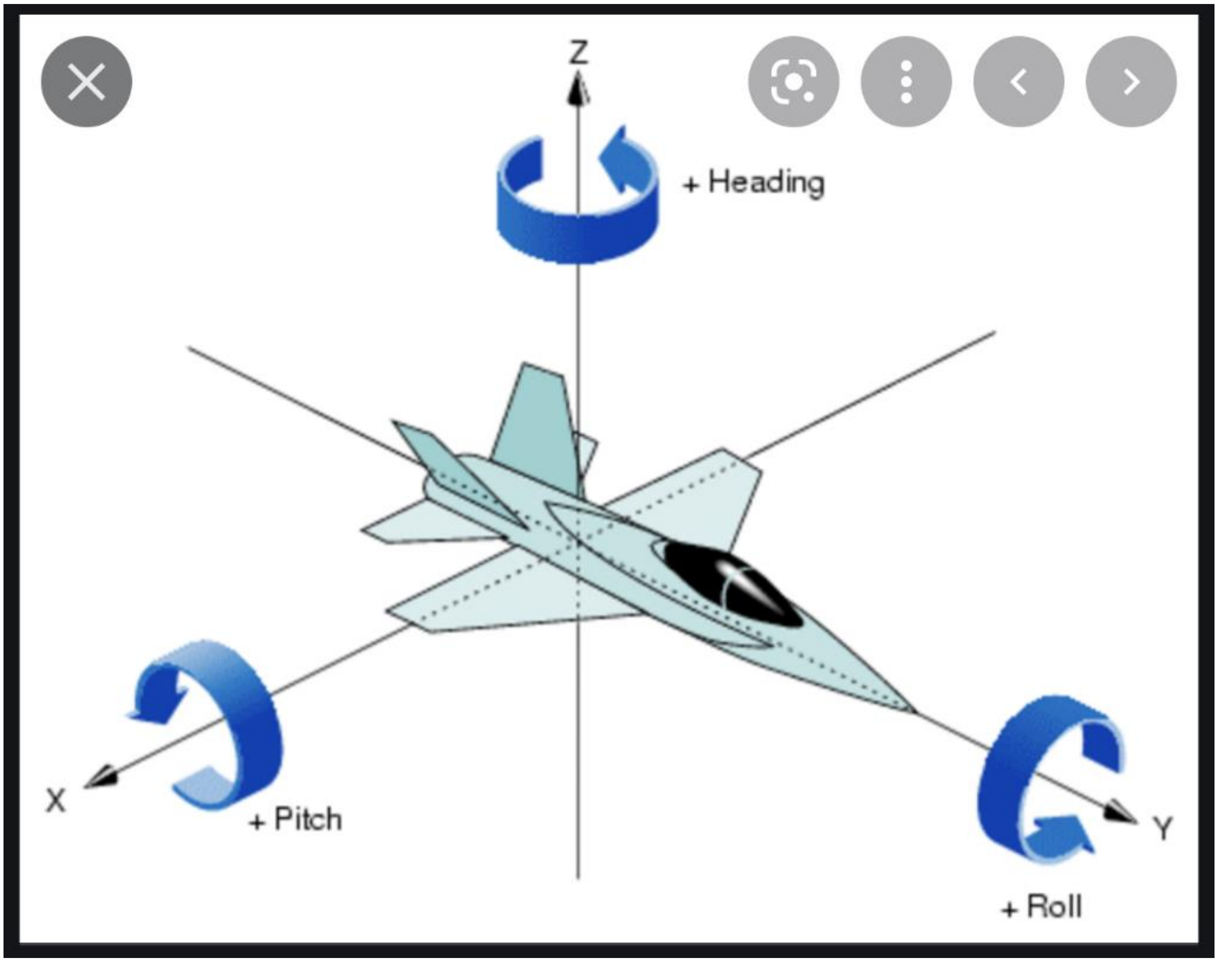
Throttle, Pitch, Roll, and Yaw on a Remote-Control with Two Joysticks.
[fromwhereidrone.com]



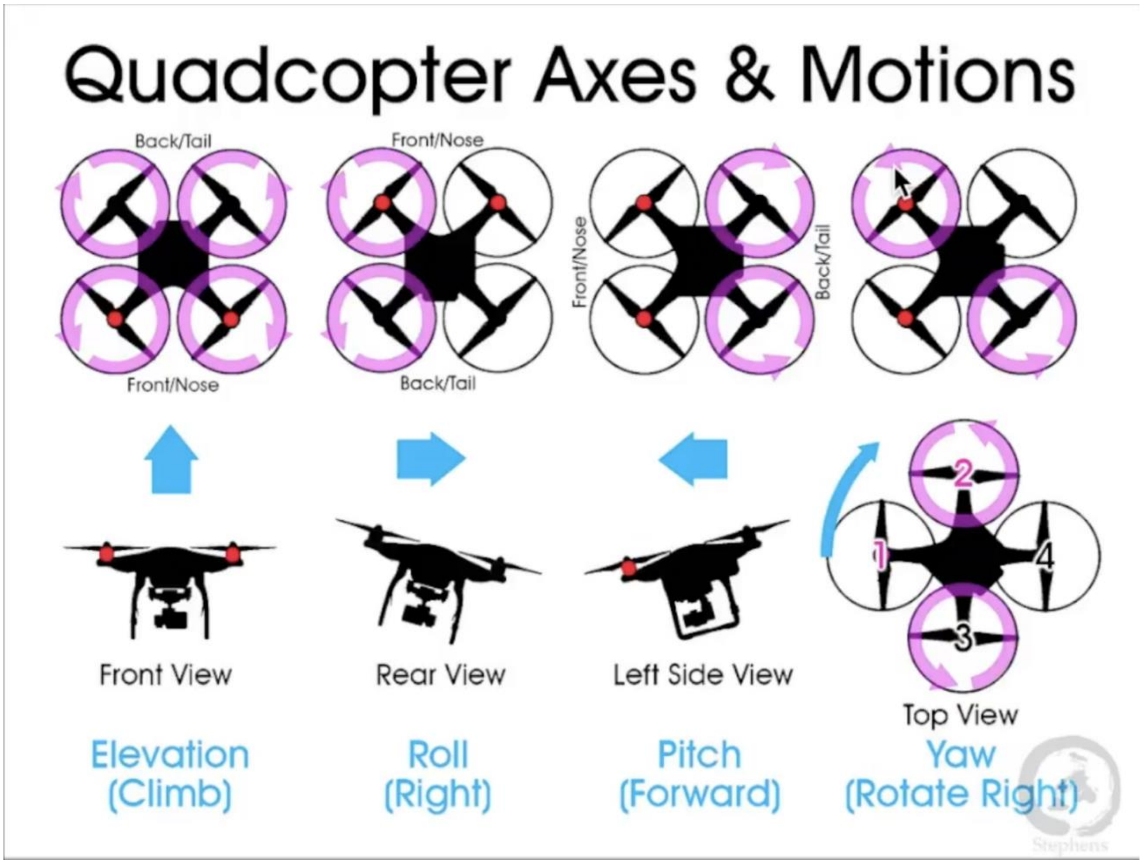
Throttle, Pitch, Roll, and Yaw Joystick Functions on a (Mode 2) Remote-Control.
[Oscar Liang "Learn How to Fly FPV Drones", oscarliang.com]



Motion Control of a Quadcopter]: The Three Axes.
[LetUsDrone.com]



Motion Control of a Fighter Jet: The Three Axes.
[ResearchGate.com]



Throttle (Climb), Roll, Pitch, and Yaw pictured with a Phantom drone.

[YouTube video "CoDrone Tutorial: Throttle, Yaw, Pitch, Roll" by Guych Kawasaki]

How a quadcopter controls its attitude:

- To climb, all four motors need to spin faster. All four of them provide lift. If they all spin faster there is more lift than needed for hovering. Consequently, the aircraft climbs.
- To roll right, the left two motors need to spin faster while the right two motors slow a little down. This provides more lift of the left side of the aircraft than of its right. Therefore, it will roll right. To roll left, the right motors get more power than the left motors.
- For the aircraft to pitch forward, the rear two propellers must spin faster while the two front ones reduce power. This causes the aircraft to have more lift in the rear than in the front and it will pitch forward.
- To yaw right, the two propellers, which spin counterclockwise (and are located on opposite sides of the aircraft) must spin faster. Due to conservation of angular momentum this causes the aircraft to rotate clockwise, which is the same as yawing to the right.

Internet Resources:

Check out the following to learn more about the principles of flight and how to fly a drone:

- Principles of Flight
  - https://www.sciencelearn.org.nz/resources/299-principles-of-flight
  - https://pilotinstitute.com/principles-of-flight/
- How to Fly a Quadcopter
  - https://uavcoach.com/how-to-fly-a-quadcopter-guide/

Module 3: You Fly – Flight Simulator

Now it is time to put some of the concepts of flight to practice.

Let's learn to fly safely first! How?

Using a flight simulator.

We will use the FSX- Flight Simulator X from Microsoft.

3.1 The Boing B737-800 Demonstration

Before you take the controls of the Flight Simulator, watch this demonstration. Pay attention to how the pilot can navigate between different internal and external views, how they approach the runway and use the navigation tools to stay on course.

Flight Demonstration: KTLH to KVLD

- Free Flight > Load
- Aircraft Boeing 737-800
- Location United States > KTLH
- Aircraft on taxiway facing runway 36 heading W
- IAS 250 kn
- Altitude 6000
- Heading 000
- Flaps 5 deg
- Radio > NAV1 > 110.90 ILS runway 35
- Flights > Flight Planner > from KTLH to KVLD, cruising at 12000, IFR
- **"`" ATC, get clearance and take off**
- Takes ~ ½ hour (71 mi) … to land.

3.2 Midwest Fly-In

Now that you've seen some of the basics with the flight simulator, it's your turn to try out flying.

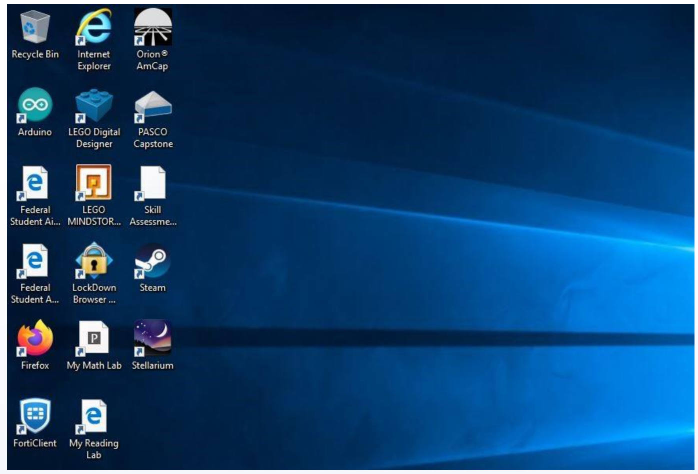
Try This: Login to the Flight Simulator

1. Get a laptop PC and turn it on. Check the battery life – plug into a power source if necessary.
2. Use the following to login onto the PC:

   Username: reset

   Password: Welcome1

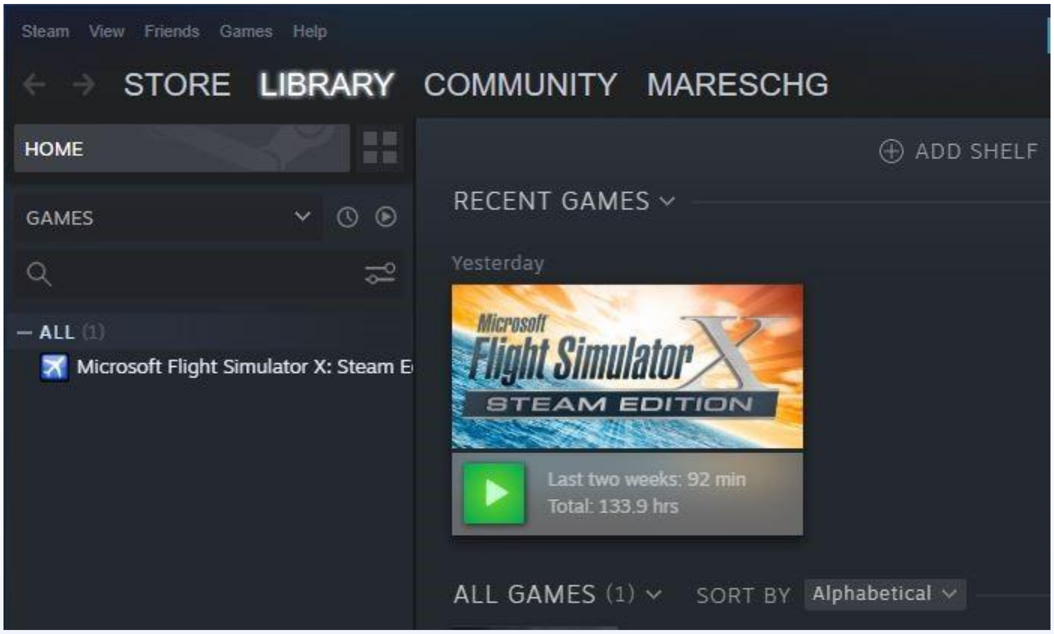
3. Find the icon for "STEAM" on the desktop and start it.



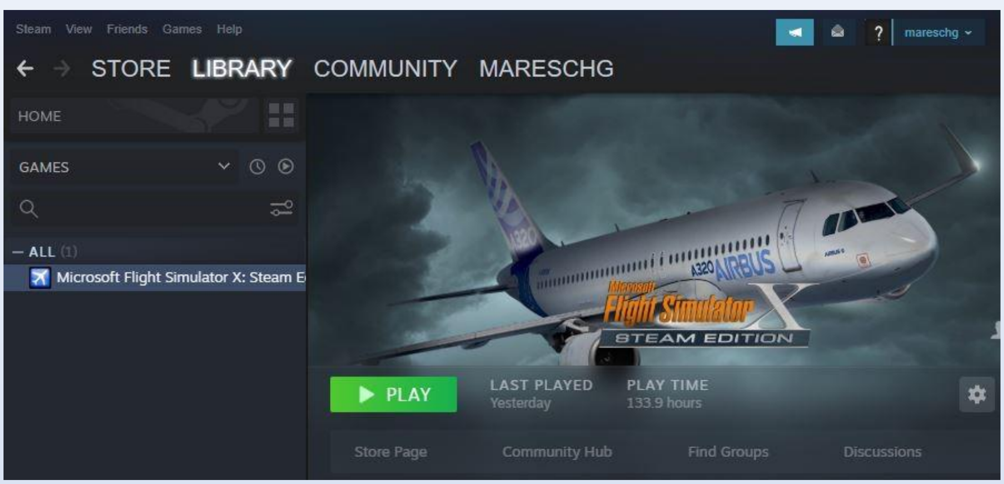
4. To log into the Steam account, you will need the PC identification number. You will find this on the underside of the laptop. The number will look something like this: B34-112-03. The last two numbers are the PC's identification numbers.
5. Log into the Steam account, use the account name "stemsealsxx" with the "xx" being replaced by your PC's identification number you just looked up. The password is "steamseals1xx". Note the added number 1 and again use your PC's identification number instead of the "xx". So in this example, it would be stemseals134.

Try This: Login to the Flight Simulator (continued)

6. Click on the green “Play” button of FSX in the Steam Library.



7. Click on the green “Play” button in the Steam Library.



8. “User Account Control” will ask you “Do you want to allow this app to make changes….?” Click “Yes”
9. You also may be asked by “Security Alert” “Do you want to proceed?” after it informs you that the identity of this web site of the integrity of this connection cannot be verified. Click “Yes”.

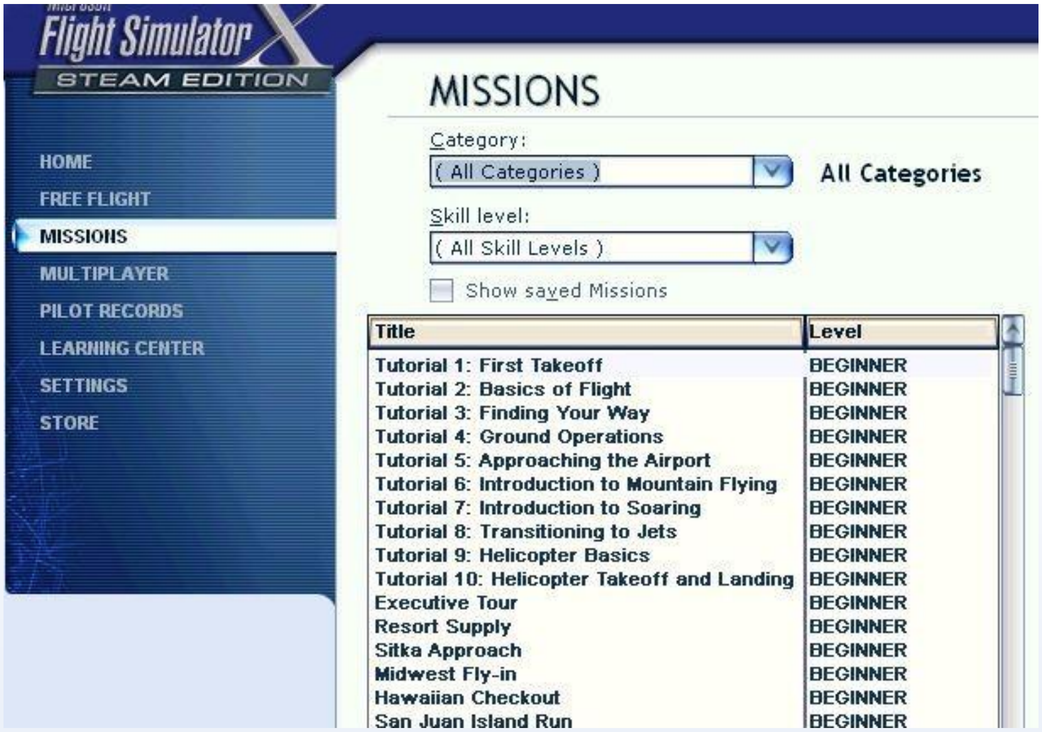
Try This: Login to the Flight Simulator (continued)

10. Wait until you see FSX's main navigation panel. There may be some additional information in the "Home" window. Ignore this extra information for now.

Now that you know how to log into STEAM and the Microsoft Flight simulator, let's try out one of the simulations.

Try This: Midwest Fly-in

1. Click on "Missions" in the main navigation Panel.
2. Choose from the many listed missions (scroll down to search) until you find "Midwest Fly-in" and click 'Go to Briefing".



3. Read the "Overview", "Details", and "Maps & Charts". You will find these by using the tabs.

Try This: Midwest Fly-in (continued)

4. Click the “Fly” button and wait…..
5. Watch and listen to the prompts.
6. Follow the instructions you hear.
7. Try to find and land on the runway.

The Basic FSX Joystick and Keyboard Commands:

- “.” on keyboard — brakes, press to release or to break
- F9 key — cockpit view
- F11 key — spot plane view
- arrow keys on NumPad — change your viewpoint
- “5” key on NumPad — center view
- lever behind joystick — throttle, for takeoff push all the way up
- F12 key — takes a screenshot*

You will only have one attempt!
After your first attempt, stop, let your partner try, and then discuss your experiences.

Flight Practice Assessment:

1. Did you miss the trees after take-off?
2. Did you find the runway at the fly-in?
3. How did you approach the runway for landing?
4. Did you see the PAPI (Precision Approach Path Indicator) lights at the runway?
5. Could you align the Piper Cub with the runway?
6. Where on the runway did you touch down?
7. Could you stop the Piper Cub on the runway?
8. Did you taxi off the runway after landing?

How do think your first flight went?

Now, read the next section 3.3 for some pointers and tips to help you fly the simulator better.

3.3 Troubleshooting for Midwest Fly-in

Here are some useful tips to improve your flying!

- **Cannot see where you are going?**

  On the ground, at low speed, the Piper Cub sticks its nose way up and you can't see where you are going.

  Give full thrust and wait until the aircraft has enough speed to lift the tail off the ground (you can watch it by pressing F11!). Then you can see.

- **Oh no! I am heading for the trees!**

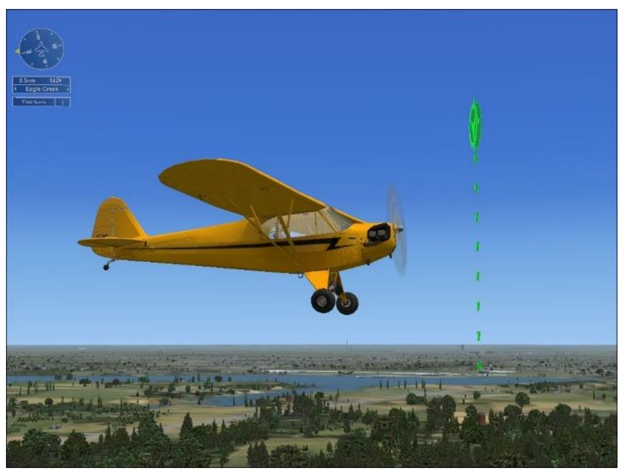
  Yes, the runway on the ranch ends at the fence. You must have enough speed to take off before you get there.

  Full throttle from the very beginning and once the plane has enough speed you go up even faster with the joystick. But be careful: A too large angle of attack reduces speed! You do not want the aircraft to stall. That will end in a crash.

- **I can't find the destination!**

  Looking left, either in cockpit view or spot plane view, you can find a green indicator. This is where the runway is on which you want to land.

  But wait: it you turn right now in that direction the runway will be across your flight path! You need to keep going a little further and then turn. This will allow you to line up with the runway for a successful landing.



- **How do I line up with the runway?**

  If the runway appears to stretch from right to left, you need to keep going more to a point straight in front of the runway. This is the view to the left in cockpit view. You see the left wing at the top.

- **How do I turn and line up to the runway?**

  The Piper Cub has now reached a point "in front of the runway". As viewed from the aircraft, the runway seems to stretch from a near point (this is where we want to land) to its end "behind it".

  This is the time to turn left and line up with the runway. The better you accomplish good alignment the easier it will be later to stay on the runway. Also reduce throttle to let the aircraft glide down towards the beginning of the runway.

- **Am I lined up correctly?**

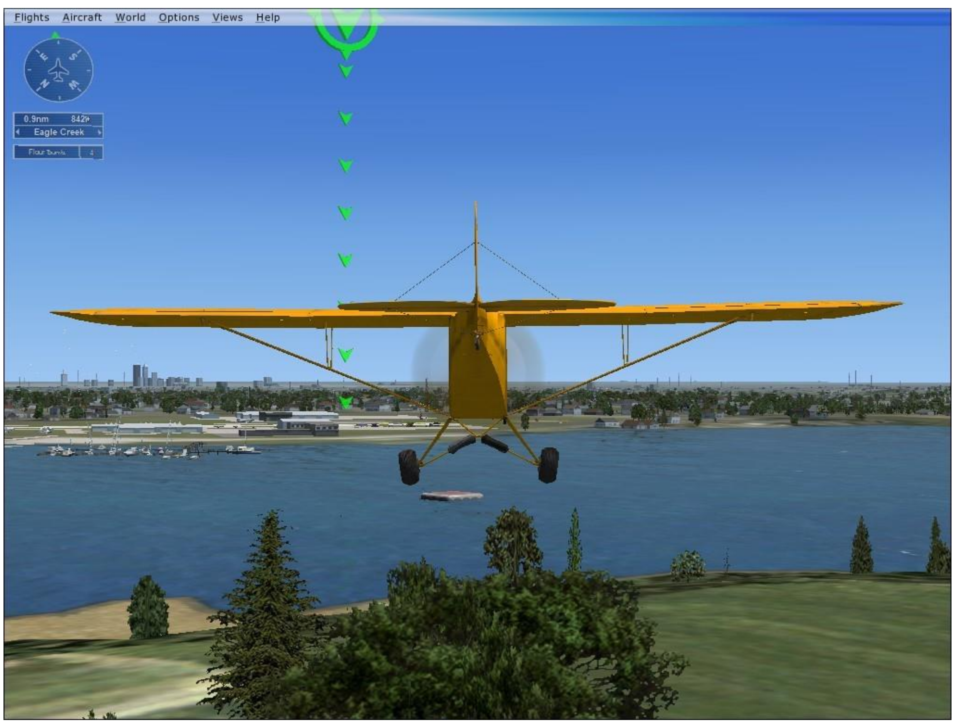
  You are striving for a position like this: The aircraft flies in the same direction as the runway. It descends towards the beginning of the runway to leave enough space for slowing down after landing.

  Notice the four lights at the left and the beginning of the runway. These are the PAPI lights which provide guidance during the descend. When two of the are white and two are red the aircraft has the correct altitude. More red lights mean you are too low, more white lights you are too high to land successfully.

➢ **Did you notice the Flour Bomb Target?**

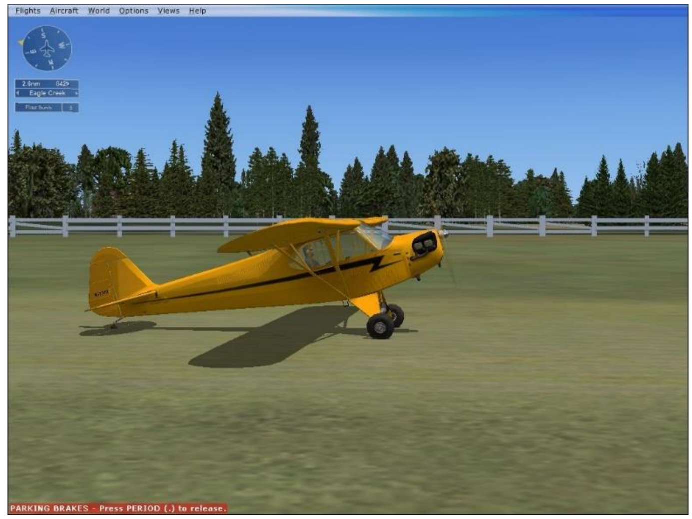
The float to the left of the Piper Cub in this image is the target. With shift-D you can drop a flour bomb. See if you can hit the target!

To be able to do so you must be low – but not too low and heading directly towards the target.



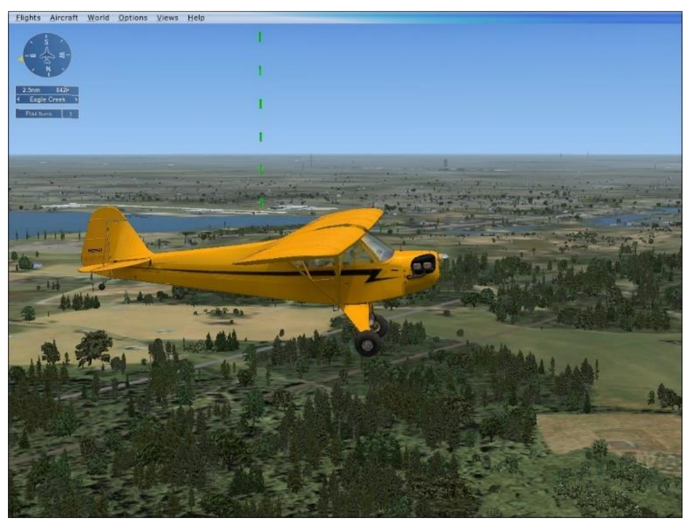
➢ Other tricks and tips:

Piper Cub at zero speed with the tail wheel touching.

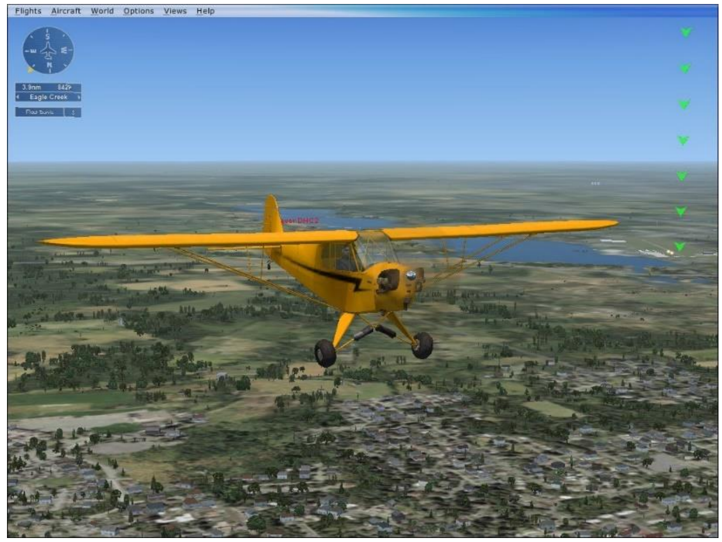


If the speed is high enough to provide some lift to the aircraft the tail wheel lifts off first and levels the plane.

When flying straight South we can see the runway to the left. Keep looking at the runway to decide when to turn for the approach.

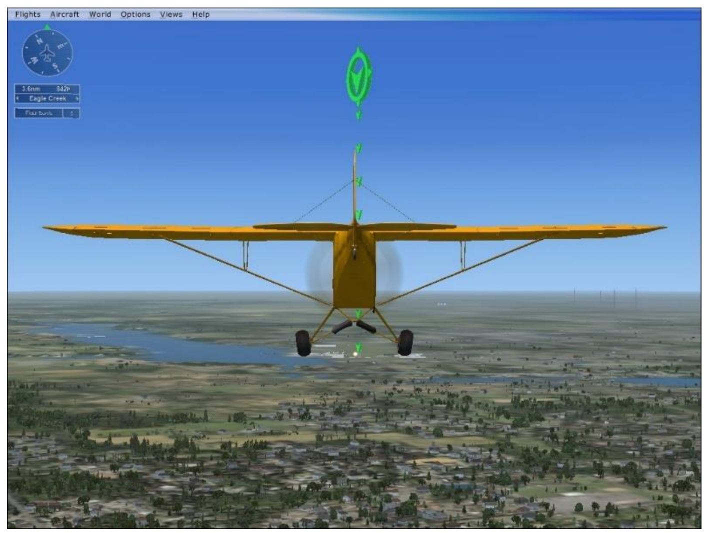


Just about time to turn. We flew away from the runway (see the distance indicator beneath the compass?) to give us more time for lining up.

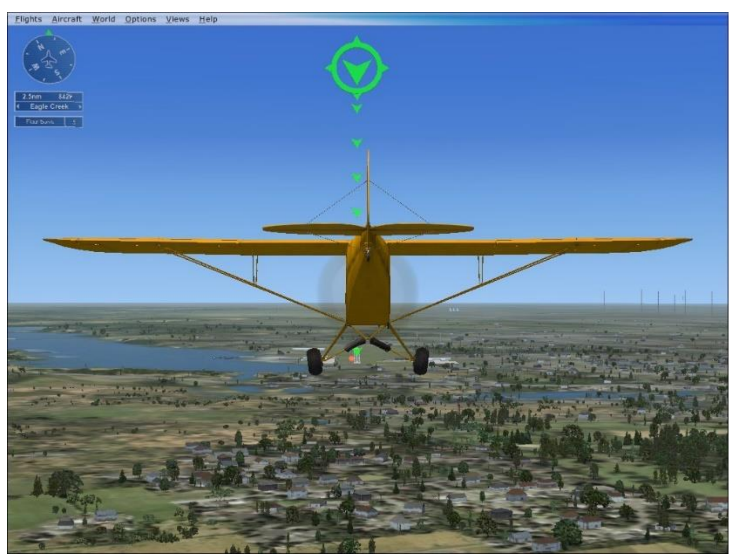


After the turn, now perfectly lined up with the runway. But: too high! (Do you see the all-white PAPI lights?)

Throttle is the tool to control if you want to climb or descend. When you are too high simply reduce throttle and wait.



The line-up with the runway is good. But the aircraft is too low! Notice the all-red PAPI lights.



Cockpit view well lined up with the runway, only a little too high.

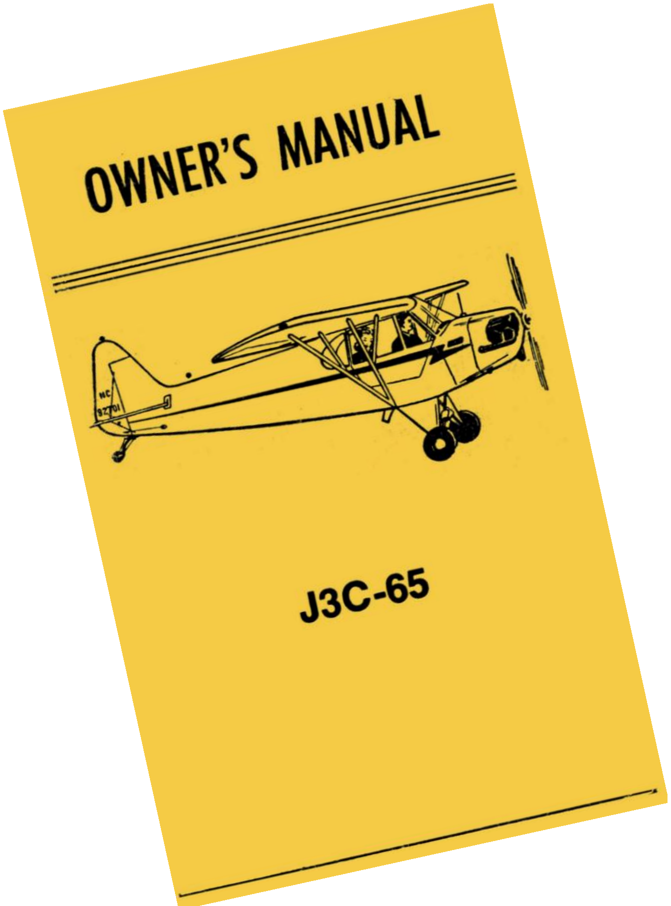
Final approach, line-up is perfect, a little high, but not too much. The Piper Cub easily tolerates small deviations from perfect.

Final approach shortly before touch-down with perfect line-up and at the perfect altitude (see the two red and two white PAPI lights!).

"PAPI" means "precision approach path indicator". These are a set of lights positioned beside the runway providing pilots with a visual indicator of their aircraft's position relative to the correct glidepath for the runway. Two of each shows the pilot is on the correct path for landing.

3.4 Know your Aircraft

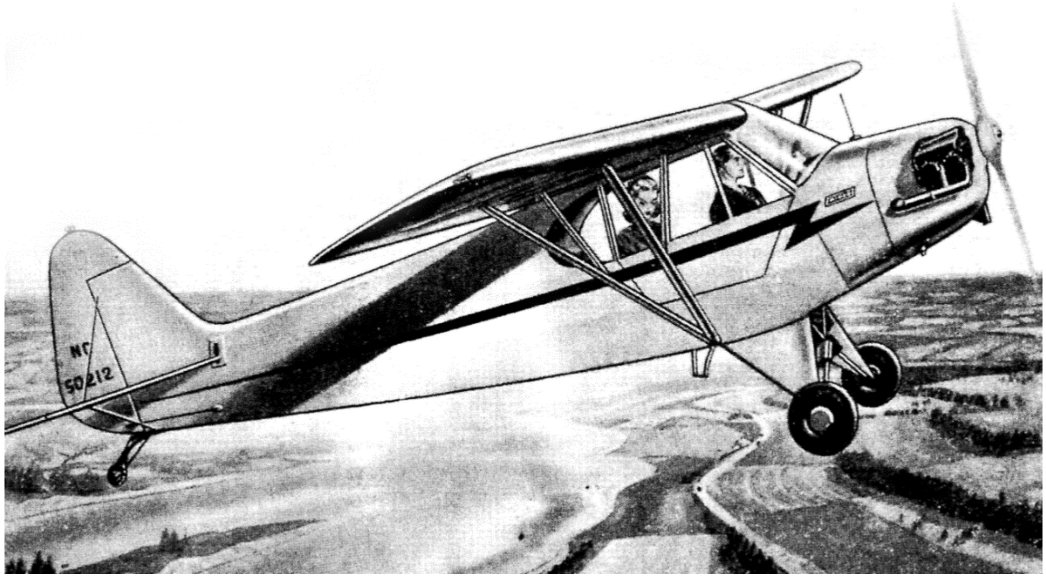
Before you begin your second flight to master the Midwest Fly-In mission, have a look at some excerpts from the Piper Cub's Owner's Manual. Before flying any aircraft, it is important to understand how it operates and under what conditions you will get optimal performance.



FLYING HINTS

The Piper Cub Special represents more than 15 years of diligent aircraft engineering and manufacturing experience. Its simplicity of design and construction, its low operating and maintenance costs, its inherent stability, ruggedness, and its outstanding safety and ease of flying, have made it the most popular airplane in aviation history. The Piper Cub Special is the time-tested product of millions of hours of flying under all conceivable conditions both in the military and in peace time.

There are hints on starting, flying, stopping, and other related topics that are important to the owner who wants to conserve his airplane—keep it in maximum airworthy condition—and enjoy a full measure of flying satisfaction.



F. GENERAL FLYING

(1) For takeoff use full throttle, headed into wind. Airplane loaded will become airborne at approximately 39 M.P.H. Best climb speed is at an indicated 55 M.P.H.

(2) Indicated R.P.M. for cruising speed of 73 M.P.H. is 2150. Take-off R.P.M. is 2300. Do not fly at full throttle over 3 minutes.

G. APPROACH AND LANDING

(1) Push carburetor heat ON prior to throttling back for glide, or for any other flight maneuver.

(2) Glide between 50-60 M.P.H. depending upon loading of airplane and gust conditions.

NOTE—"Clear" engine by opening throttle gently, every 200-250 feet of descent during a long glide so that engine temperature will be maintained.

Throttle action on the part of the pilot should be smooth and gentle at all times.

TEN COMMANDMENTS

For Safe Flying

1. THOU SHALT NOT BECOME AIRBORNE WITHOUT CHECKING THY FUEL SUPPLY: It only takes a few minutes to gas up . . . it may save you a forced landing.
2. THOU SHALT NOT TAXI WITH CARELESSNESS: Taxi slowly and make S turns to clear the area in front of the nose. Know the proper use of the controls for taxiing in a strong wind.
3. THOU SHALT EVER TAKE HEED UNTO AIR TRAFFIC RULES: Keep a constant lookout for other aircraft. Follow the rules so that pilots of other planes will know what you are going to do.
4. THOU SHALT NOT MAKE FLAT TURNS: This is particularly important when making power-off turns. You steer with the ailerons, not the rudder.
5. THOU SHALT MAINTAIN THY SPEED LEST THE EARTH ARISE AND SMITE THEE: Don't be fooled by the increase in ground speed resulting from a downwind turn. Keep sufficient airspeed.
6. THOU SHALT NOT LET THY CONFIDENCE EXCEED THY ABILITY: Don't attempt instrument flying in adverse weather conditions unless you have the proper training and the necessary instruments. Instrument flying is a highly developed science. Don't pioneer.

7. THOU SHALT MAKE USE OF THY CARBURETOR HEATER: The carburetor heater is your friend. Know when to use it. Remember that it's easier to *prevent* ice in the carburetor than to eliminate it after it has formed.
8. THOU SHALT NOT PERFORM AEROBATICS AT LOW ALTITUDES: Aerobatics started near the ground may be completed six feet under the ground. There's safety in altitude.
9. THOU SHALT NOT ALLOW INDECISION IN THY JUDGMENT: Be certain! You can't afford to make errors of judgment. "I think I can make it" is on the list of famous last words.
10. THOU SHALT KNOW ALWAYS—THE GOOD PILOT IS THE SAFE PILOT: It's better to be an old pilot than a bold pilot.

Quiz: What did you learn about the Piper Cub?

1. You should not fly at full throttle for more than ____________________.
2. What is the best climb speed of the Piper Cub? ____________________.
3. You should steer with the ailerons, not the ____________________.
4. It's better to be an old pilot than a _____________ pilot.

Here's some more information about the Piper Cub that might be helpful:

Flight Instruments:

Aircraft instruments are confronting arrays of dials, gauges and gadgets located in the cockpit of an aircraft. Pilots rely on these instruments to understand where the plane is, how fast it is travelling and what it is doing.

The flight instruments of the Piper Cub are

- Heading indicator (the compass in the middle)
- Attitude indicator (the level below the compass)
- Airspeed indicator (the small instrument on the left)
- Altimeter (the small instrument on the right)

The engine instruments of the Piper Cub are

- Tachometer (engine rpm, the large instrument on the left)
- Temperature gauge (coolant temperature, top of large instrument on the right)
- Pressure gauge (oil pressure, bottom of large instrument on the right)

In the cockpit of larger and more complex aircraft the instrument panel can be overwhelmingly busy. But any aircraft has the above listed instruments in some form or another.

Shown above is the instrument panel of a Boeing 737-800, one of the most popular aircraft you may experience when traveling across the United States and many other countries.

Can you find the Altimeter in the image of the B-737 instrument panel?

3.5 Planning and Completing Your Second Midwest Fly-in.

The three important rules (not only for flying!):

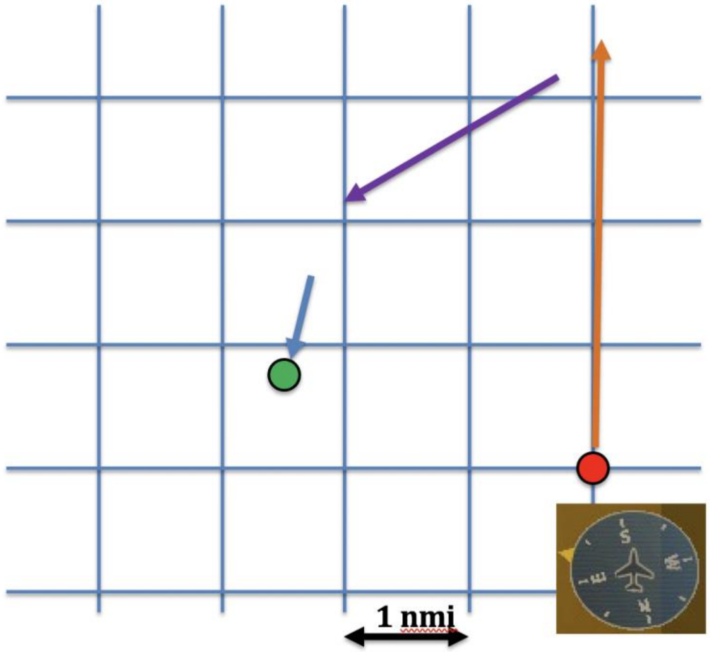
1. Know where you are!
2. Know where you want to go!
3. Have a plan!

Before take-off we can see where we are (1):

a. Somewhere on a ranch,
b. heading a little West of South (compass!),
c. 2.6 nm (nautical miles) from our destination,
d. at an altitude of 842 feet.

Our destination is (2) Eagle Creek Reservoir

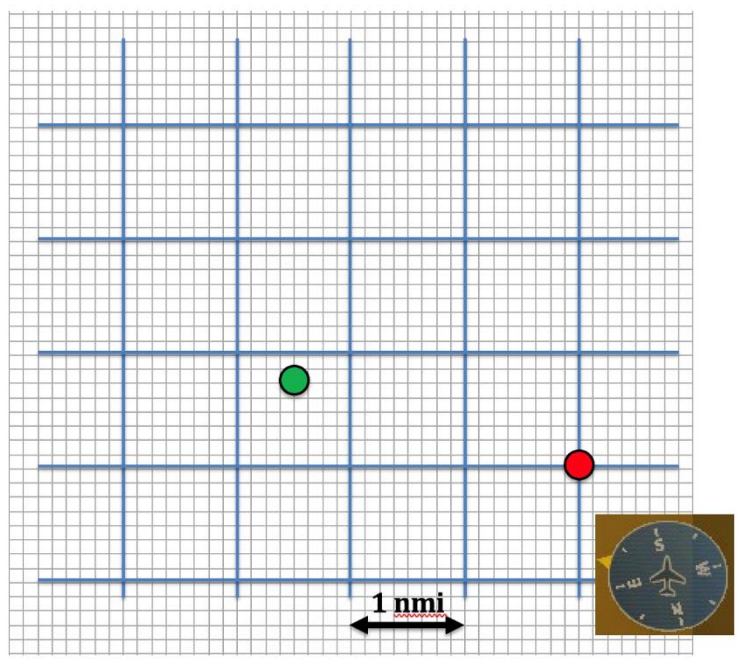
Let’s make a plan (3)!



The red dot is our location on the map

The green dot is our destination

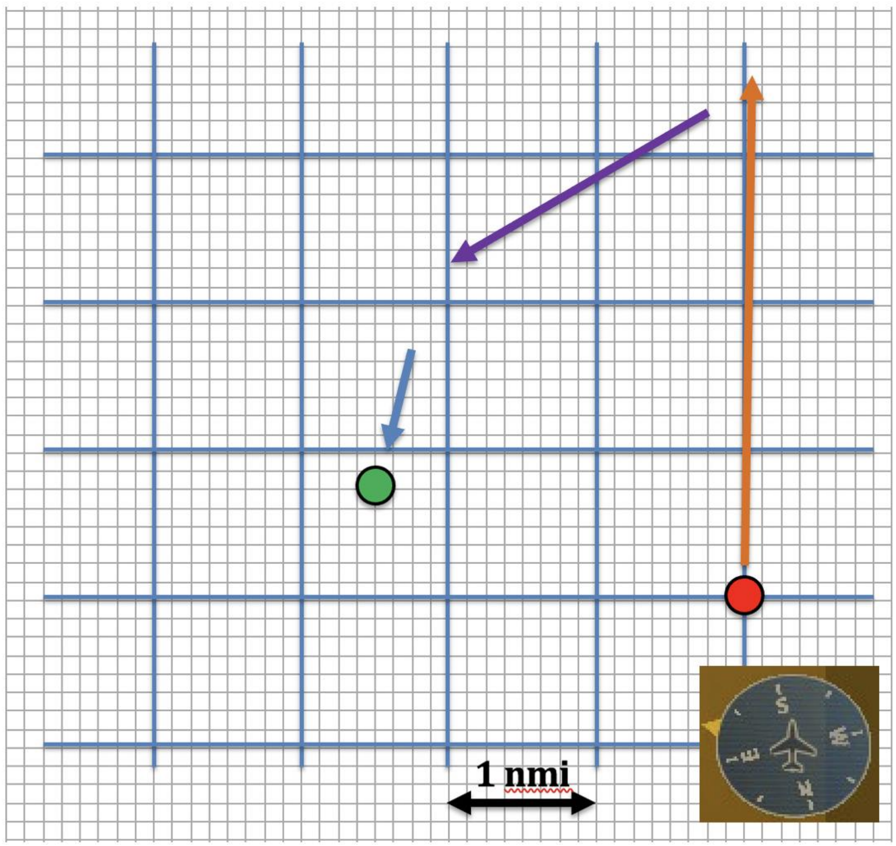
The blue lines are spaced on the paper by 1 inch and represent 1 nmi (nautical mile). There are about 73,000 inches in a nautical mile, 72,913.4 to be exact. Therefore, our map's scale is 1:73,000 (read "one to seventy-three thousand"). This is the ratio of 1 nmi on the map to 1 nmi.



The map tells us that we must turn left. But not now!
We need to take off first with the current heading which makes the aircraft location move up on the map.

If we turn after a little while directly East then we will intersect the straight line extended from the runway, which puts us in the perfect situation to align the flight path with the runway. Did you notice the large "3" on the runway? Runways all over the world have a number written on them, which is a tenth of the runway direction, rounded to the next integer. Refer to the compass rose handout.

This means the runway is in the direction of 30 degrees meaning 30 degrees East of North. This allows us to complete our map:



The blue arrow is pointing in the runway direction

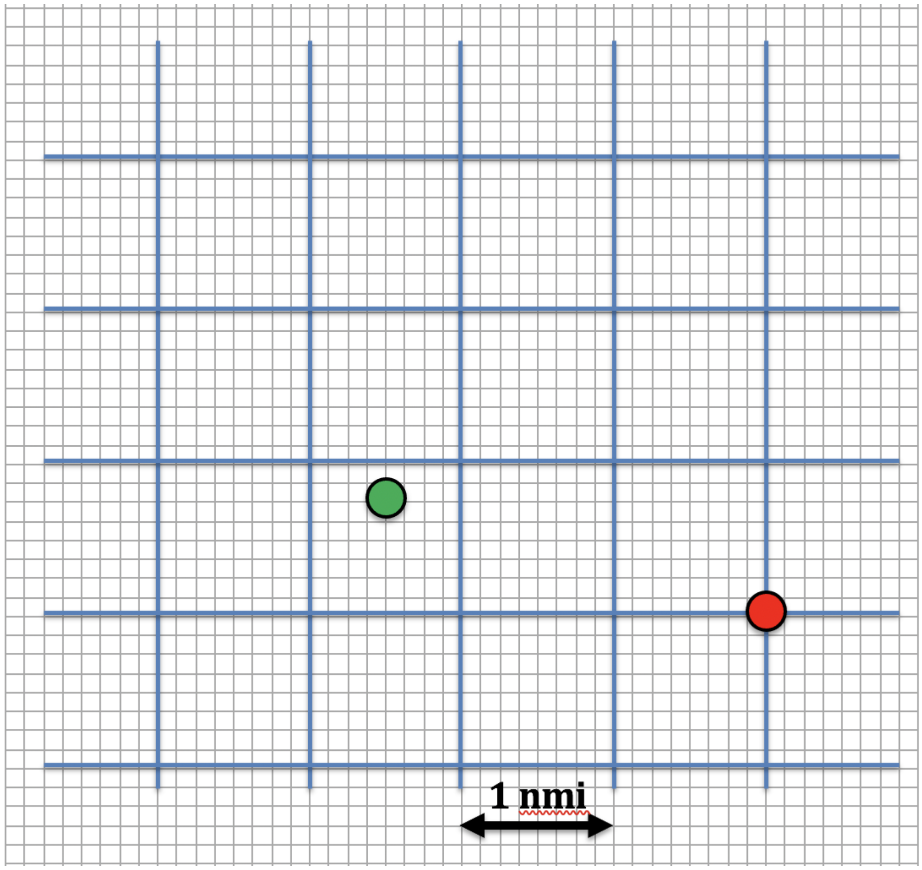
The orange arrow is pointing in the take-off direction

The magenta arrow is our flight direction for lining up with the runway

We notice to land we must fly a much greater total distance than just the distance to the destination.

Could you imagine how to plan for a shorter route?

If yes, use the blank map below to draw the course you want to fly … and fly it!



Now that you have planned out your flight and learned a bit more about the Piper Cub- it is time to try your second Midwest Fly-in.

You might want to time yourself during the flight to see how long it takes you from take-off to landing.

Think About It:

Were you more successful on your second attempt at the Midwest Fly-in?

What tips do you think helped you fly better this time?

Module 4: CoDrone Basics

The CoDrone Pro will be the first drone you will be flying during this program.

The CoDrone is a mini drone fighter:

- Manufactured by BYROBOT
- Model BR-PT-100
- Lithium polymer battery
- Drone fighter with missiles via infrared
- Automatic take-off and landing
- Automatic recovery flip
- PETRONE app on cell phone can function as a remote control
- Blockly, Python, Arduino code on a computer can function as a remote control.
- The CoDrone Pro package includes a Smart Inventor Board to be used as a remote control.

4.1 Unpacking the CoDrone Kit and Basic Maintenance

Locate your CoDrone package and see what’s inside.

Some parts are delicate- treat them carefully and make sure you do not lose even the tiniest of parts!

Here are some basic maintenance measures for best CoDrone performance:

(The following maintenance measures are from the Robolink Basecamp.)

The Battery:

The CoDrone uses a 3.7V, 300mAH, 15c Lithium-Ion Polymer (LiPo) battery.

To charge your CoDrone battery, slide it into the charger and then connect the charger to a USB port. If the battery is charging or idle, the charger will show two red LEDs. If the battery is fully charged, the LEDs will turn off.

Battery Maintenance
Follow these rules to make your battery last!

1. Avoid leaving the battery charging for longer than an hour. LiPo batteries are not meant to be fully charged for too long. If they're charged for extended periods, they will overheat and weaken.
2. Do not leave the battery inserted in the CoDrone when storing. This will drain the battery completely and will also weaken the battery or make it puffy.
3. When storing batteries for longer than 2 days, it is best to store them with a 50% charge.
4. Like a lot of other batteries, LiPo batteries are prone to heat and moisture damage. Keep your CoDrone battery safe at room temperature and in a dry place. Do not get your battery wet unless you want to destroy it!

5. Flight time from the battery will decrease in colder environments.
6. Make sure not to step on or puncture the LiPo battery.
7. Do not use a puffy LiPo battery. This can occur when you let the battery drain completely. You will need to purchase a replacement battery for your CoDrone.

Propellers

Your CoDrone has four propellers. Two spin clockwise, and the other two spin counterclockwise.

The reason for this involves some physics called gyroscopic precession. The opposing motors cancel out each other's net torque, or the sum of their rotational motion. If the propellers were all going in the same direction, your CoDrone would most likely be constantly yawing. If your CoDrone is tilting while trying to take off, there is a good chance your propeller placement has something to do with it!

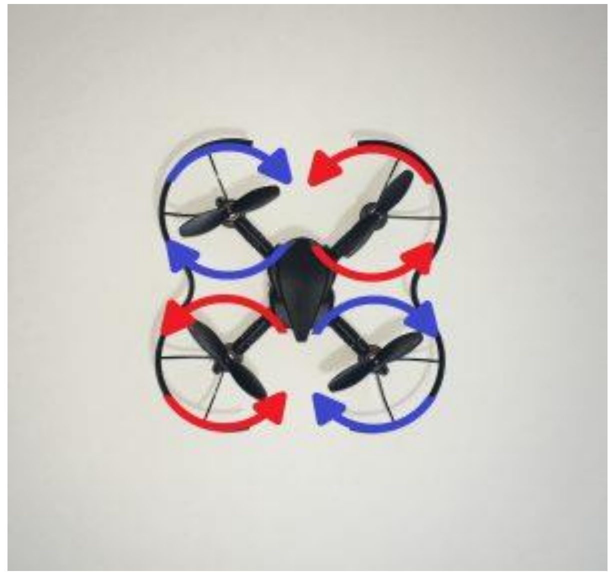
Propeller Maintenance
Propellers are the most vulnerable component on the CoDrone because they are moving much faster than anything they will crash into. Keep yours in top condition to avoid any problems!

1. Propellers are extremely aerodynamically sensitive. If your propeller has a crack or chip, it may cause your CoDrone to lean to one side or become unbalanced. Make sure to inspect and replace your propellers as often as possible. Replacement propellers are available on the Robolink website!
2. Make sure your propellers are clean. Avoid hair and fibers.
3. Although it's fun, try not to crash into stationary objects!

Replacing your propellers
The CoDrone propellers are designed to be easily pulled off, cleaned, and replaced.

1. To remove your propellers, pinch the motor with one hand and pull the propeller off with the other.
2. To replace a propeller, align the through hole with the shaft and squeeze the propeller back on.
3. Remember that propellers are directional and should be put in their correct placements!

Need a tip to remember how the propellers spin? Match the arrow direction up with the propeller guards! For example, the upper righthand motor should have a counterclockwise propeller attached since the propeller guard is pointing counterclockwise.



Motors

CoDrone motors are two-way DC motors and are controlled entirely by voltage. A DC motor will run faster with a large voltage, slower with a small voltage, and reversed with a negative voltage.

Motor Maintenance:
CoDrone motors are easy to detach from the main body — just pull them off (gently)! Inspect your motors often to clear away any obstructions or buildup.

1. Hair can easily get caught and wound in the shaft. Be sure to pull off your propellors and inspect the shafts of the motors carefully.
2. A bent shaft can cause a lot of problems. When adding and removing propellers, be careful not to push or pull too hard.

Body

The CoDrone body has a lot of sensors and circuits. Take good care of it so you don’t damage any of the onboard components!

Body Maintenance:

1. The base contains the majority of your CoDrone’s vital hardware. Keep it in a safe, dry place always. Don’t fly it outdoors or over wet ground.

2. The base also contains multiple infrared (IR) sensors. Keep these surfaces clean so your CoDrone works properly.
3. Keep the base away from any magnetic objects.
4. The top and bottom of the base separate easily when pulled apart. Use this to remove anything that might be stuck inside the body.

CoDrone Components (with added STEM SEALs Rescue Parts)

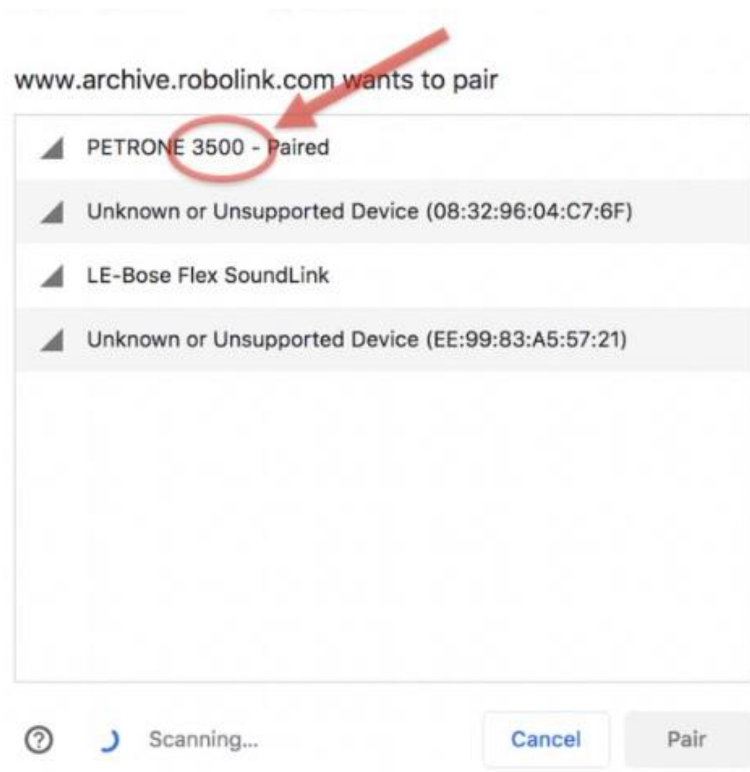
At this time, **you will only need the assembled drone, the lithium battery, the battery adapter, a USB cable, and a powered USB port (either on a computer or an AC power adapter).** Later in the week, we will use the other parts to create a remote control for the drone.

Before we begin an activity with the Codrone, we need to get the unique identification number for each drone.

Your computer communicates with the CoDrone via Bluetooth. Bluetooth technology is a widely used wireless communication method. For example, some earphones use wireless Bluetooth to connect and communicate with a smartphone.

We can use the computer to identify each of the many drones in the room, but we will need to do this process one at a time. Later when you need to pair your drone to the computer, you will see a pop-up window like this with available Bluetooth devices – you will need to look for your unique number. It will always start with PETRONE.

See below:



In this example, the drone we are identifying ins "PETRONE 3500". Later you will notice there will be multiple drones listed when you go to pair your CoDrone- so it is important you know your number. This number will not change so write it in the back of your Lectionary on the page provided. You will also be given a small round sticker to write it on and place on the bottom of your drone where the battery snaps in.

To identify each individual drone, we will go ONE by ONE inserting a battery and identifying each drone. Only insert your battery when told to do so.

Write your identification number on a small round sticker and place it on the underside of the drone.

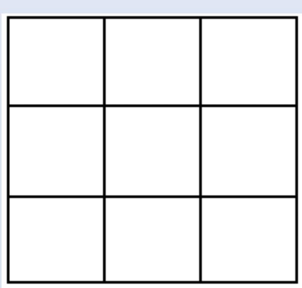
4.2 Flight 1, 2, 3: Taking Off and Landing Autonomously (Sensor Calibration)

We will first use the built-in sensor calibration feature to take off and land the drone autonomously. The sensor calibration is an important feature to remember. **You should always calibrate your drone before flying it during a practice session or anytime throughout flying if the drone drifts during or takes a particularly rough crash or landing.** Follow the instructions given by your teacher or watch this short video on how to calibrate your CoDrone. https://youtu.be/WIbfCFp8g7c

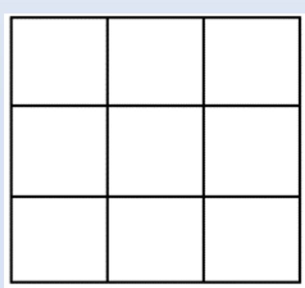
Try this: Flight 1 - Drone Calibration

1. Turn the drone on by inserting the battery. This is how your turn the drone on and off. When not in use always disconnect the battery. When the drone first comes on the eyes should be solid red, and the LEDs on the arms should be solid red. The two LEDs in the rear of the drone should be blinking green twice.
2. Pick the drone up by the body using your thumb and forefinger, flip it upside down, and push and hold the small black button on the left (port) side of the drone. Continue to hold the button in - the arm LEDs should turn white and eventually flash white. When the lights flash white, flip the drone back over and place it on the center of the landing target and wait.

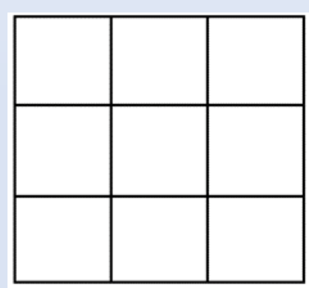
3. The LEDs will flash green and then red to let you know it will begin the takeoff. **Caution do not touch the drone at this point.**
4. The drone should fly a calibration procedure where it will take off, hover, and then land. Notice that the drone may drift (left, right, forward, or backward) and then try to correct this drift.
5. Measure how far the drone drifts (in cm) from the center of the landing pad (target).
6. Enter the value in the diagram provided by writing the measurement in the square representing the direction the drone traveled using the center square as the take off point.
7. Repeat the calibration procedure two more times by disconnecting the battery and reinserting and following steps 2 – 6.



Flight 1

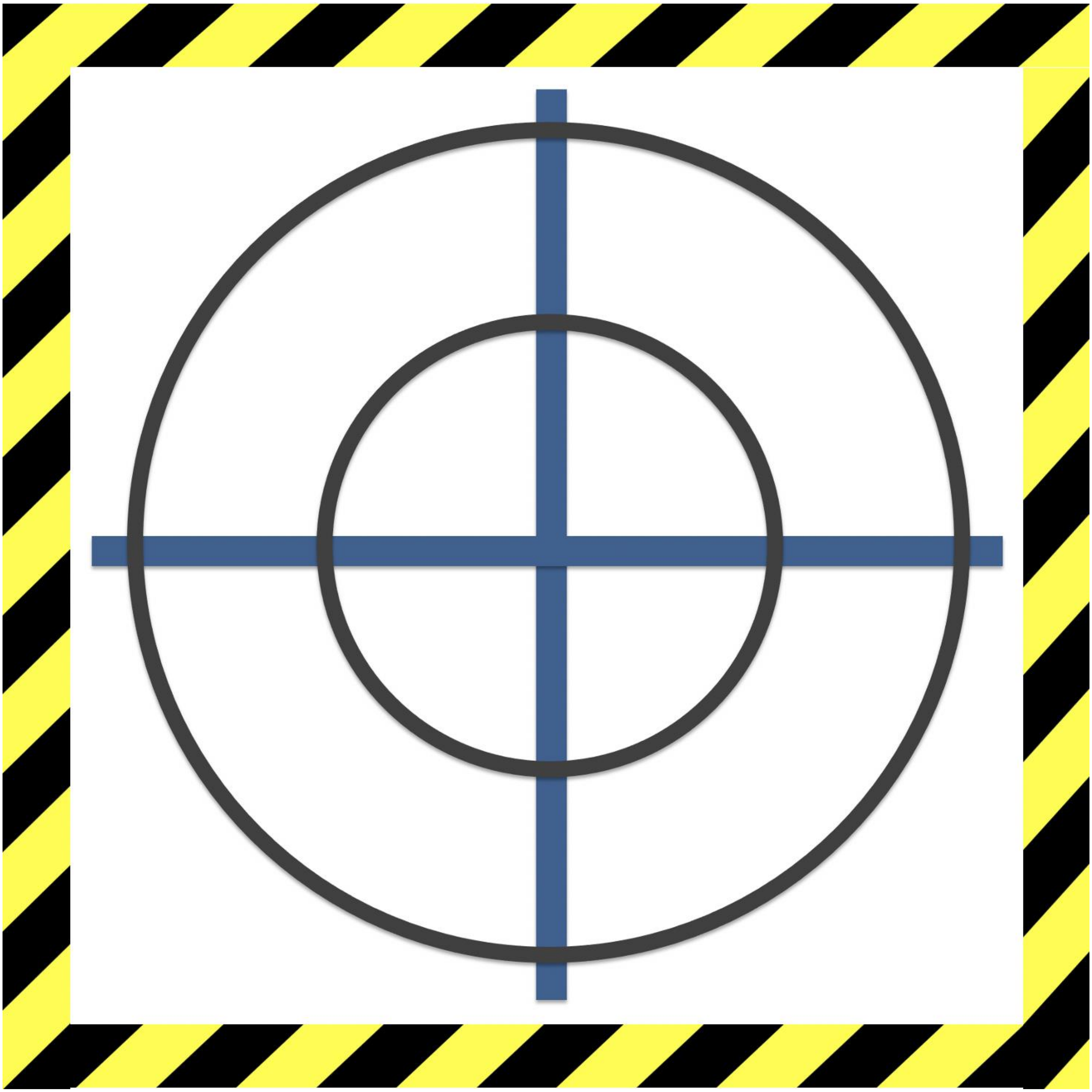


Flight 2



Flight 3

Drone Landing Target



4.3 Flight 4: Using Blockly to Take-off and Land

The calibration procedure is the only feature that the drone can perform without some outside source or device being used to send commands to the drone. Through computer programming or code, you can write commands that the drone can receive and perform. Blockly is a visual code editor platform that uses interlocking, graphical blocks to represent code concepts such as flight commands, variables, logical expressions, loops, and more.

Let's use Blockly to write a simple take-off and landing commands for the CoDrone.

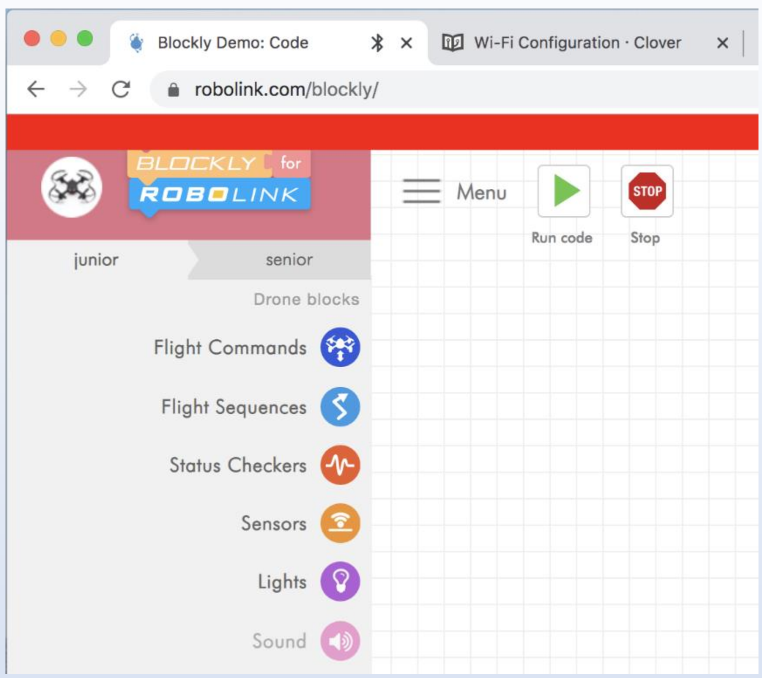
Try This: Flight 4- Take-off and Land Using Blocky

1. Login to a PC (Username: reset Password: Welcome1)

2. Start the **Google Chrome browser**.

3. Go to https://www.robolink.com/blockly/ or search "robolink blockly" (which is a quick way to locate the page).

4. Once on the webpage, you first must choose which drone you are using – choose CoDrone Pro. The webpage should look like this:



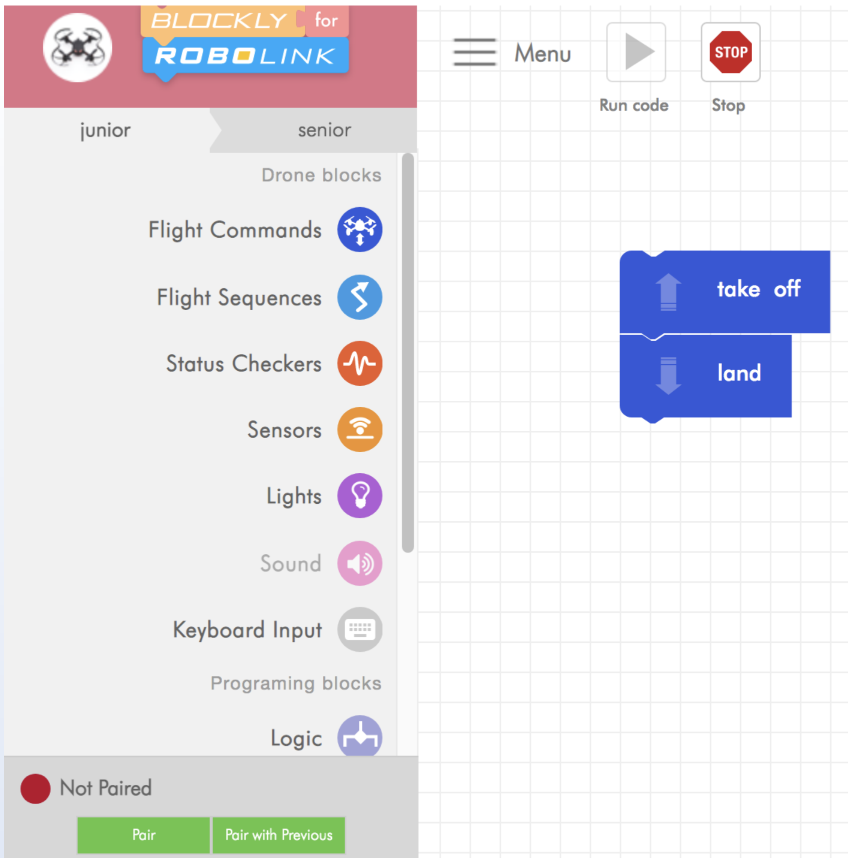
5. On the left side of the page are two tabs: **junior** and **senior** with slight variations of CoDrone commands. The "Flight Commands", "Flight Sequences", "Status Checkers", etc. are the toolboxes from which you will get the commands for the drone. You will drag and drop the various blocks onto the gridded workspace to create strings of commands for the drone.

6. On the right side is the workspace or grid of squares. At the top of the workspace is a menu, a button to run the code, and to stop.

Try This: Flight 4- Take-off and Land Using Blocky (continued)

7. Click on "Flight Commands" (junior) and drag and drop the "take off" block into the gridded workspace.

8. Click on "Flight Commands" again and this time drag drop the "land" command under the "take off" block. they should lock together like puzzle pieces. Your code should now look like this:



9. In the lower left corner – there is a red button indicating "not paired". This means that while you have written some code, the PC does not yet know where to send the code or which drone should be controlled by the code. Let's pair the PC with your drone.

10. Turn your CoDrone on – insert the battery. Remember the eye and arm LEDs should be red and the two rear LEDs should be green.

11. Click the green "Pair" button on the computer. A small popup window will appear saying "www.robolink.com wants to pair" as well as a list of nearby Bluetooth devices (which and how many depends on the number of devices near you). Look for a device named PETRONE followed by the number of your drone ID. Be sure to always pair to your drone's ID number and not someone else's!

Try This: Flight 4- Take-off and Landing Using Blocky (continued)

12. Select your drone in the window and then click the “pair” button. The rear LEDs on your drone should now be solid green and not blinking. This means that your drone is now paired with the computer and accepting its code.

13. Set your drone on the landing target as before.

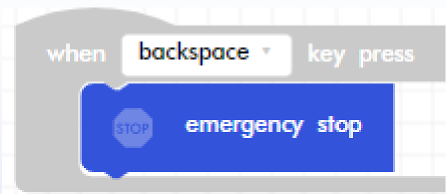
14. Click the green “Run code” button at the top left of the workspace. The drone should take off and land.

Did your drone take off and land?

While it appears that we accomplished the same task as the calibration procedure in a more complicated way- that is not quite correct. You can now use Blockly to create much more complex commands for the drone to follow.

Note: Emergency Landing

Notice there is a “Stop” button on the Blockly workspace. This button can be used to stop code from running but in an emergency where you may wish to land the drone immediately, it is recommended to always include the following in your code:



The “when key pressed” block may be found in the “keyboard input” toolbox. You may select which key to use in the dropdown menu in the block. The “emergency stop” block is found in the “flight command” toolbox.

These two blocks can be placed in the workspace separate from your other code. In an emergency such as a bad crash is eminent, or the drone is not performing correctly – you can immediately get your drone to land by pressing the designated key on your computer such as the backspace bar or the down key. You can select which key you want to use from the drop-down menu.

4.4 Flight 5: CoDrone Yaw Calibration

Now that you have learned a bit about using Blockly to create code commands for the drone and how to pair the drone, let's write some more complex code to turn the drone around and land.

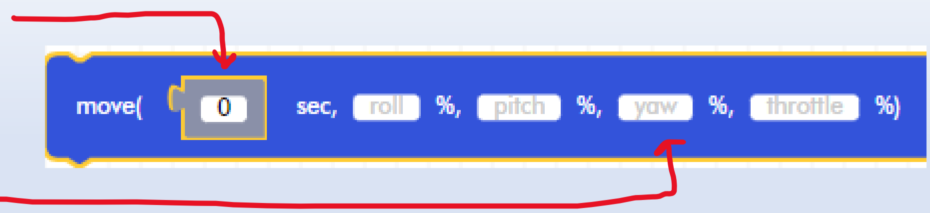
There are several ways in which you can turn an aircraft:

- **Yaw** is more like a spin.
- **Yaw** and **pitch** together allow arcs.
- A **roll** does not actually turn the drone – it allows the drone to fly sideways.
- We can use all three: **roll**, **pitch**, and **yaw** to turn the aircraft but this is rather complex.

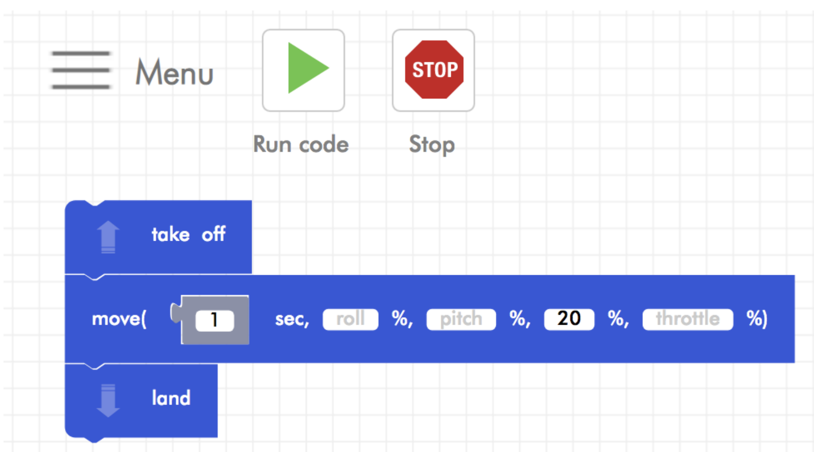
Time to Code: CoDrone Yaw Calibration

1. Open the Blockly editor again – your previous code should still be what is in the workspace. We will add to the previous code.

2. Find the "move (0 sec, roll%, pitch %, yaw %, throttle %" block from the senior "Flight commands toolbox and place it between the "take off" and "land" blocks.
   - Change the zero for time to 1 sec.
   - Enter a 20 % into the field for yaw.

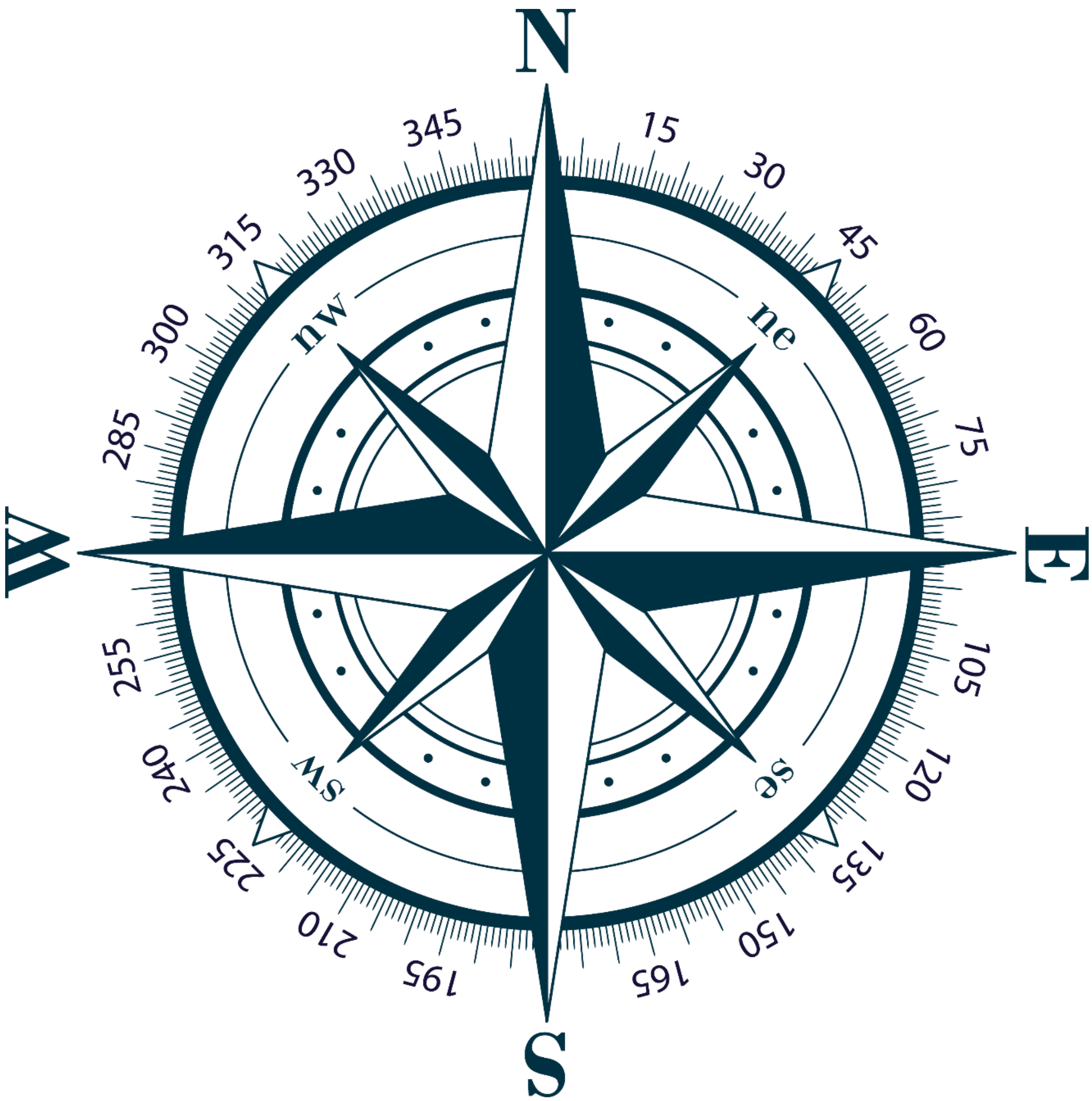


Your finished code should look like this:



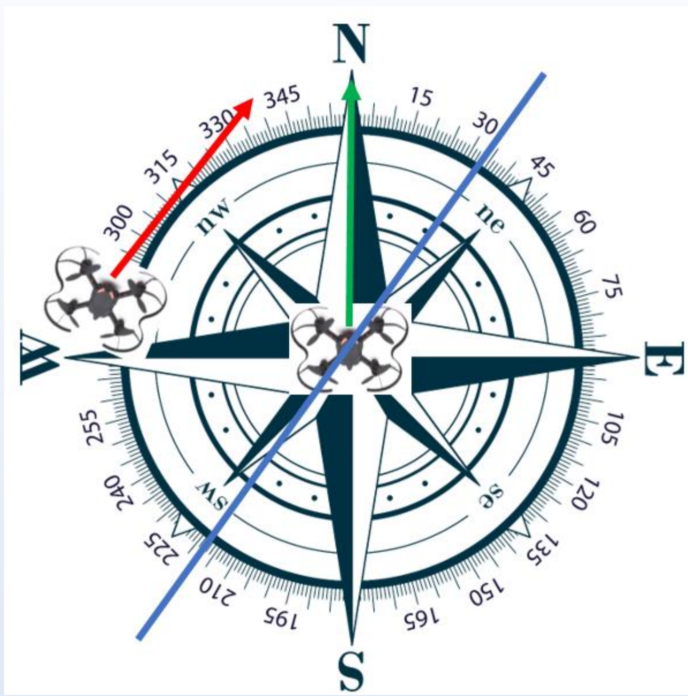
Now let's test out this code. We will use a compass rose like the one on this page to help us determine how to far the drone turns. This compass rose has four cardinal directions (North, East, South, and West) as well a full protractor circle which will allow you to measure how far the drone turns in angle measurements.

The image shows both cardinal directions and angles. North corresponds to 0 º or 360 º, East to 90 º, South to 180 º, and West to 270 º. The ordinal directions are halfway in between. E.g., SE would be 135 º. Obviously, any direction in degrees can be found on the scale of the angles; cardinal and ordinal directions are only special selected and important directions.



Try This: Yaw Calibration Test

1. Insert the battery into your drone and pair the drone with your PC.
2. Place your drone on the compass rose facing North.
3. Make sure your flight path is clear.
4. Run the Yaw Calibration code by clicking the green "run code" button in the Blockly editor.
5. When the drone lands- use the angle indicators on the compass rose to estimate the degree turn that the drone performed.
6. Note: The drone rarely lands exactly in the middle of the compass rose. Therefore, to estimate the angle correctly, use a meter stick to make a straight line along the longitudinal axis of the drone and then shift this line until the compass ross intersects the line. See the diagram.



The green arrow indicates the direction of the drone before takeoff (North). This is zero degrees

The red arrow points in the direction that the drone is facing after landing.

The blue line is parallel to the red arrow and passes through the center of the compass rose/protractor which allows you to read the degree of turn.

7. Read the angle from the protractor where the blue line intersects the circle. Notice that the line intersects the protractor at two points the correct one would be 33 degrees in the above example, 217 degrees is where the rear of the drone is facing. Your angle might be different than the example. Enter the angle in **Table 1** on page 58.

Try This: Yaw Calibration Test (continued)

8. In the Blockly editor, change the Yaw percentage from 20 % to 30 %.

9. Test the new yaw percentage with the drone and enter the data (angle measurement) into **Table 1**.

10. Continue to test the yaw percentages from 20 % to 100 % and add the data into **Table 1** on page 58.

- **What did you notice when you increased the yaw percentage?**

You cannot enter larger percentages than 100 for yaw. Try it and see what happens.

Does this mean that that is as far as the drone can turn (100 % yaw)? Of course not!

11. Look at the data in **Table 1**.
Can you take that information and graph it on the blank graph below **Table 1**?

There is a sample data table and graph on Page 59 to help you understand how to complete the requested information.

We can also affect how the far the drone turns by increasing the time in seconds in the "move" block.

12. Change the percentage yaw back to 20 % and test the drone again (1 sec and 20% yaw). You should have a very similar measurement from above.

13. Enter this new measurement into **Table 2** on page 60.

14. Complete **Table 2** on page 60 by keeping the yaw percentage the same (20 %) and changing the time in the increments listed in the table.

- **What did you notice about the turns of the drone this time?**
- **Was your drone able to make a complete circle? If so, at what time increment?**
- **Did your drone make more than one complete circle?**

Note: If your drone traveled past a full 360 º turn you will need to add the extra degrees to 360 º to get the correct measurement. (Example: At 14 seconds if your drone traveled 130 º as read from the protractor, you will need to add 360 º to the reading. 360 º + 130 º = 490 º)

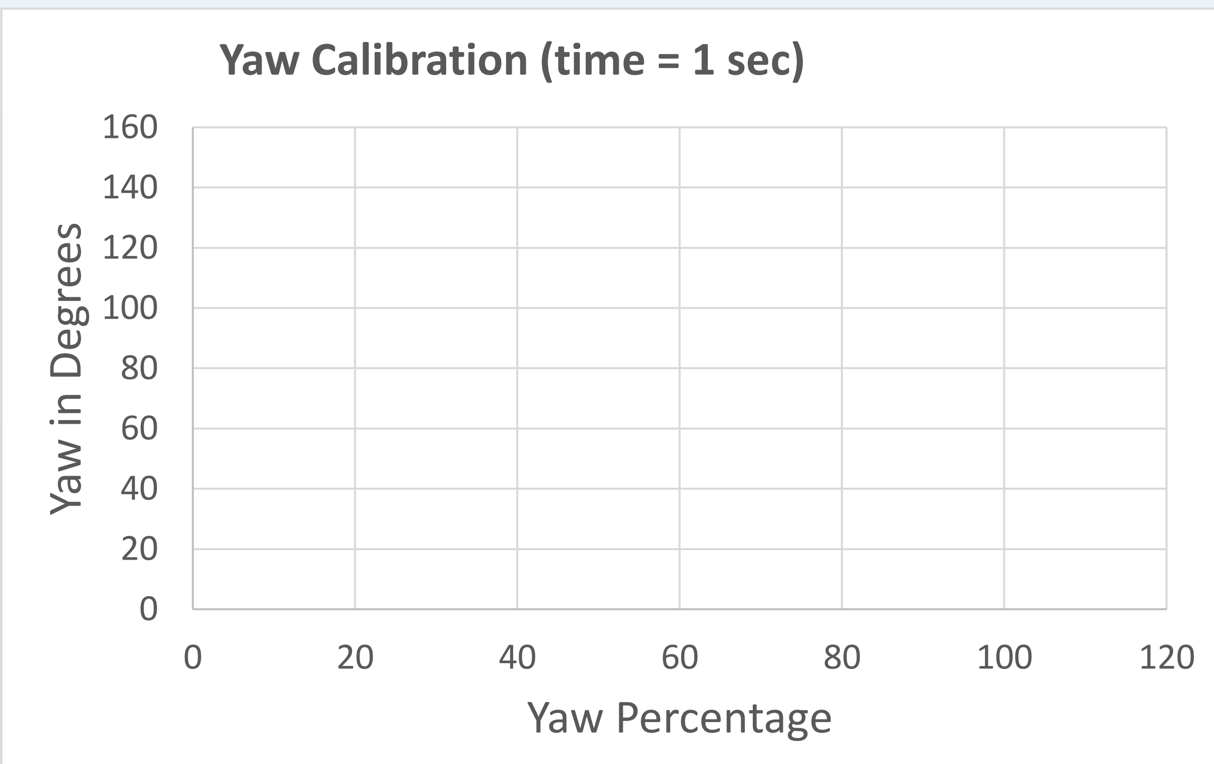
- Let's graph your results in a graph under **Table 2**.
- Look at the sample data and graph on page 61.

Try This: Yaw Calibration Test (continued)

Record your data from the Yaw Calibration (time = 1 second) here.

Table 1: Yaw Calibration (time = 1 second)

move time / sec	yaw / %	yaw / degrees
1	20	
1	30	
1	40	
1	50	
1	60	
1	70	
1	80	
1	90	
1	100	

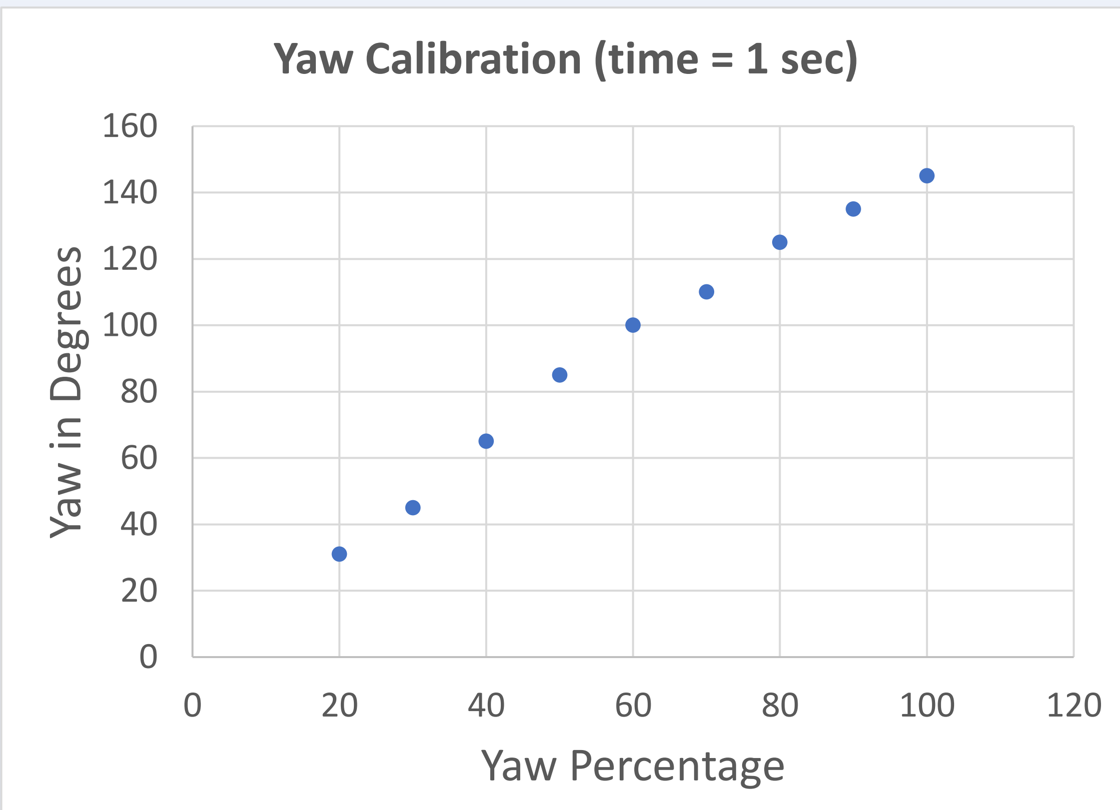


Try This: Yaw Calibration Test (continued)

Here is a sample table and graph from a Yaw Calibration (time = 1 second) test.

Yaw Calibration (time = 1 sec)

move time / sec	yaw / %	yaw / degrees
1	20	31
1	30	45
1	40	65
1	50	85
1	60	100
1	70	110
1	80	125
1	90	135
1	100	145

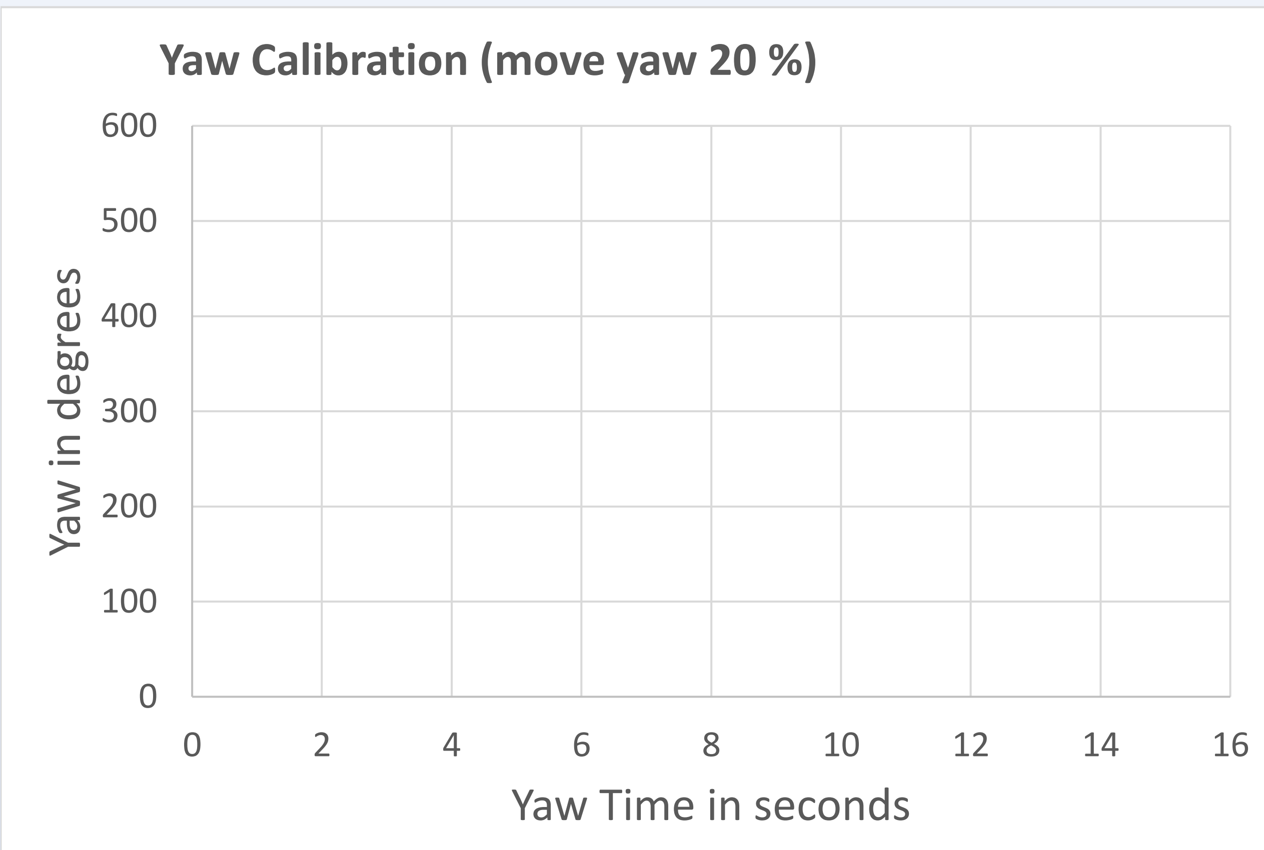


Try This: Yaw Calibration Test (continued)

Record your data from the Yaw Calibration (yaw = 20 %) here.

Table 2: Yaw Calibration (move yaw 20 %)

move time / sec	yaw / %	yaw / degrees
1	20	
1.5	20	
2	20	
2.5	20	
3	20	
4	20	
7	20	
10	20	
12	20	
14	20	

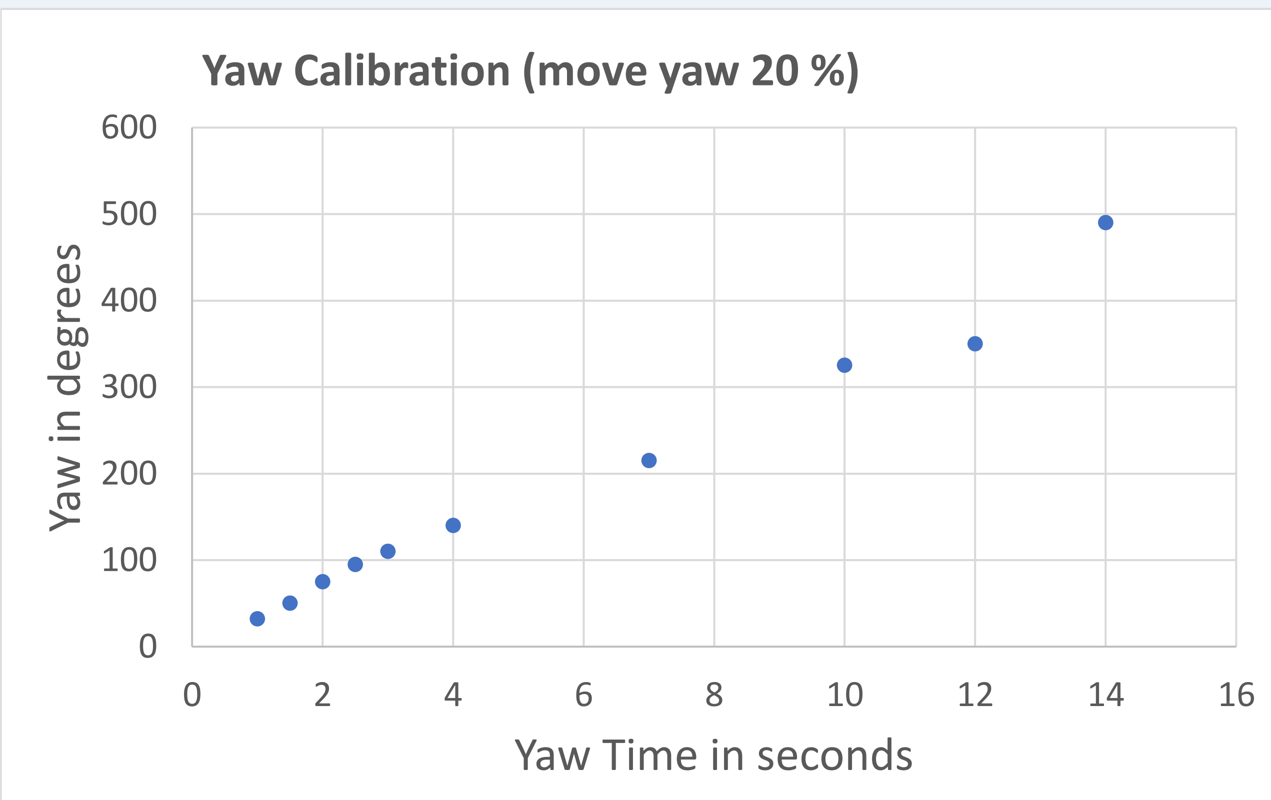


Try This: Yaw Calibration Test (continued)

Here is a sample table and graph from a Yaw Calibration (yaw = 20 %) test.

Yaw Calibration (move yaw 20 %)

move time / sec	yaw / %	yaw / degrees
1	20	32
1.5	20	50
2	20	75
2.5	20	95
3	20	110
4	20	140
7	20	215
10	20	325
12	20	350
14	20	490



Try This: Yaw Calibration Test (continued)

Look at the graphs you created and compare them to the sample graphs:

- **The blue dots on the graph are almost aligned in a straight line, are yours also?**
- **Is your line identical to the sample graph? What is a possible explanation for this?**

Can you envision doing a similar calibration as the one above but with a larger yaw percentage?

- **Where would you expect the dots to be in the above graph?**

Check out the optional activity to find out!

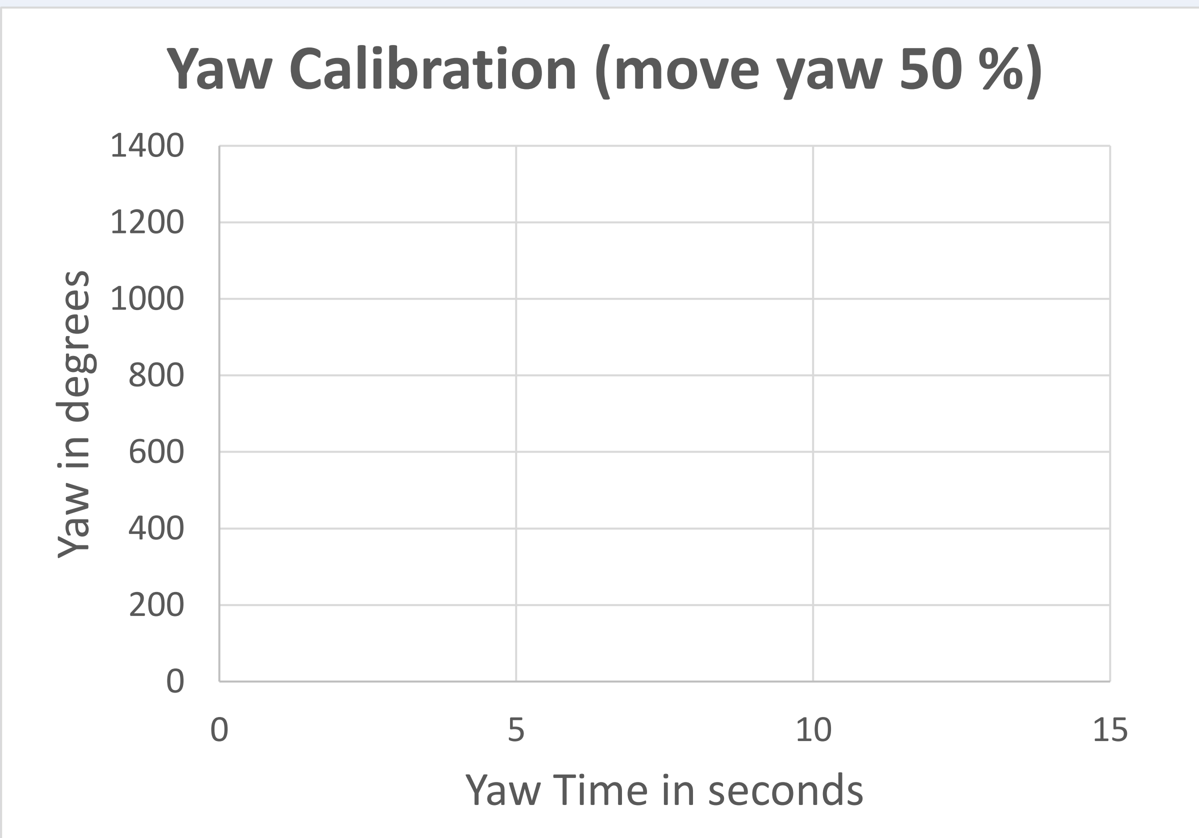
Try This: Yaw Calibration Test (Optional)

Change the yaw percentage to 50 % and then use the same series of different times as above.

Record your data measurements on **Table 3** below:

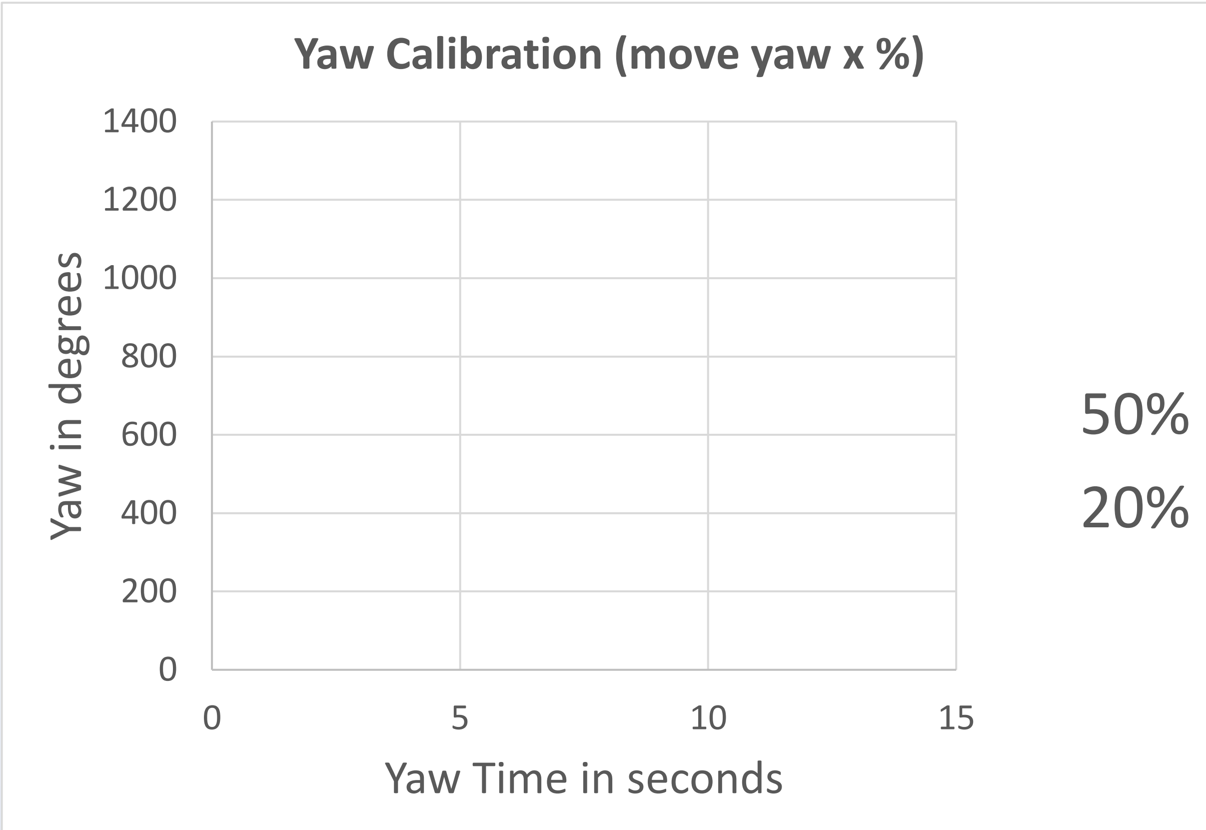
move time / sec	yaw / %	yaw / degrees
1	50	
1.5	50	
2	50	
2.5	50	
3	50	
4	50	
7	50	
10	50	
12	50	
14	50	

Are your data different that the 20 % test? Enter the data into the graph below. Can you still draw a straight line close to all the data points?



Try This: Yaw Calibration Test (Optional)

Make a graph (with the data you measured), which includes both, 20 % and 50 % yaw data. The blank graph for this is below:



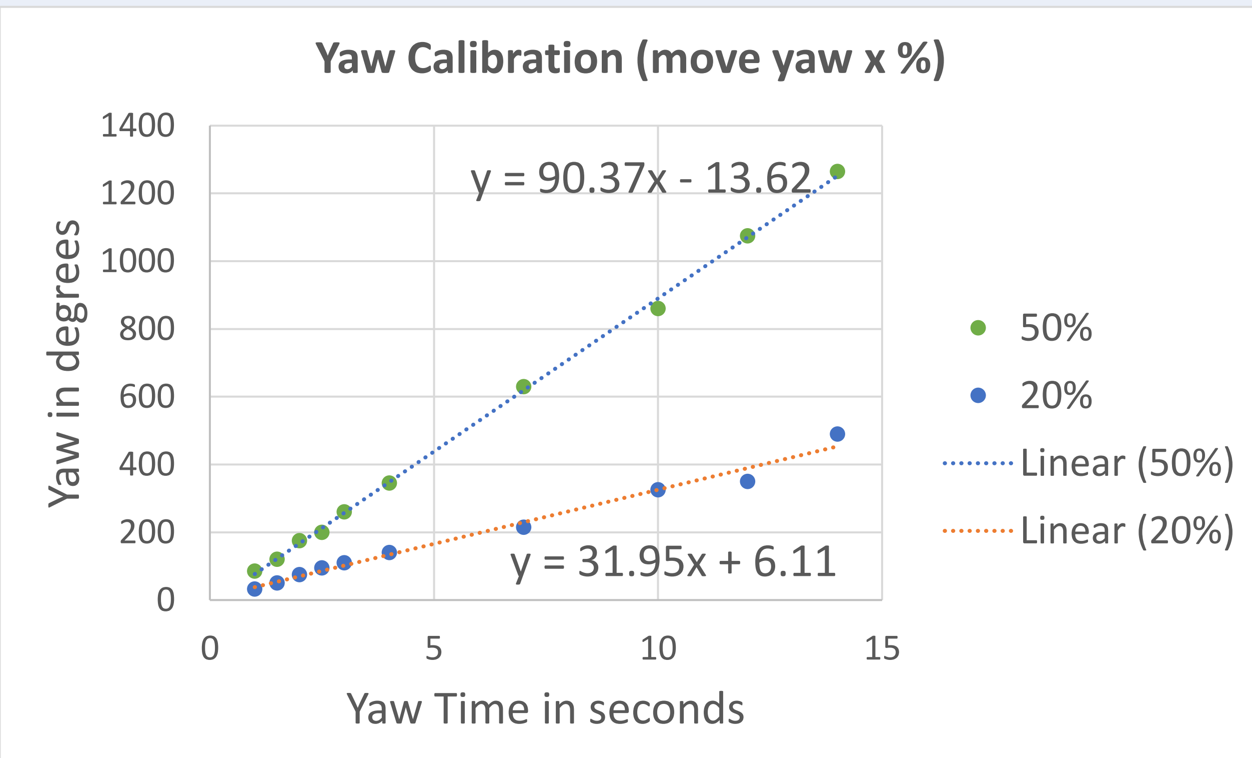
- **Do both set of data show straight lines?**
- **Can you find the slope of the two lines? What does this represent (slope)?**
- **Can you use this information to predict how far the drone might move at 5 seconds with 20 % yaw?**

Try This: Yaw Calibration Test (Optional)

Have a look at the sample data and graph:

Yaw Calibration (move yaw 50 %)

move time / sec	yaw / %	yaw / degrees
1	50	85
1.5	50	120
2	50	175
2.5	50	200
3	50	260
4	50	345
7	50	630
10	50	860
12	50	1075
14	50	1265



Try This: Yaw Calibration Test (Optional)

- Notice that the graph contains two dotted straight lines.
- Each straight line is the result from a linear approximation, given by the equation $y = mx + b$ where m is the slope and b the y-intercept.
- Can you see that both slopes are different, but both y-intercepts are very small?
- Did you learn that any slope is rise over run?

$$slope = \frac{rise}{run}$$

Optional Evaluation of Slopes of Lines

- Since the rise (along the vertical axis) is yaw in degrees, and the run (along the horizontal axis) is yaw time in seconds), the two slopes in the example graph are approximately

$$slope\ (at\ 20\ \%\ yaw) = \frac{32\ degrees}{1\ sec}$$

or simply 32 degrees per second; and

$$slope\ (at\ 50\ \%\ yaw) = \frac{90\ degrees}{1\ sec}$$

or simply 90 degrees per second.

- Then, using algebra we could compute any point of each graph:
- E.g., we did not measure yaw for a time of 5 sec. Can you compute

$$yaw(5\ \text{sec}, 20\ \%) = 32\ \frac{deg}{sec} \cdot 5\ sec = \_\_\_\_\_\_\_\_\_\_\_$$

and

$$yaw(5\ \text{sec}, 50\ \%) = 90\ \frac{deg}{sec} \cdot 5\ sec = \_\_\_\_\_\_\_\_\_\_\_$$

(Note: read the above as “yaw of 5 seconds and at 50 % yaw percentage is …”. If you learned about functions in school already, then you will remember that *f(x)* is read as “f as a function of x”, or short “f of x”.)

- Check if those two values you computed would fit into the graph shown above. Draw them in the graph as additional points.

Try This: Yaw Calibration Test (Optional)

- We also could use our first graph, yaw (in degrees) as function of yaw percentage at a fixed time of one second with a slope of

$$slope\ (at\ 1\ sec) = \frac{1.5\ degrees}{1\ \%}$$

- Then we could compute all points on the straight line in the first graph (again: $y = mx + b$) as

$$yaw(x, 1\ \text{sec}) = 1.5\ \frac{deg}{\%} \cdot x$$

- Let's check this, too, and compute the missing point at x = 10 %

$$yaw(10\ \%, 1\ \text{sec}) = 1.5\ \frac{deg}{\%} \cdot 10\ \% = \_\_\_\_\_\_\_\_$$

- Compute this value and enter the point in the original graph. Does it fit?

- The actual numbers for your drone may be different. But instead of not knowing anything about how different numbers for time and yaw percentage may work out, it is better to have an approximate idea of what the effect might be.

Optional Challenge to Make a Yaw Model

- With the data we have so far, we can go one step further: The equation under 39.) provides us with an estimate for the yaw in degrees for any yaw percentage, given we have the time at 1 sec in the move () block.

- Additionally, we also know that we need to multiply by the time in seconds to get the yaw in degrees when there is another number for time in the move () block. Putting this together, we have

$$yaw(x, t) = 1.5\ \frac{deg}{\%\ sec} \cdot x\ \cdot t$$

where *x* is the yaw percentage, and *t* is the time that percentage is applied.

- Test this equation for several of the values in any of the tables. The results likely are not exactly what you measured for three reasons:
- The measurements are not exact.
- The drone does not always the same thing (for example when the batteries become weak, you would expect less action).
- The model is only a model. It computes exact values, which are only approximately what you would expect.

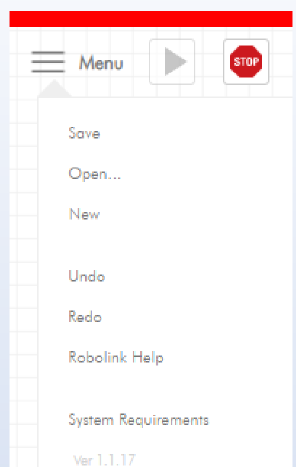
4.5 Flight 6: CoDrone Throttle Calibration

Now that you have learned how to control the yaw or the turning of the CoDrone, let's practice how to control another important movement of the drone- throttle.

The **throttle** controls the CoDrone's vertical (up and down) movement. A positive throttle will make the CoDrone fly higher, and a negative throttle will make the drone fly lower. Understanding and knowing how your drone functions will help you later to program more precisely when running a course with the drone.

Note: Saving and Creating New Programs in Blockly

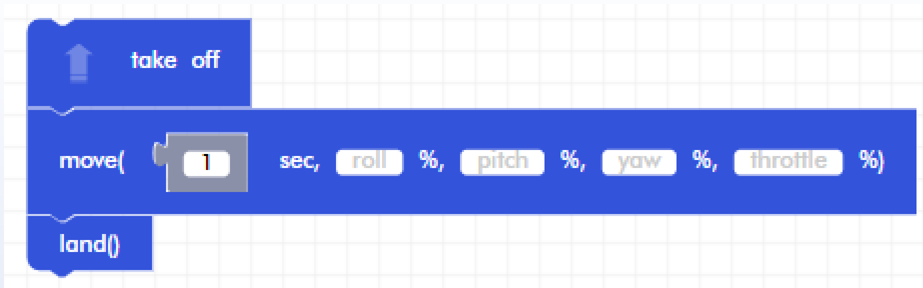
1. Open the Blockly editor.
2. Notice that the last code you wrote for the Yaw Calibration is still in the editor workspace. This code will always be on the workspace until you change it or delete it and create something else.
3. If the code is something you want to keep and reference later, you should save the code before changing it or deleting it.
4. For the AIR Challenge, you have been provided with a USB flash drive to save all your programing. This is important in case you do not work on the same computer; you will always have access to your code.
5. Click on the 3 bars next to "menu" in the workspace.
6. Insert your USB flash drive into the PC.
7. To save the "Yaw Calibration" program, click on save.
8. A pop-up window of your files will appear to ask you where you would like to save the file – choose the location of the USB drive (usually E:).
9. It will also ask you to name the file- do so accordingly.
10. The file will be saved as an ".xml" file. So, in this case it should look like this "Yaw Calibration.xml".
11. After saving the file, you can now click on "new" to start a program.
12. Notice that the old program is still in the workspace. You will need to drag and drop it in the trash can at the bottom right of the workspace to clear the workspace.



However, you will not need to delete the previous code as you can easily modify it for next coding exercise.

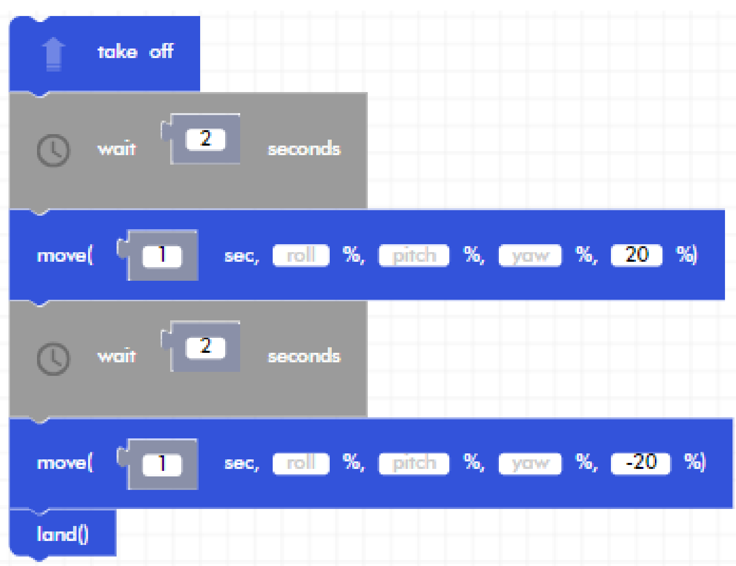
Time to Code: Throttle Calibration

1. Open the Blockly editor.
2. You can use the same blocks from the "Yaw Calibration" program.



3. Test this code as it is written above with (1 sec and 0 % throttle). The drone should hover for one second and then land. If you run this code a couple of times, you should note that the drone always takes off and hovers at an approximately the same height each time.
4. Add two "wait" blocks from the timing toolbox. Place one before the move block and one after.
   a. Change the time to "2 secs".
5. Change the throttle setting from 0 % to 20 %.
6. Add a second "move" block after the second "wait" block.
   a. Set it to 1 second and negative 20 % throttle (-20).
   b. This will help make the landing smoother but will not affect the measurement.

The complete code should look like this:



Try This: Throttle Calibration Test

1. Use the “Throttle Calibration” code.

2. Insert the battery into your drone and pair the drone with your PC.

3. You will need to hold a meter stick vertically near the drone for this test. Work in pairs to make this easier and try taping the meter stick to a water bottle.

4. Run the code on the drone and measure the height (in millimeters) of the drone at two points: initial take -off and the height after it climbs with 20 % throttle.

5. Enter the values in **Table 4**.

6. Change the two throttle values in increments of 20 (positive and negative) and measure.
- The next measurement will be at 40 % and -40 % respectively.

7. Continue increasing the throttle value while keeping the climbing time the same and complete **Table 4**.

8. **Compute the difference between the two heights** (height after climb minus height after take-off) and enter the value into the last column of **Table 4** on page 72**.**

9. Make a diagram (graph) of the difference of the heights, the climb amount.
- Analyze the graph: **what do you notice**?
- Look at the sample data and graph on page 73. Does yours look similar?

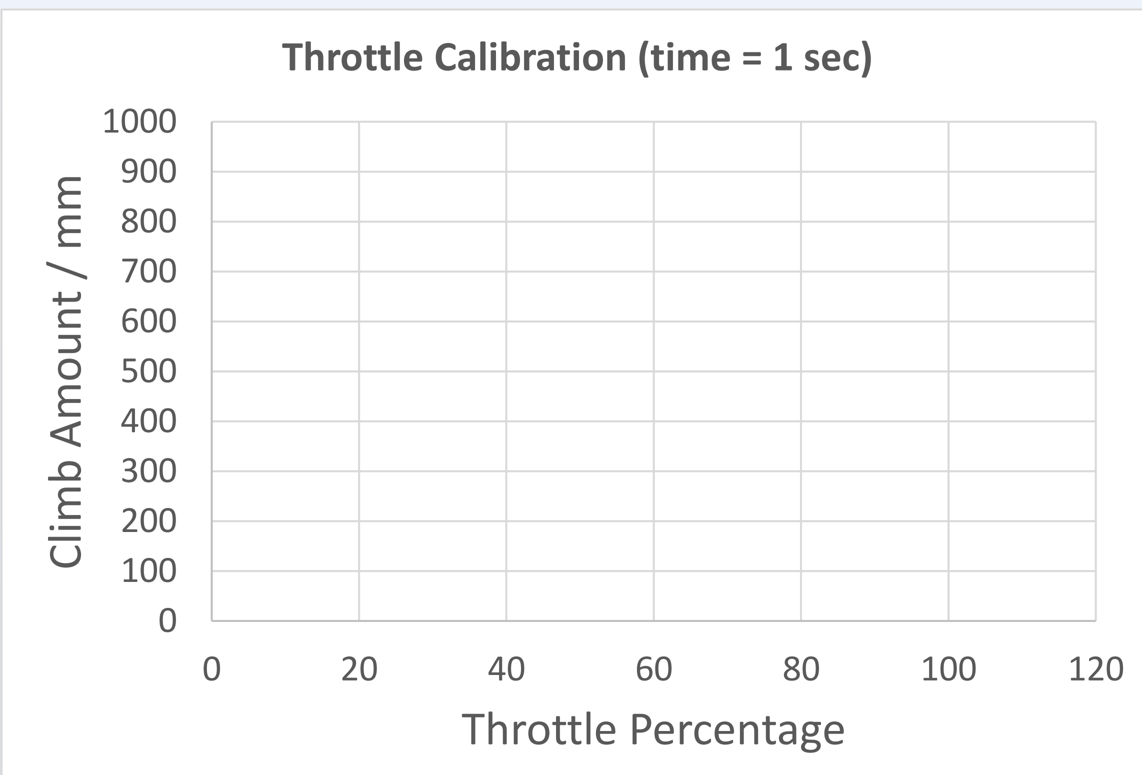
10. Let’s make a second table for a two second climb.
- Note, we stop at 60 %. Any value larger may result in your drone hitting the ceiling!
- Use **Table 5** on page 74 to record the data.
- Graph your results for this data.

11. Let’s compare the throttle Calibration test for 1 second with the 2 second data.
- Use the graph on page 76.
- Plot the data for the 1 second and 2 second test on the same graph.
- It might be helpful to use “x’s” for one set of data points and “dots” for the other so you can distinguish them apart.
- See the sample data and graph on page 76.

Try This: Throttle Calibration Test

Table 4: Throttle Calibration Test (time = 1 second)

Thrust Time / sec	Throttle / %	Height after take-off / mm	Height after climb / mm	Climb amount/mm
1	0			
1	20			
1	40			
1	60			
1	80			
1	100			



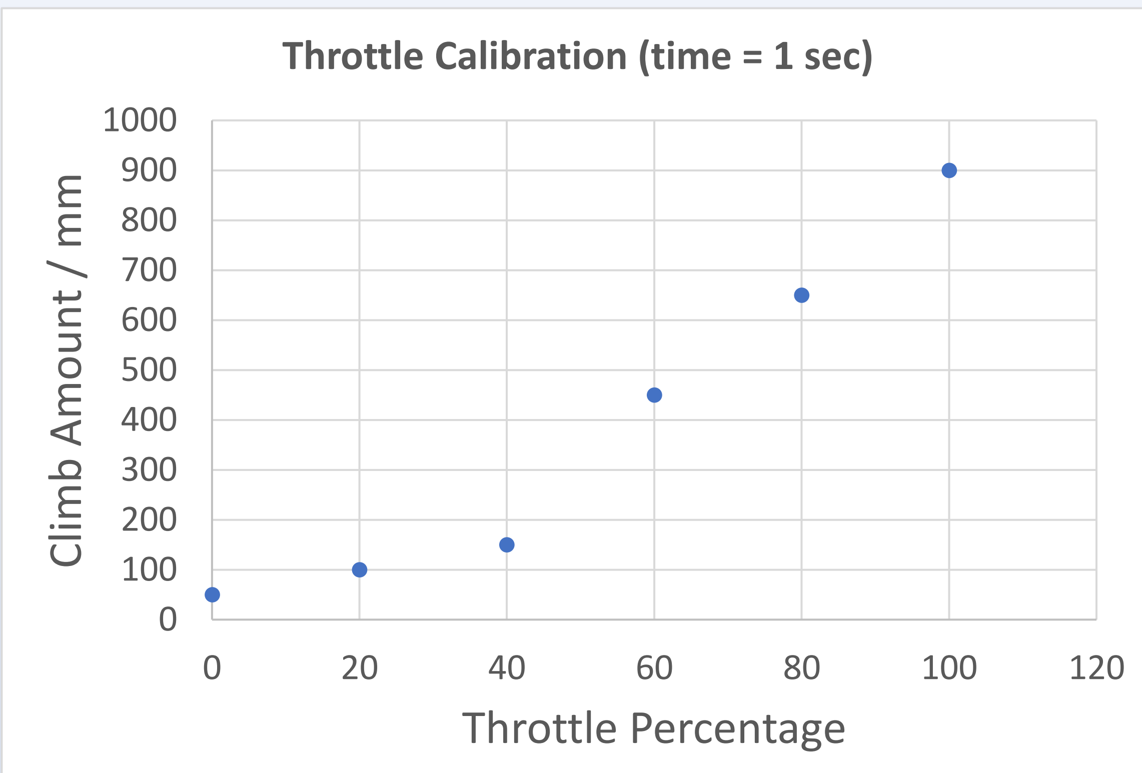
You should see the increase of how high the drone climbs at different throttle settings. At percentages larger than 50 a straight line could be drawn through the points. At smaller percentages the points deviate noticeably from that straight line.

Try This: Throttle Calibration Test

Here is a sample Throttle Calibration Test (Data and Graph)

Throttle Calibration (time = 1 sec)

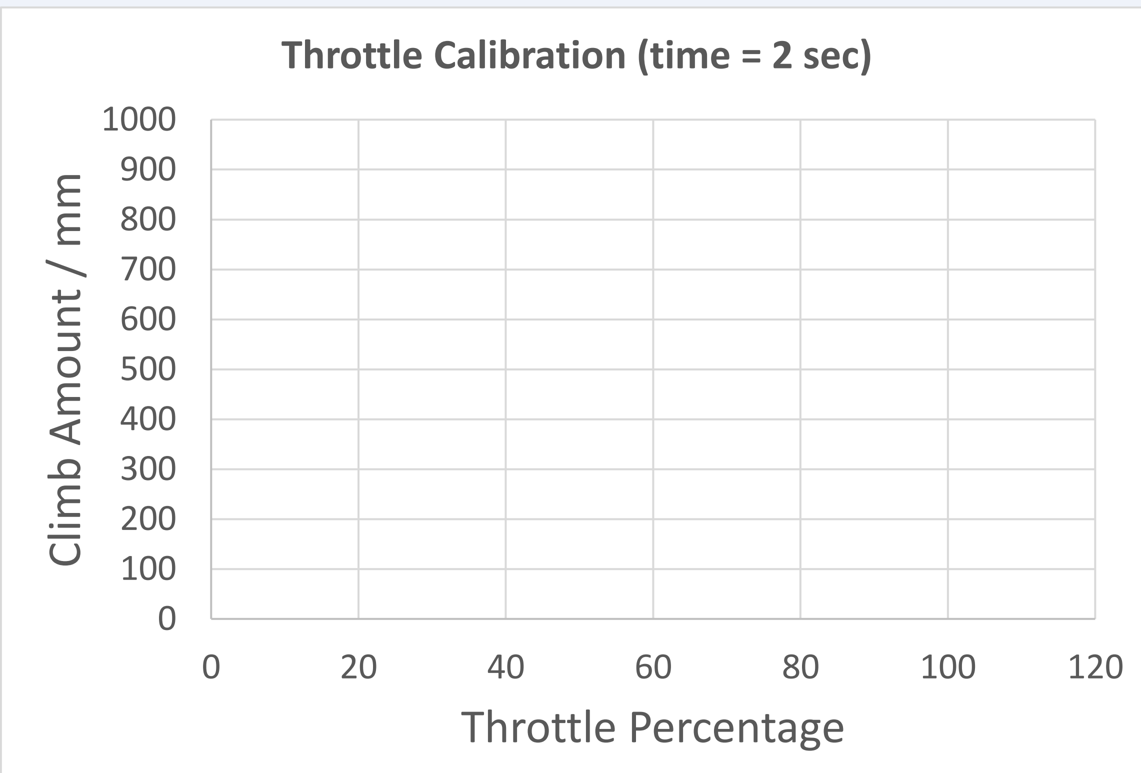
Thrust Time / sec	Throttle / %	Height after take-off / mm	Height after climb / mm	Climb Amount / mm
1	0	500	550	50
1	20	500	600	100
1	40	600	750	150
1	60	550	1000	250
1	80	550	1200	650
1	100	550	1450	900



Try This: Throttle Calibration Test (continued)

Table 5: Throttle Calibration (time = 2 seconds)

Thrust Time / sec	Throttle / %	Height after take-off / mm	Height after climb / mm
2	0		
2	20		
2	30		
2	40		
2	50		
2	60		

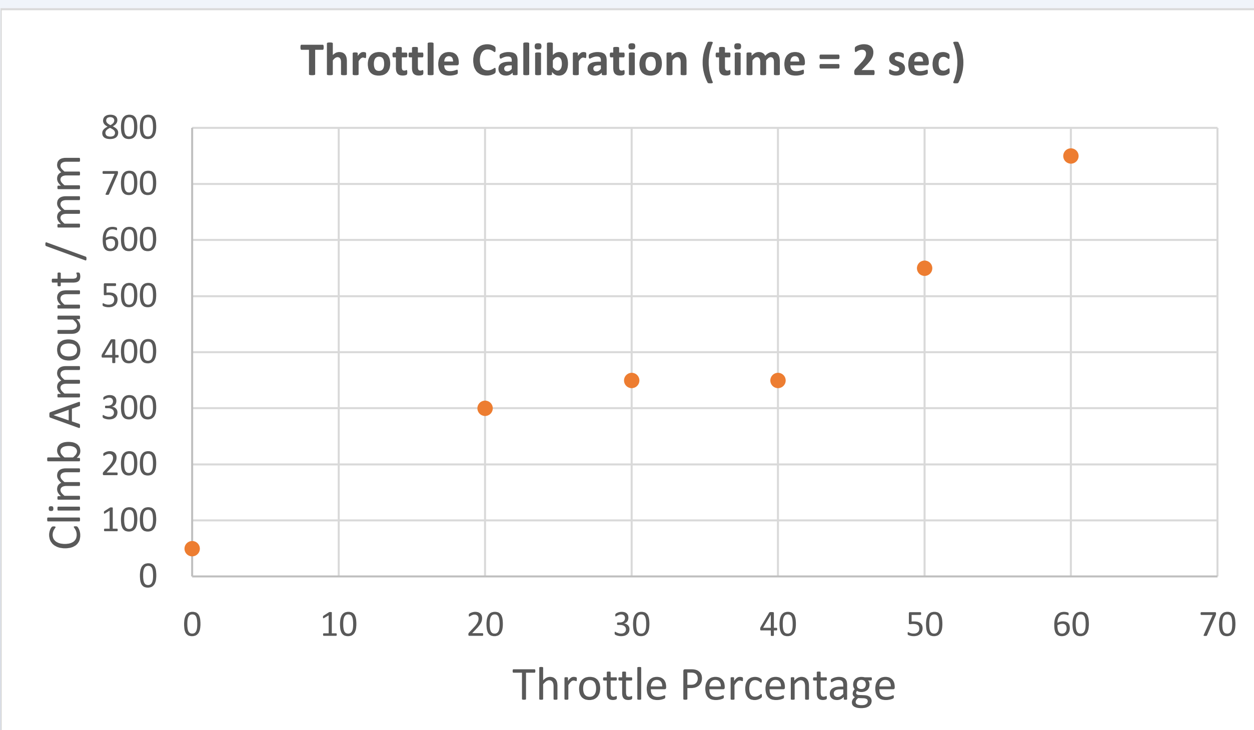


Try This: Throttle Calibration Test (continued)

Here is a sample Throttle Calibration Test (Data and Graph)

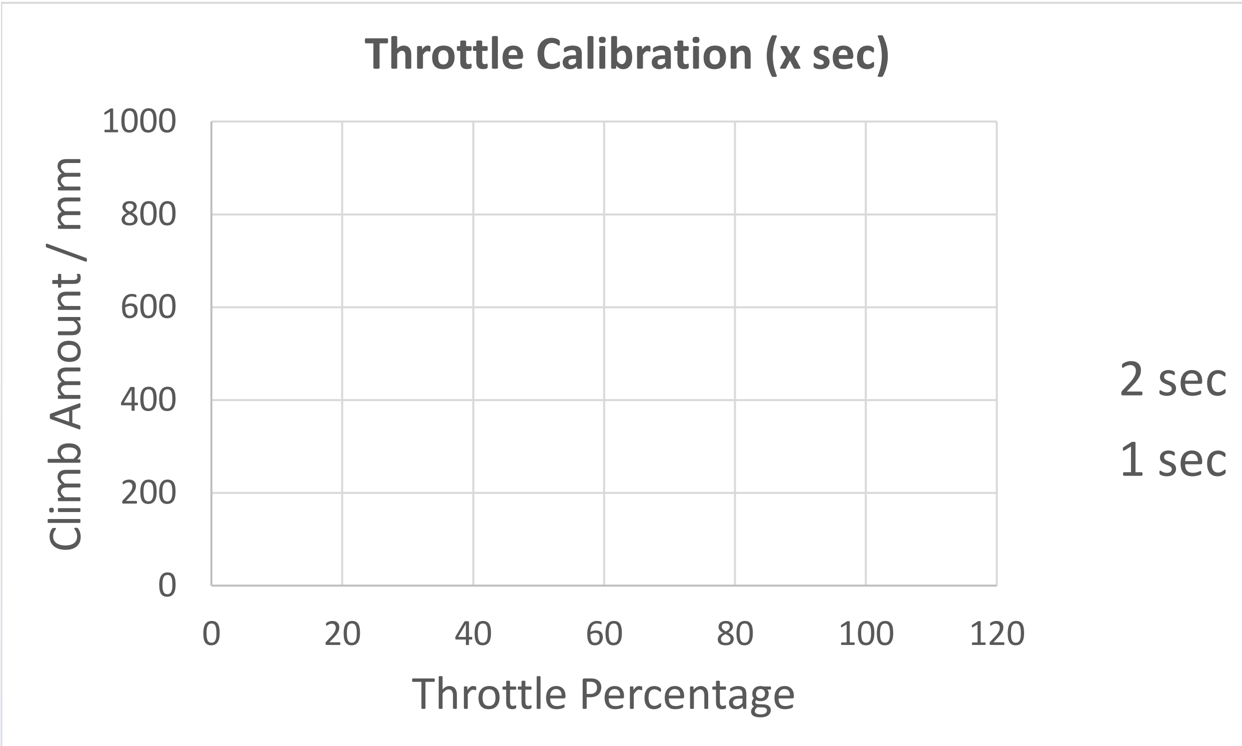
Throttle Calibration (time = 2 sec)

Thrust Time / sec	Throttle / %	Height after take-off / mm	Height after climb / mm	Climb Amount / mm
2	0	500	550	50
2	20	600	900	300
2	30	550	900	350
2	40	550	900	350
2	50	550	1100	550
2	60	550	1300	750



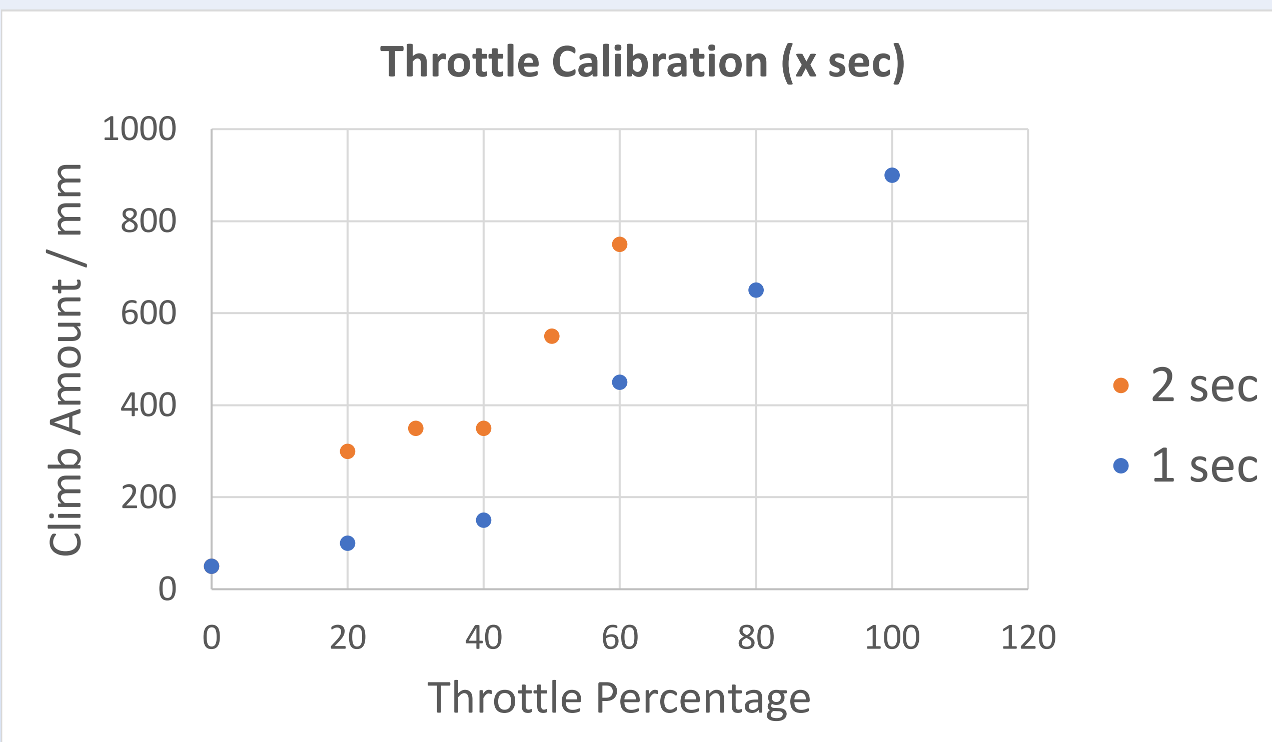
Try This: Throttle Calibration Test (continued)

Throttle Calibration comparison 1 second and 2 second



Sample graph for the comparison of Throttle Calibration

(1 second and 2 second)



Try This: Throttle Calibration Test (continued)

You should be able to clearly observe that the drone climbs higher in 2 seconds. Does the graph also reflect that the height of the drone varies quite a bit? We could make another test series and we would probably find it varies as well.

We can estimate that the drone climbs about:

a) 300 mm at 50 % throttle in 1 sec, and
b) 600 mm at 50 % throttle in 2 sec, or
c) In algebraic form:
   H = 6 mm · 1 · t / (% sec)
   "l" is the throttle percentage and "t" is time.

Can you test the above formula with some of your measured data points and compare?

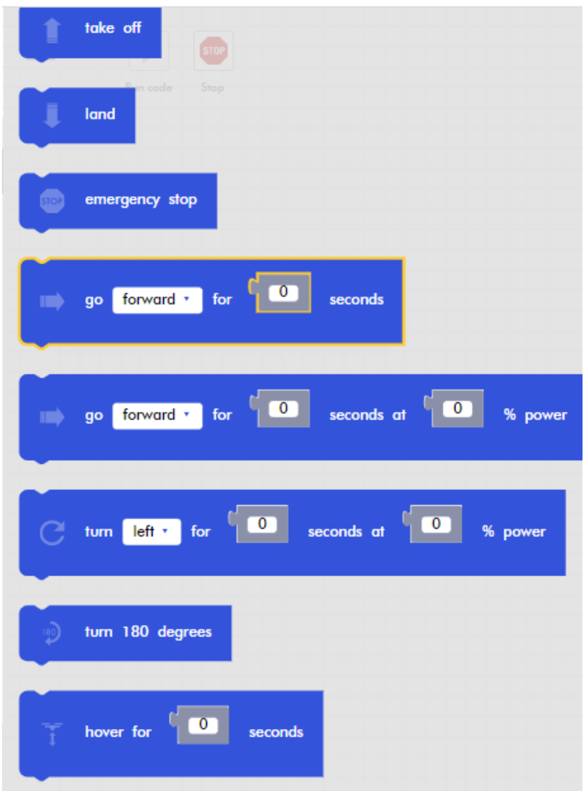
At this time, you are ready to fly the drone in a larger space. Drone flight practice will take part in the Basketball gymnasium or a larger space. Be sure to take your drone, batteries, charger, USB cable, and your Student Lectionary. It is recommended to place all items in the box provided for easy transport. You will be assigned a designated Flight Practice Station by your instructor.

4.6 Flight 7: Pitch Calibration

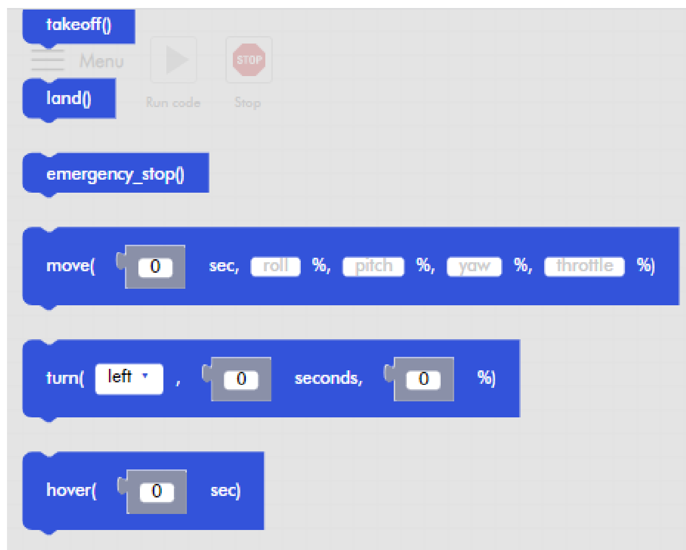
Now that you have accomplished and thoroughly tested the yaw and throttle capabilities of the CoDrone, it is time to practice flying the drone forward. This is accomplished by pitching the drone forward. For this action the propellers need to work a bit harder to still provide the adequate lift to keep the drone in the air while also moving the drone forward.

In Blockly, look at the "Flight Commands" toolbox in both the Junior and Senior blocks. They both have commands for moving the CoDrone forward. The Junior Blocks allow you to go forward for () seconds with or without a selected % power. There is also a block to turn the drone exactly 180 degrees. The Senior blocks allow for more control of all four of the drone movements: roll, pitch, yaw, and throttle.

Junior Flight Command Blocks:



Senior Flight Command Blocks:



We are going to use the Senior blocks for our Pitch Calibration but keep in mind the other blocks are available for use when you begin to try to create more complex flight pattern combinations.

Let's get started learning more about pitch!!

Time to Code: Pitch Calibration

1. Open the Blockly Editor.

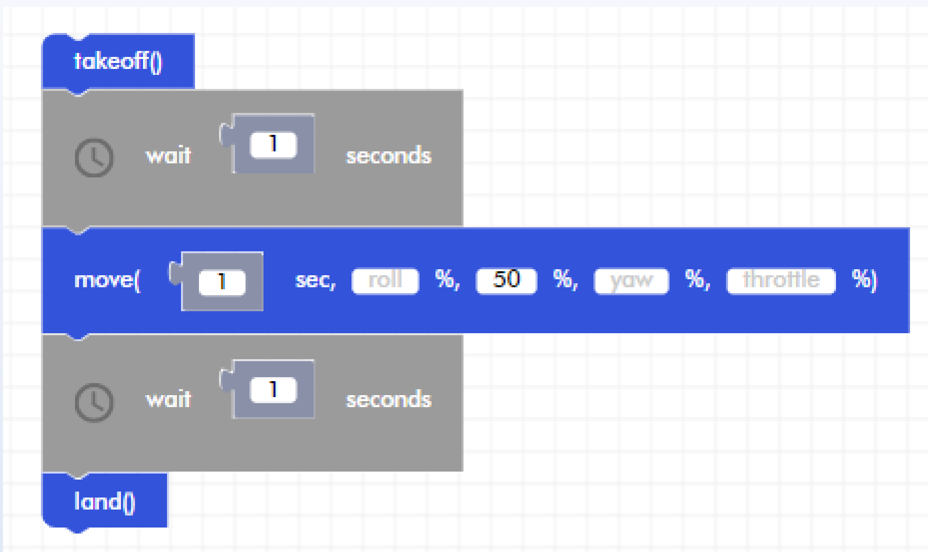
2. If you have not already saved your code to your USB flash drive for the Throttle Calibration you may wish to.

3. Delete the blocks on the workspace except for the "take-off" and "land" blocks by dragging them to the trash can.

4. Add a "move (1sec, 50 % pitch)" block between the "take-off" and "land" blocks.

5. Add a "wait (1 sec)' block before and after this block.

   Your code should look like this:



Let's test out this code.

- Make sure your flight path or area is clear.
- Place the drone on the floor on your landing pad and press play.
- The drone should take-off and move forward and land.
- Use a meter stick to measure the distance that the drone moved.
- You will do an activity to calibrate pitch.

Before we begin the Pitch Calibration test, let's add a functional improvement to the coding to terminate the flight when something goes wrong. Earlier you were instructed to include an emergency landing on your workspace when you create code. Let's modify that to make it more functional.

Time to Code: Soft Emergency Landing

1. In the Pitch Calibration code from the previous activity, add the following new blocks.

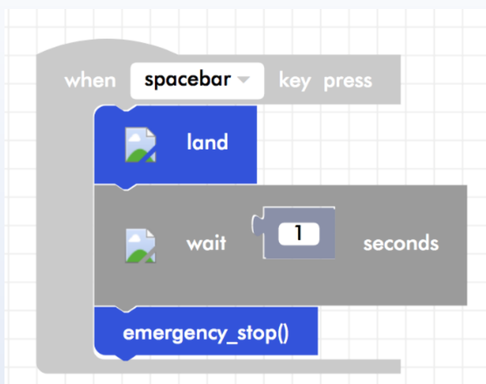
2. Add from the "Keyboard Input" toolbox a "when backspace key press" block somewhere on the work bench. Change the "backspace" variable to "spacebar".

3. Add a "land" block inside the "when spacebar key press" block.

4. Add a "wait (1 sec)" block after the above.

5. Add an "emergency stop ()" block after the above.

It should look like this:



Test within any flight. Pressing the spacebar interrupts the running code.

Depending on when you press the spacebar, the other code may continue to run after, which may make it appear it did not work. Therefore, it is recommended to repeatedly press the spacebar until the drone remains on the ground.

Be sure to include these blocks in any new code on the workspace to allow for stopping the drone in an emergency and to allow for a softer landing to minimize damage to the drone.

Try This: Pitch Calibration

In the Blockly Editor, you should still have the Pitch Calibration code with the now added Soft Emergency Landing. You will need to make sure you have adequate space to fly the drone.

1. Run the Pitch Calibration code for "move (1 sec, 50 % pitch)".
2. Measure how far the drone travels in 1 second using a meter stick.
3. Record this information in **Table 6** on page 82 in both centimeters (cm) and meters (m).
4. Change the seconds by the increments shown on **Table 6** and test for each increment.
5. Record each distance that the drone travels.
6. On the last test, note that both the seconds and the % pitch change. (100 % pitch)
7. After collecting the data on **Table 6**, complete the graph below it to represent the data points.
    - See the sample table and graph on page 83.

Look at your graph!
It makes sense the longer the drone pitches the further it goes.
This should give an approximate straight line.
One point is special. Does that make sense to you?

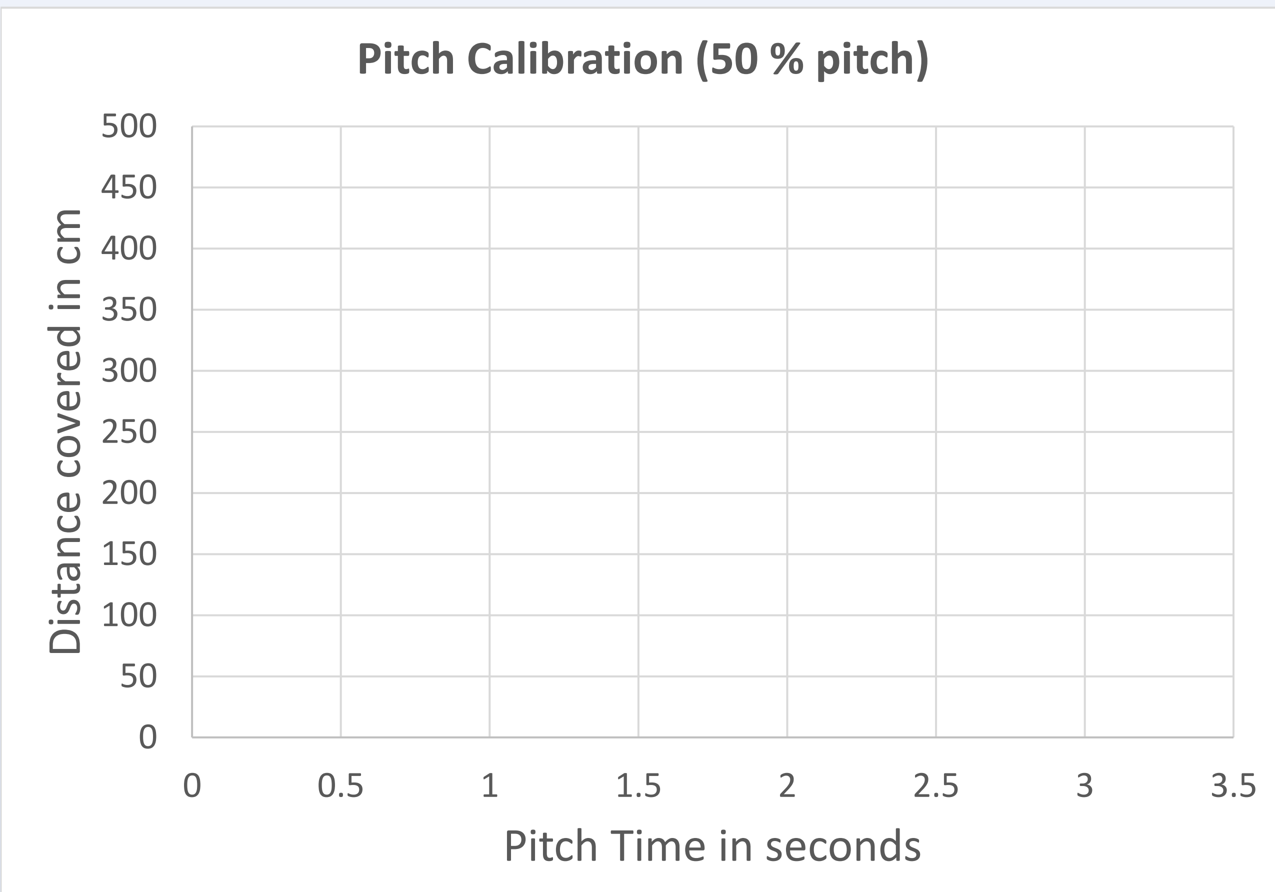
Try This: Pitch Calibration

Record and graph your Pitch Calibration information here:

Table 6: Pitch Calibration

Pitch Calibration (50 % Pitch)

Pitch Time / sec	Pitch / %	Distance covered / cm	Distance covered / m
1	50		
1.5	50		
2	50		
2.5	50		
3	50		
1.5	100		

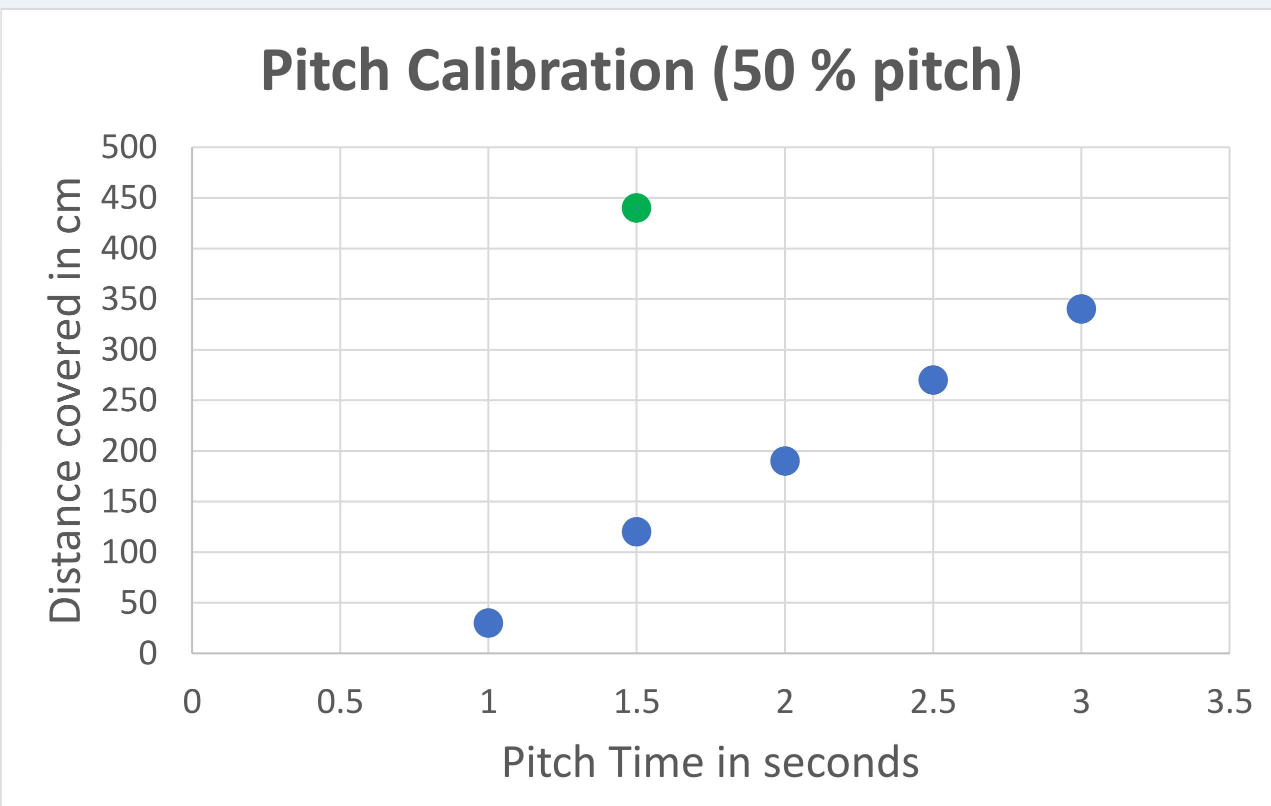


Try This: Pitch Calibration

Here is a sample of the Pitch Calibration. Does yours look similar?

Pitch Calibration (50 % Pitch)

Pitch Time / sec	Pitch / %	Distance covered / cm	Distance covered / m
1	50	30	0.3
1.5	50	120	1.2
2	50	190	1.9
2.5	50	270	2.7
3	50	340	3.4
1.5	100	440	4.4



Flight Practice 1: Earth to Mars (and Back)

Flight Practice is your time to practice the skills you have learned with the drones and to prepare for the competition event at the end of the camp. The first practice will involve traveling from Earth to Mars. You will use flat representations of Earth and Mars to make landing on them easier.

Earth to Mars: (Practice for Competition Event #1)

1. Measure the distance between center of Earth and Mars.
- Record the measurement here: ______ cm

2. Look back at your Pitch Calibration graph and find the distance you just measured on the vertical axis. Move horizontally to a straight line connecting your data points (with 50 % throttle).
- What time is needed in your code to get you to Mars? _______ time

3. Change the Pitch Calibration code to this time.

4. Test your drone to see if it reaches the target. (Mars)
- Place your drone on the center of Earth and run your code.

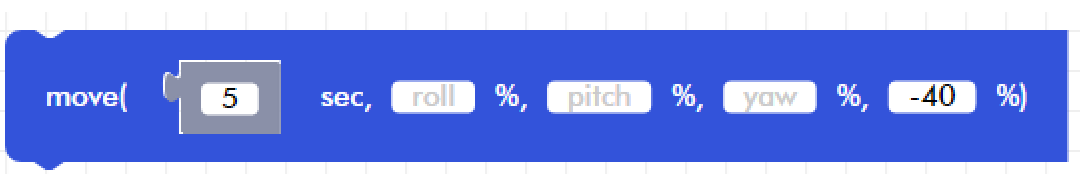
5. If needed, change the code to try and land on Mars.

The goal is to land as close to the center of Mars as possible with the fastest time and best accuracy.

Optional Improvements:

Soft Landing:

Whenever you want your drone to land, it is always a good idea to do so softly to minimize damage to the drone. Add this block to your code right before the land block:



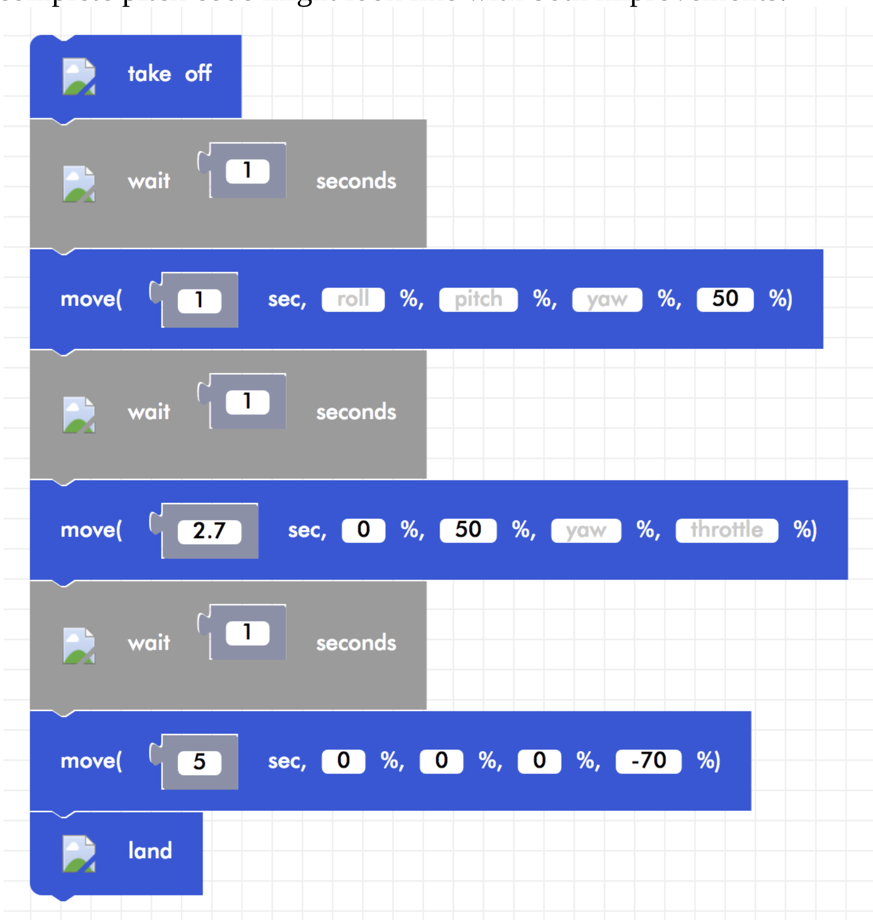
The time and (negative) yaw can be adjusted for best results.

Fly Higher:

You may want your drone to fly at a higher altitude. To accomplish this, you can add an additional block.

- Add an additional "move (1 sec, 0 roll, 0 pitch, 0 yaw, +50 % throttle)" after the first "wait block."
- Add a second "wait (1 sec)" block before the next move command.
- Test if your drone flies to Mars at a higher altitude.
- Adjust time and throttle value in the last "move ()" block until your soft-landing works better again.

Here is what the complete pitch code might look like with both improvements:



Once you have perfected the code and practiced for the Earth to Mars task – it is advisable to save this code to your USB flash drive and title it "From Earth to Mars".

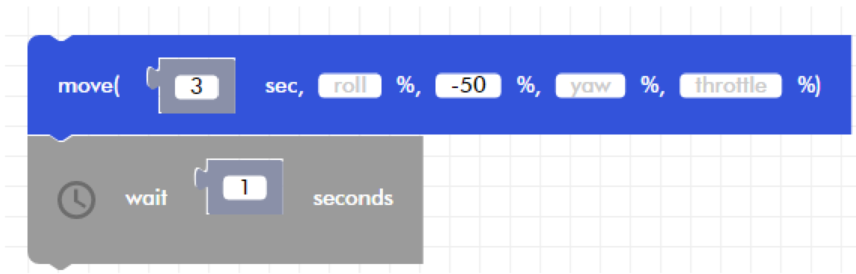
To Mars and Back: (Practice for Competition Event #2)

For this practice you will travel to Mars from Earth, then return to Earth and land. You will not have to land on Mars but try to turn as close as you can over Mars.

1. Use of the code from Earth to Mars.

2. Add an additional "move (3 sec, 0 roll, -50 % pitch, 0 yaw, 0 throttle)" block before the last (soft landing) move () block.
   - Notice the negative pitch value! A negative value here moves the drone backwards instead of forward.

3. Add a "wait (1 second)" block after it.



4. Test this code.
   - The drone should travel to Mars, then come back to Earth to land.
   - Adjust the times and pitch values to in the last block to get the drone to land as close to the center of Earth as possible.
   - Practice this.

The goal of this task is to travel to Mars, turn around and come back to Earth and land as close to the center of Earth as possible with the fastest time and best accuracy.

Optional Improvements:

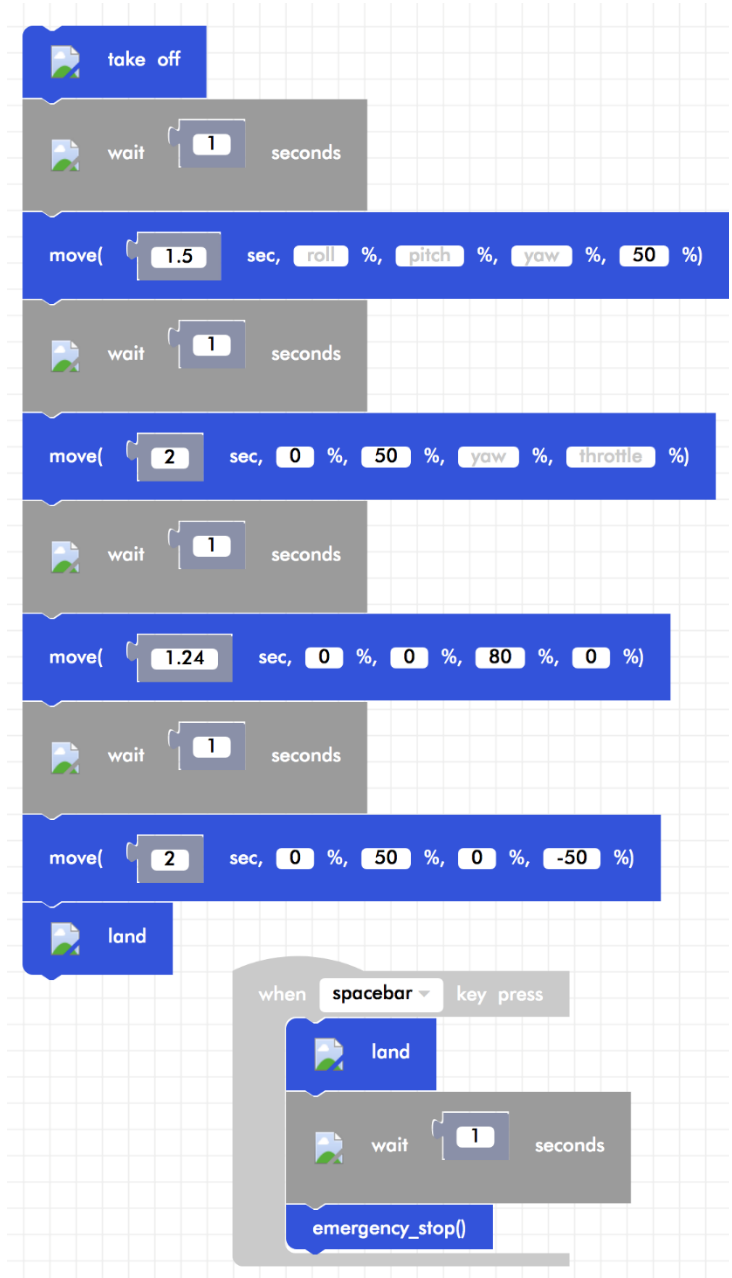
Save Time:

You can save time with waiting to land the drone softly in the last block by combining the return to Earth block (negative pitch) and the descending blocks (negative throttle).

- Delete the soft-landing block and modify the last "move ()" block with your parameters to combine the flight back (pitching with a negative value) and the descend (with a negative throttle value) at the same time.

You may also wish to rotate the drone over Mars and fly home to Earth in a forward motion. You can look back to your Yaw Calibration data to determine what Yaw % and time will turn your drone 180% degrees so that it is facing home (Earth). Remember to change the pitch value for the return!

Here is what both of those additions might look like in your code:



Here is what each move block is controlling:

1st move block- Adjusting the height of the drone

2nd move block- Moving the drone to Mars

3rd move block- Turning the drone around to face Earth

4th move block- Returning to Earth and descending at the same time.

Once you have perfected the code and practiced To Mars and Back – it is advisable to save this code to your USB flash drive and title it "To Mars and Back".

4.7 Flight 8: Change the LED Colors and Turn in a Circle

For the next test flight, you will be learning how to change the LED colors of the drone and how to code the drone to make a circle. The LEDs are an important feature of the drone and can allow you to use them as status indicators during flight practice. Read below more about LEDS.

What is an LED?

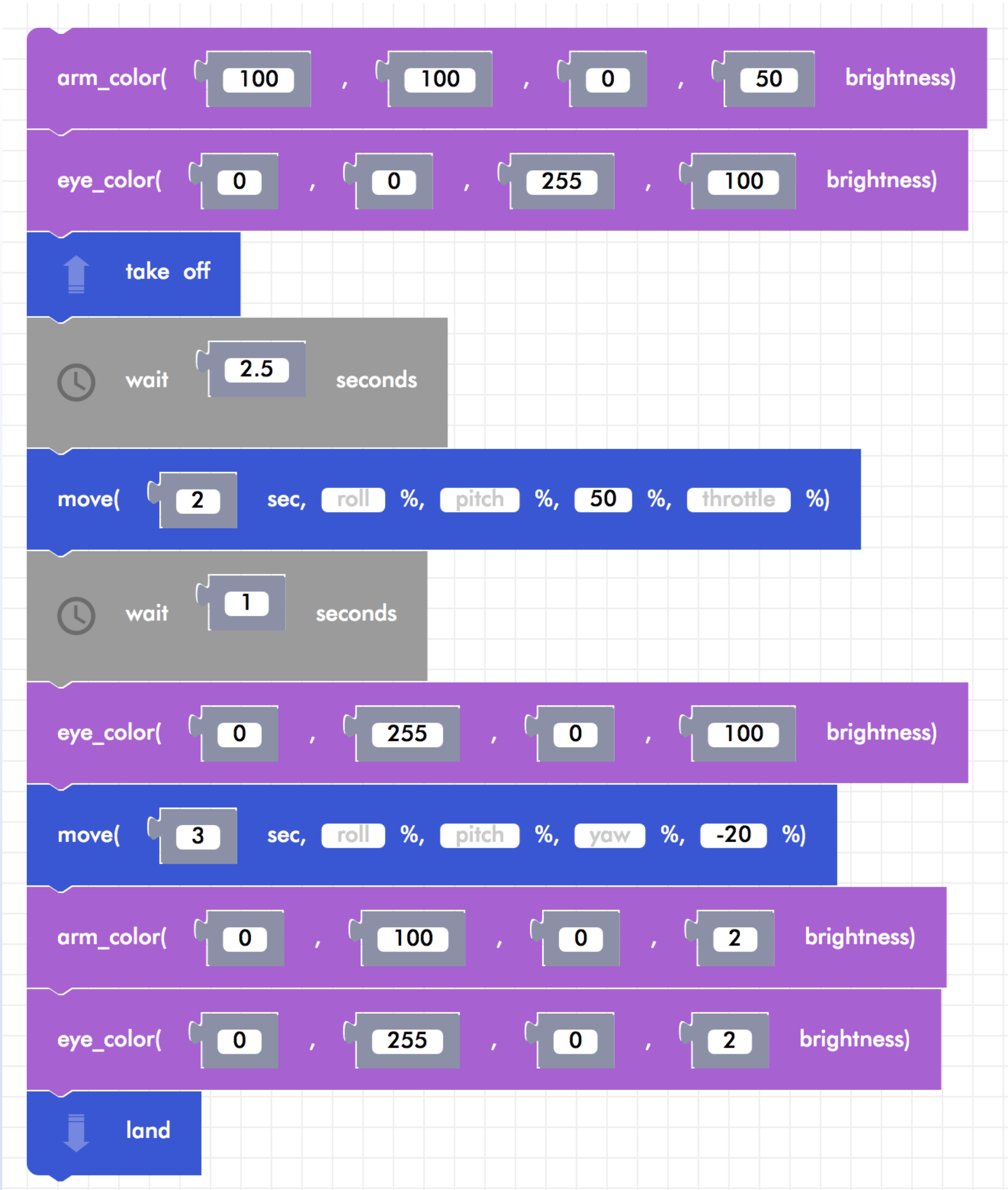
LED stands for "light-emitting diode". A diode is an electronic device that allows an electric current to flow in only one direction. LEDs are beginning to replace traditional light bulbs because they are more efficient.

The CoDrone has 6 RGB LEDs: one for each of the four arms, one for the eyes, and one for the tail. RGB stands for red, green, and blue. Using these three colors, you can create almost every color imaginable! Aside from making your drone look cool, LEDs provide another form of feedback. For example, the taillight can tell you the connection status of drone. A solid green taillight means you drone is connected, while the flashing green light indicates the drone is in pairing mode. You cannot change the color of the taillights on the CoDrone but all the other 5 LEDs can be programmed to produce any color and pattern that you want!

There are blocks in both the Junior and Senior Blockly editors for programming the LEDs. Look at the differences in the blocks. We will be using the Senior blocks in the next coding activity. You can read more about them here: https://learn.robolink.com/lesson/codrone-blockly-sr-leds/

Time to Code: Changing the LEDs and Turn a Circle

1. Open the Blockly editor.
2. Be sure to save your Throttle Calibration program to your USB flash drive.
3. Clear the workspace by dragging the blocks into the trash can.
4. Using the Junior and Senior blocks, recreate the following code in the workspace:



Test this code in your drone. Note the changes in arm and eye color as well as the movement that the drone performs.

Time to Code: Changing the LEDs and Turn a Circle (continued)

Let's break down the code to make sure you understand it.

Look at the purple blocks that control the color:

- What color were the arm LEDs before take-off? right before landing?
- What color was the eye LEDs before take-off? right before landing?
- Can you figure out what the first three fields are for in the color blocks?

The first three fields in the color blocks represent RGB or red, green, and blue respectively. Does the arm color before take-off and the way you filled in the fields make sense?

Did you notice that the 2 in the brightness field caused the lights to dim? This is done to save battery power.

After take-off there is a long "wait 2.5 seconds" block. This wait times allow you to check to see if the colors are correctly coded.

Note: wait times of 3 seconds or more cause the drone to immediately land and not execute any later commands.

The first "move (2 sec, yaw 50 %)" causes the drone to turn in a clockwise direction with 50 % power for a duration of 2 seconds. A negative number for the yaw percentage would cause the drone to turn in the opposite direction or counterclockwise. The second "move (3 seconds, throttle -20 %)" causes the drone to descend. This is intended to allow the drone to land more softly. You can also change the duration of 3 seconds to a larger or smaller value to improve landing performance.

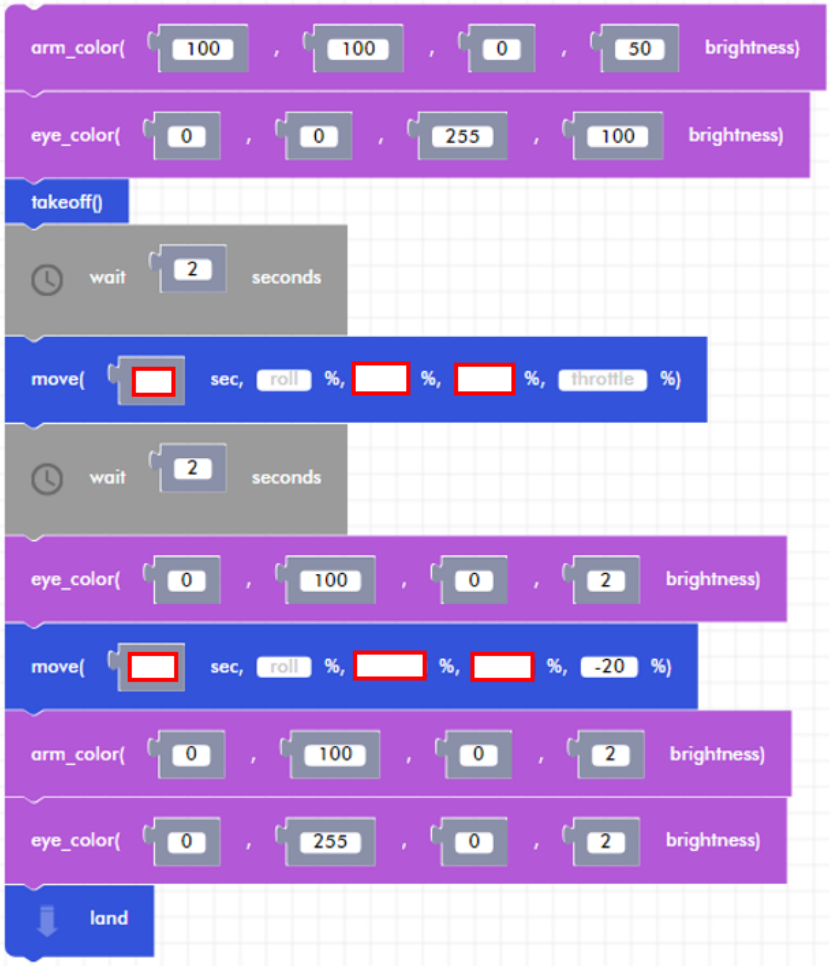
Now that you know what affect theses coding parameters have on the drone, see if you can vary some of them to make your drone maneuver better.

For now, only change the seconds, yaw %, and throttle % in the move blocks. We will work with the other move parameters later.

- Can you make your drone rotate so that it lands facing the same direction it was facing at take-off?
- Does your drone land softly?

When you are satisfied with the performance of your drone enter the new parameters in the blank table on the next page so you will remember them.

Enter the parameters in the blank table:



4.8 Flight 9: Flying a Circle

Now that you have a better understanding of how the yaw and throttle control the movement of the drone, let's look at how adding pitch can change the drone's course.

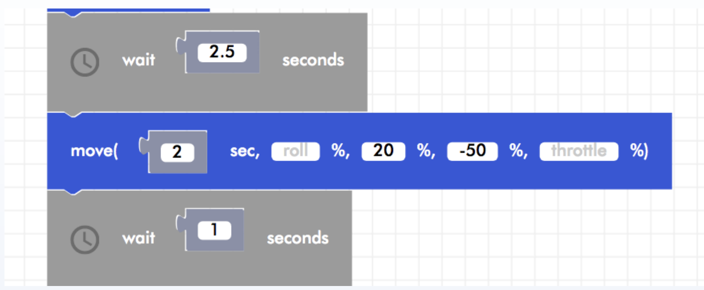
Remember:

Pitch is the CoDrone's forward and backward tilt. Positive pitch will make the CoDrone tilt and move forward, and negative pitch will make the CoDrone tilt and move backwards.

Time to Code: Flying a Circle

1. Open the Blockly editor.
2. Save the previous code "LEDs and Turn a Circle" to your USB flash drive.
3. Do not delete the code in the workspace, you will just modify it for Flying a Circle.
4. Change the first "move" block:
   - Change the yaw % to "-50 %".
   - Add a pitch % of "20 %".

Your code should now look like this between the two "wait blocks".
(You may have adjusted the times in your move and wait block. If so, just add pitch to your block.)



5. Test the new code with the drone.

   Start the drone on the compass rose take-off/landing target. You should be able to do this seated at your desk. **Note:** The drone will probably land somewhere behind you! Try to catch it before it crashes if possible.

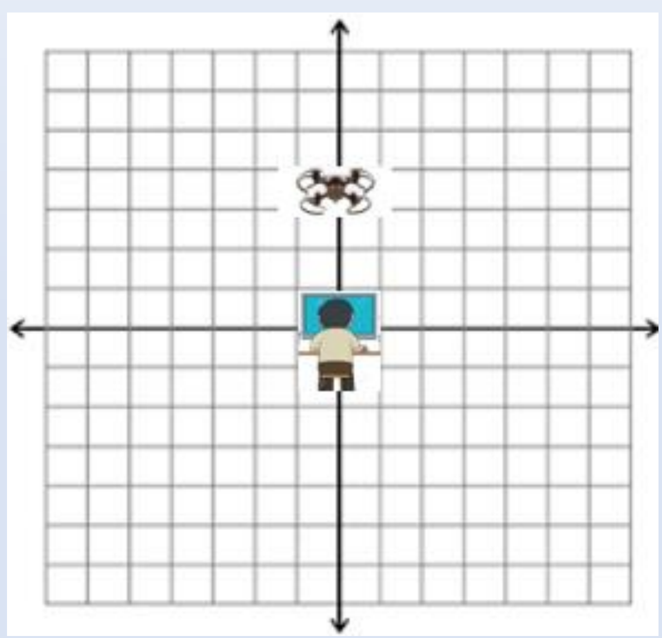
   Does the drone move forward?

   Does the drone turn? Which direction?

   How close to completing a full turn or full circular flight (orbit) was the drone?

6. Make a map of the course your drone flies. Draw the essential locations and orientations of your drone into the following diagram.

   Did your drone fly in an arc?



4.9 Flight 10: Completing the Circle

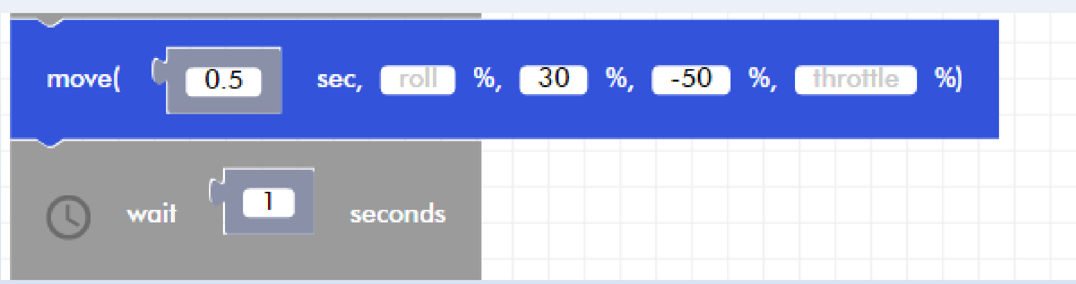
You may have noticed that your drone does not fly a complete circle around you. It probably lands somewhere between 240° and 280°. Can you extend the course to complete the circle? So far, we have worked with pitch, yaw, and time. Which of these controls do you think would complete the circle? Logically, it might seem that the easiest way to close the circle would be to increase the time so that the drone will fly the course longer. Try this out and see!

What did you discover? You may have noticed that adding more time does not work well because the drone gains speed as it turns and thus the course does not make a full circle but an outward spiral. So, what might be a better way to get back to the take-off point? What about adding another new arc? This should allow you to adjust the parameters to efficiently close the turn in a more circular pattern.

Let's try this!

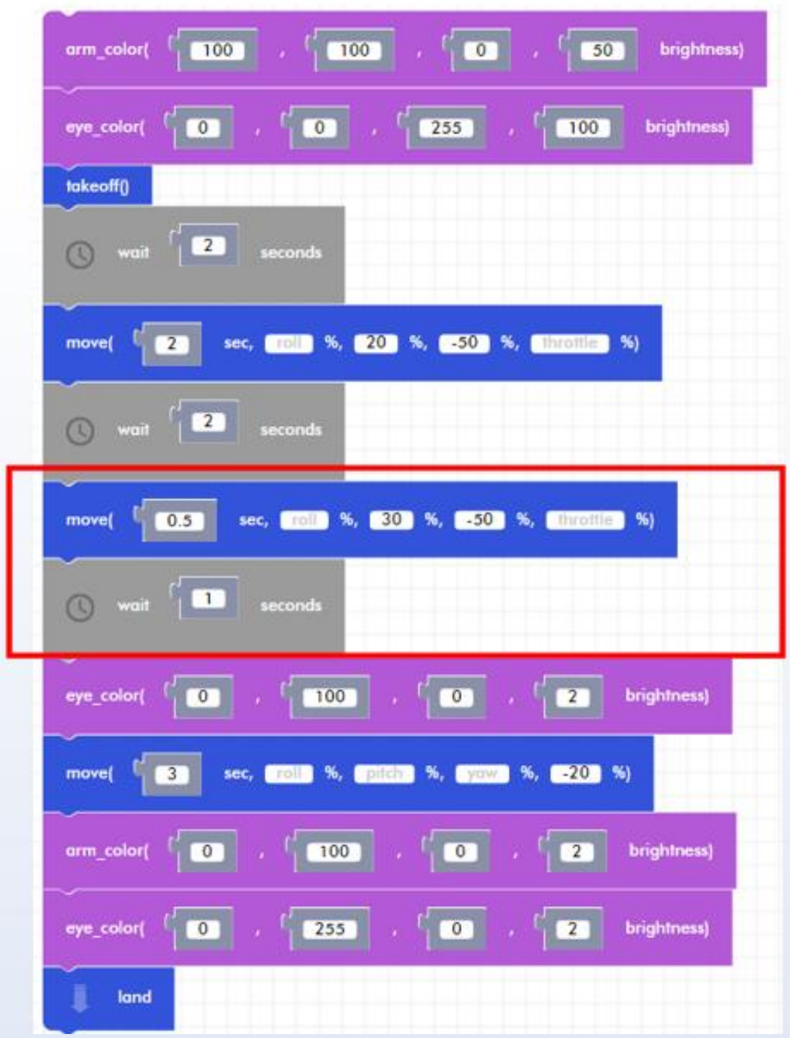
Time to Code: Completing the Circle

1. **Open** the Blockly editor.
2. **Save** the previous code "Flying a Circle" to your USB flash drive.
3. Do not delete the code in the workspace, you will just modify it for Completing the Circle.
4. Add an additional move block to your previous code for the second arc:
    - Place this block after the second wait block and before the LED eye color signal for the "soft landing".
    - Set the move (0.5 sec…) to complete the loop.
    - Set the pitch to 30%.
5. Add a wait block after the new move block. Set the wait for 1 sec.



Time to Code: Completing the Circle (continued)

Your code should look something like this but remember you may have different values for your time and pitch based on your previous testing. (New blocks outlined in red.)



6. **Test this new code.**
   - Did your drone complete the circle?

     If not, you may need to adjust the time in the new move block.
   - Did your drone fly in a circular pattern?

     If not, you may need to adjust the pitch.
7. Continue adjusting the parameters until your drone can complete a circular pattern or orbit consistently.
8. Go back to the diagram in "Flying the Circle" and draw this second arc on the diagram.
   - Can you estimate the angles of both arcs? The radius of the circle?
9. Be sure to save this code on your USB as "Completing the Circle"

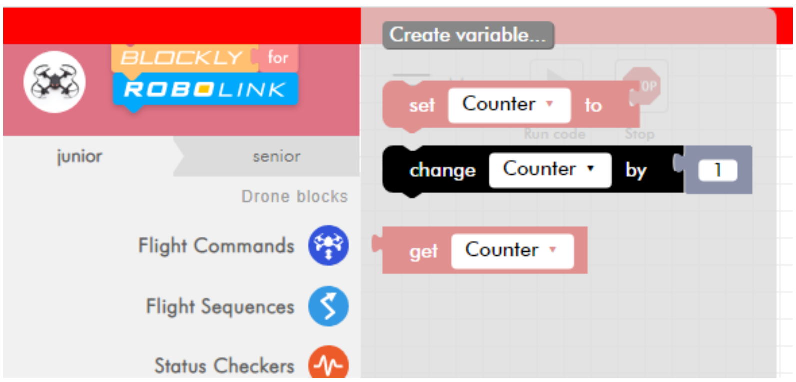
4.10 Flight 11: Using a Coding Loop to Complete a Circle

Now that you have a better understanding of how pitch and yaw can control the movement of the drone and create a circular pattern or orbit, let's see if we can make the movements in the orbit a bit more precise. For this we will use some new coding blocks: variables and loops.

What are Variables?

"Variables are a way to store information in a program, and they're used most often to store values. If you're playing a game on your phone or elsewhere, there's a variable to store your score, another one for your lives, and maybe another one for your time. With your CoDrone Mini, you might use them to store the value of a sensor, the speed of your drone, or how long it will travel for." Robolink, Basecamp

To learn more about variables, visit the Robolink Basecamp lesson 2A: Variables.
https://learn.robolink.com/lesson/variables/

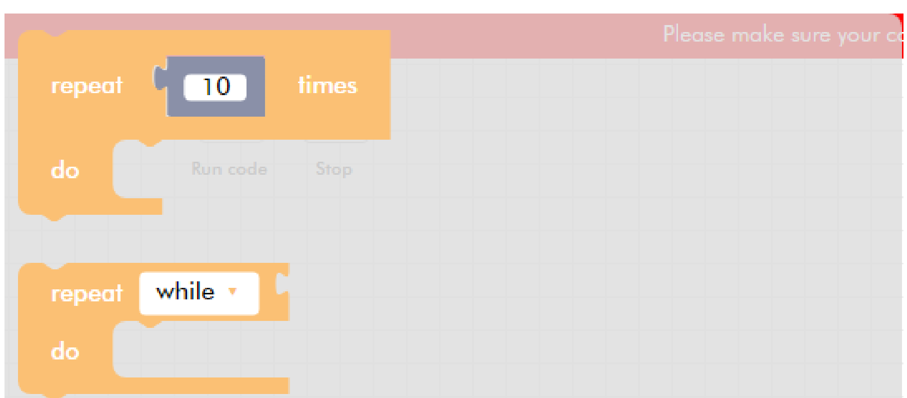


Here the variable created was named "Counter". We will use this to count the number of times we run a certain execution of code.

What are Loops?

"Loops let you repeat sections of your code. They're a great shortcut — without them, you might have to connect the same blocks over and over again!" Robolink Basecamp

To learn more about using loops in code, visit the Robolink Basecamp lesson Loops.
https://learn.robolink.com/lesson/codrone-blockly-sr-loops/

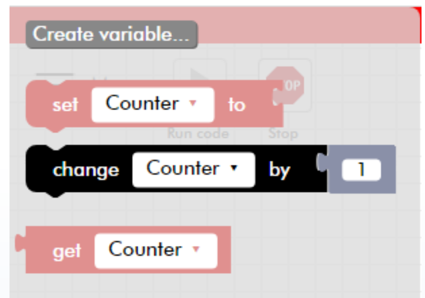


With these new blocks, you can imagine that we should be able to repeat a movement multiple times to complete an orbit or circular path.

Let's try out a new way to write our code!

Time to Code: Complete a Circle (or Orbit) with a Loop

1. Open the Blockly editor.
2. Make sure you have saved your code for "Completing the Circle" to your USB.
3. You will modify this code with loops and variables.
4. Create a variable named "Counter".
    - Select variables from the Blockly toolbox.
    - Type in the name of the variable "Counter" in the box that says create variable.
    - New blocks will appear with the variable inside of them under the variables in the toolbox.



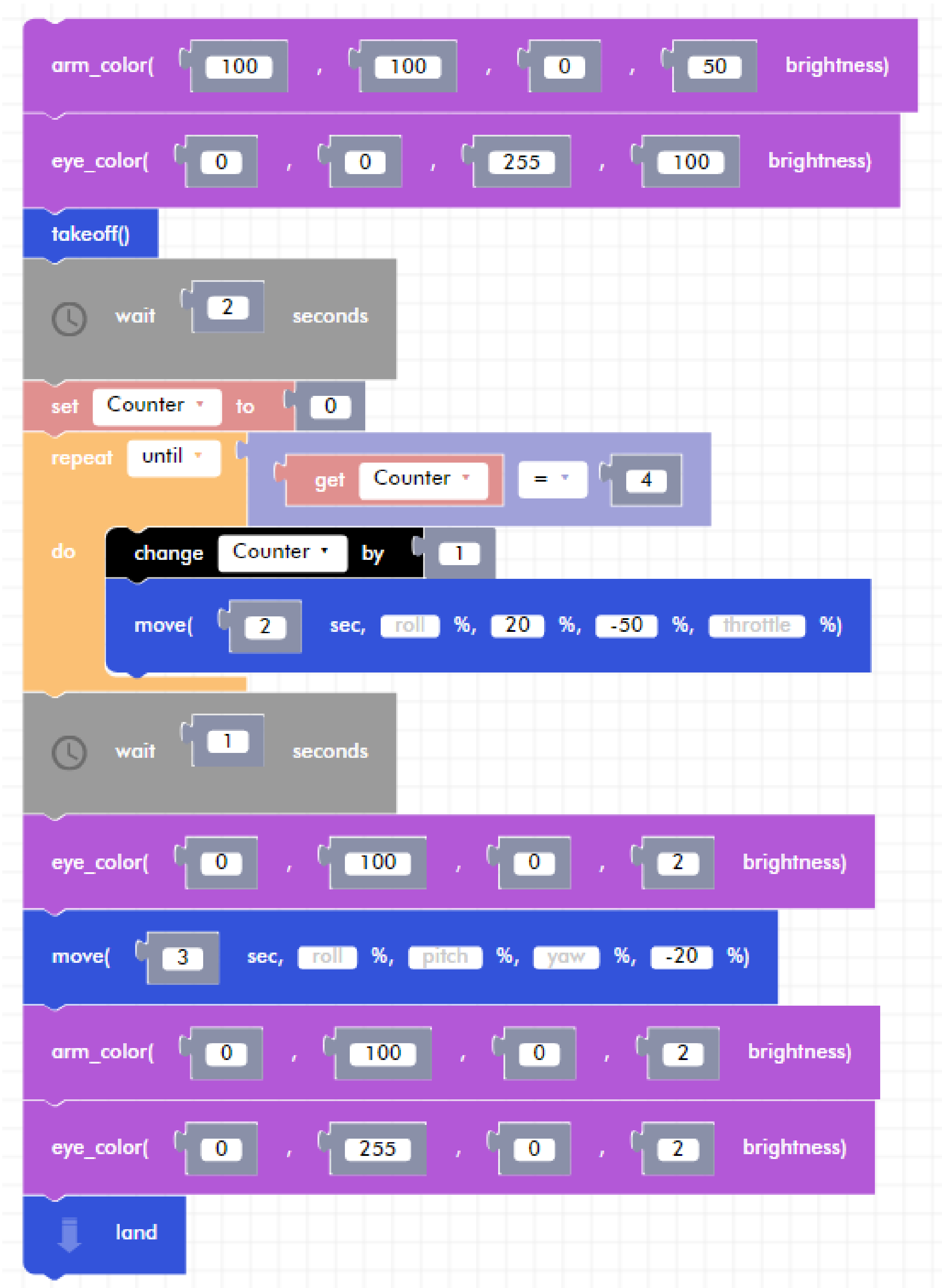
5. Drag the "set Counter to" block and place it right after the first "wait (2 secs)" block.
6. Drag an empty number block from Math and place at the end of the "set Counter to" block.
    - We will leave this as a "zero".
    - Wonder why "0"? Computers start counting at "0" not "1".

Now let's make the loop!

7. Drag a "repeat while/until" block from Loops in the toolbox.
    - Place it after the "set Counter to" block.
    - Click the drop-down menu in this block and change the "while" to "until".
    - Drag a "_ = _" block from Logic and place into the first notch of the "repeat until" block.
    - Drag a "get Counter" block from the Variables toolbox and place it in the first space of the "_ = _" block.
    - Drag an empty number block form the Math toolbox and place it in the second space of the "_ = _" block.
    - Click on the number block and change the zero "0" to "4".
    - Drag a "change by Counter" block into the "repeat until" block in the slot after the "do".
    - Add an "number block" to the end of the "change Counter by" block and set this number to "1".
    - Now move the first "move (2 sec….)" block from your previous code into the "repeat block" and place it under the "change Counter by" block.
8. Delete the second "wait" block and "move" block from the previous code.

(See the next page for how this new code should look.)

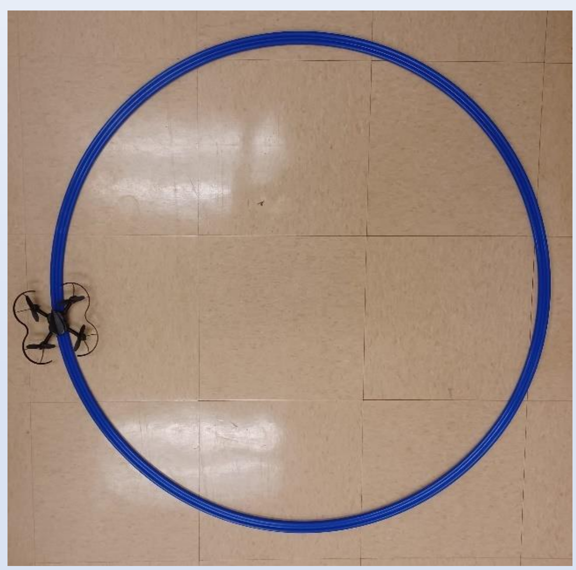
Using Loops to Complete a Circle code:



Try This: Flying a Precise Orbit (Complete Circle)

Now that you have learned to complete a turn, it is time to perfect it more by completing the turn or orbit with a specific circumference. Part of the competition will involve flying around obstacles and you will want to be able to fly close to them to minimize your time factor but also not too close to hit them. You can practice this by using a large hoop. The goal is to try to follow the path of the hoop which is 36".

1. Place the hoop on the floor in a safe flying zone. You will use the hoop as a guide to perfect your circle or orbit.
2. Use the previous code "Complete a Circle with a Loop".
3. Place your drone on the landing pad facing right.
4. Run your program.
   - How close did your drone come to following the path of the hoop?
5. Adjust your parameters to make the drone follow the path of the hoop.
6. Continue practicing flying a consistent orbit along the path of the hoop.
7. Make sure you understand how to adjust your drone to accommodate a larger or smaller orbital circumference. Be sure to write down notes.



4.11 Optional Flight 12: Fly with Sensors (Adjusting Your Course)

At this point you should have some understanding of how to control the drone by adjusting the yaw, and pitch to control the drone. Remember, the goal will be to fly a course autonomously with the drone. Part of this course will involve orbiting (asteroid) an object in “space”. In the last test you practiced how to control the drone to reach a specific orbital circumference. You also have calibrated the throttle and tested it under specific conditions. Now you will learn about some of the CoDrone’s built in sensors to help you fly at a specific height.

You may have noticed that the drone does not do exactly as told by the code. This is because the drone has built in sensors that makes the drone do many things on its own which can help the drone fly better. However, this comes with a drawback in that the automatic functions may interfere with our intentions.

To understand this better, let’s look at some of the CoDrone’s built in sensors and learn how we can use this to our advantage.

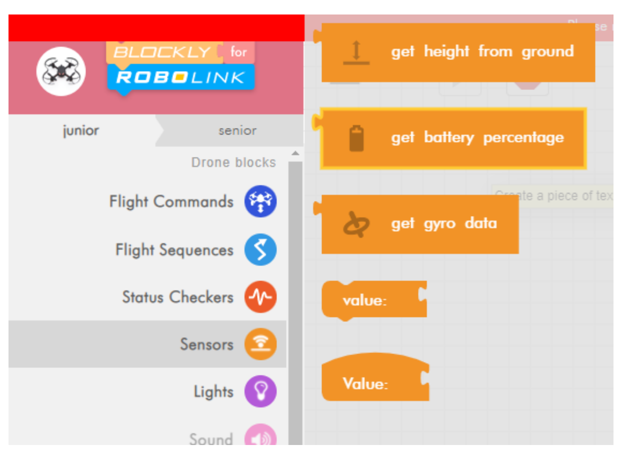
In the Blockly editor look under the Junior blocks for a toolbox named Sensors.

The CoDrone has three sensors:

1. Battery Level Sensor
   - The battery level sensor gives a reading between 0% and 100% - with the latter being a fully charged battery.
2. Height Sensor
   - The height sensor gives you the height of the drone in millimeters.
3. Gyroscope Data Sensor
   - The gyroscope is a bit more complicated as gyros are similar to compasses – they sense rotational motion and detect orientation to figure out which way something is facing.
   - They are used in many things from space shuttles to cell phones and even your CoDrone!
   - The gyroscope is used to stabilize the drone and uses math and some algorithms to correct any unwanted movements to maintain stabilization. The gyroscope uses roll, pitch, and yaw to stabilize the drone. Note throttle is not used because a move up or down is not reflected as angular displacement.

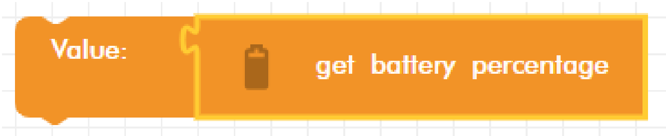
Here are the Blockly blocks for the sensors:
Note there are two blocks labeled “value”. The thinner block can be placed in the code between other blocks to return a value at a specific point during the code. The rounded “value” block is used as a stand-alone sensor measurement and will return one value for the whole execution of the code.

We will use the Battery and height sensors for the next activity.



Try this: Using Sensors (Optional)

1. Use your previous code "Completing a Circle with a Loop"
2. Place a "value" block from the Junior Sensors toolbox at the beginning of your previous code.
3. Drag a "get battery percentage" to the end of this block.



4. Run the code.

After running the code, you should notice a number in the "value" block that gives you the percentage of battery charge at the end of your flight.

- Why might this value be important?
- Do you think the drone performs differently based on battery percentage?

Based on your investigation of the throttle calibration, you know that you can make your drone fly higher or lower using different throttle percentages. Let's insert another sensor block for height and see how the collected sensor data compares to your earlier investigation.

5. Now insert another "value" block after the first "wait" block.
6. Drag a "get height from ground" block to the end of this block.
7. Run the code again.

Notice the height value now listed in the added "value" block.

- Go back and look at the table you completed for your throttle calibration. Is the value close to what you measured for zero throttle?
- If you run the code several times is the value close each time?

Understanding and learning how your individual drone performs will help you set parameters during the competition for the best results.

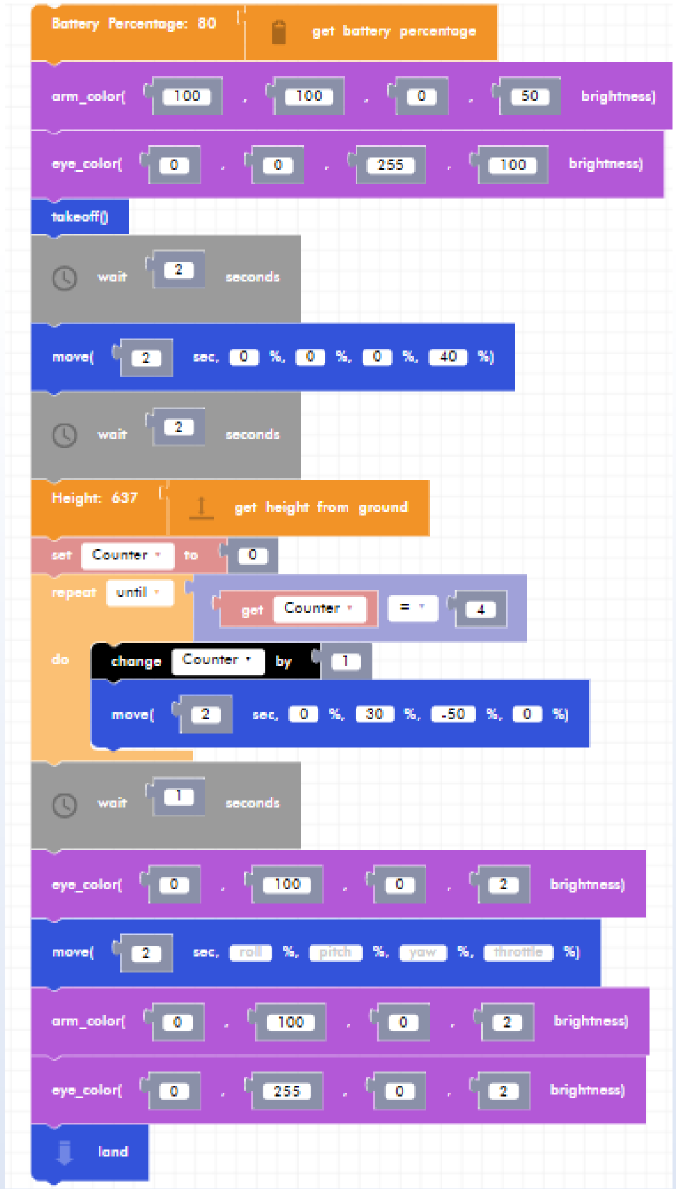
In the "Completing the Circle using a Loop" the goal was to complete an orbit of 36" in circumference. Being able to control the circumference or orbit will help you travel closely around obstacles without hitting them but also to minimize your time. Another factor of the competition will be to travel within a specific orbital plane regarding height. Let's suppose the midpoint of the obstacle is 90 centimeters. At what height in millimeters will your drone need to be to reach this goal? Hint: How many millimeters are in a centimeter?

Let's see if you can code the drone to reach this height.

8. Before the "value" "get height from ground" blocks add a "move…" block.
   - Set the seconds to 2.
   - Look at your throttle calibration table for 2 seconds thrust and choose the percentage throttle that came closest to our target of 90 centimeters or 900 millimeters. Insert this percentage for throttle.
9. Insert a new "wait" 2 seconds block after this.

Try this: Using Sensors (Optional continued)

Your finished code should look something like this: (your values may be a bit different)



10. Run the code.
    - You may want to use the meter stick to sight whether your drone reaches the target orbital plane of 900 millimeters.
    - Adjust the throttle percentage of the first “move” block if your drone did not reach this target.
11. Continue running the code with your drone and adjusting the code until you get a consistent 36” (inch/hoop size) orbit at approximately 900 millimeters in height.

Extension: If time permits, you may wish to insert the “get gyro data” into your code and see if you can make sense of the values this information provides. Do you think this would be helpful information to have as well?

Flight Practice 2: Asteroid Missions

So far you have practice two task (Earth to Mars and To Mars and Back) related to the final competition, now let's use what else you have learned to practice for two more. You should be able to control the CoDrone to takeoff, land, and to fly around an orbit at a specific height. It is now time to take our drone beyond the classroom and to a larger space to practice flying autonomous maneuvers. However, you will need to modify some programs for this before going to the gym. Then you will only need to modify them slightly to get the results you want.

Asteroid Mission: (Practice for Competition Event #3)

Fly to an Asteroid from Earth and maintain a height within the asteroid's orbital plane for an observational mission.

1. Refer to the code on page 85 for this practice. Earth to Mars with Height Improvement.
2. To reach the height indicated on the Mission Boundary polls, you may need to change the throttle in the first "move () block.
3. Change the time in the second "move ()" block as you do not want to land at the asteroid but a bit past it.

The goal of this task is to fly within a certain orbital plane. It will be measured by time and best accuracy.

Orbit an Asteroid and Return to Earth (Practice for Competition Event #4)

This will likely be the most difficult task as you must develop your code from the multiple previous coding activities. You must take off from Earth travel to the asteroid, orbit it, return to Earth, and land.

1. As the beginning code you can chose to use the Flying a Circle, Using a Loop to Fly a Circle, or start from scratch.
2. To this you will need to make sure your drone is at the appropriate height, so adjust height after take-off to fly within the asteroid's orbital plane.
3. After adjusting height, you will need to fly to the asteroid, before starting your orbit around it.
4. After completing the orbit, you will need to fly back to Earth – you may have to make some adjustments to the drone's orientation as it comes out of the orbit to point it back towards Earth.
5. Try to land as close to Earth as possible.

The goal is to fly to and orbit an asteroid and fly back to Earth. Accuracy and time are measured.

Be sure to save your successful coding programs for these practices! Name them accordingly.

Module 5: CoDrone Remote-Control

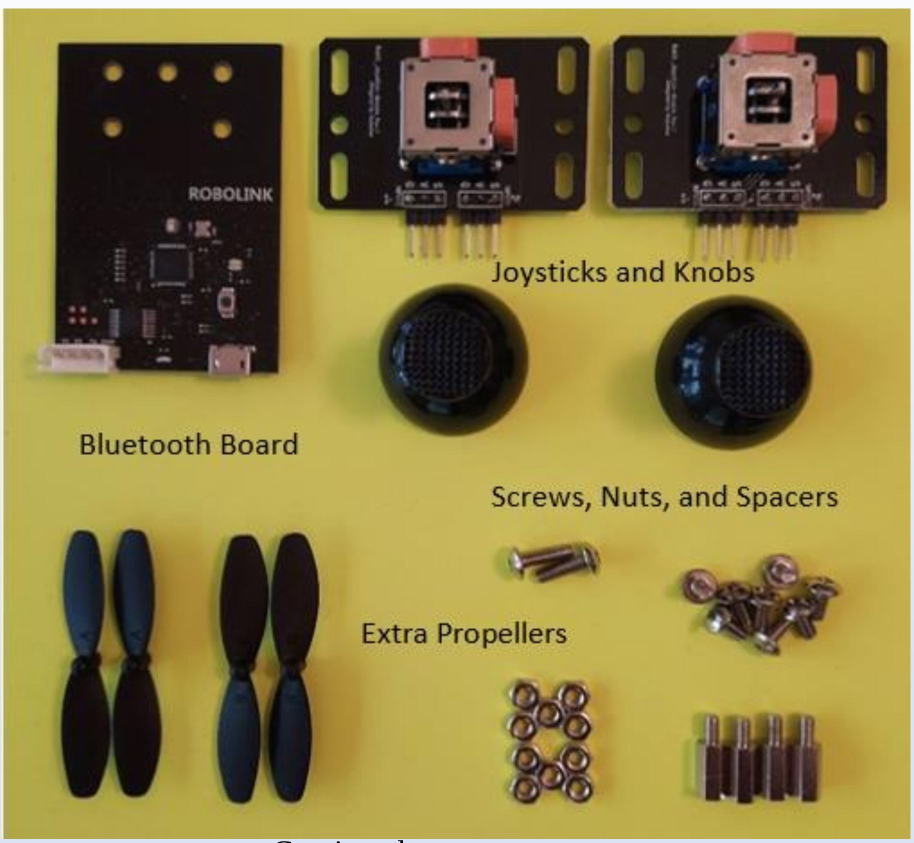
So far, you have only controlled your drone with autonomous programming. Most drones come with a preprogrammed remote-control to operate the flight commands of the drone. We could program your computer keyboard to act as a remote for your drone, but the CoDrone Pro comes with a remote to build and program.

5.1 Build the CoDrone Remote-Control

Let's build a remote for the CoDrone!

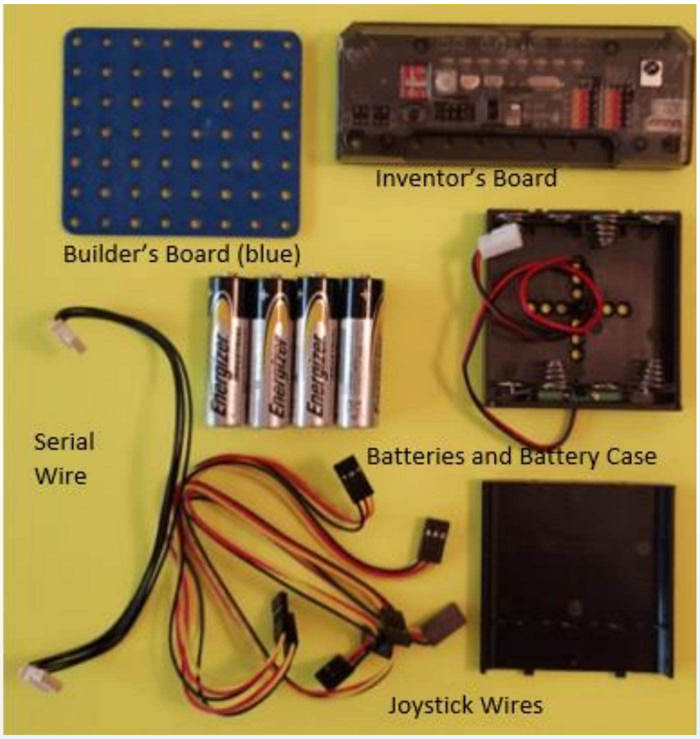
Assembly: Building the CoDrone Remote-Control

1. In the CoDrone box, you will find all the pieces for building the remote-control.
2. Take care when opening the packages not to lose any of the pieces. Lay them out on the provided foam mat.
3. Codrone Remote Control parts:



Continued next page.

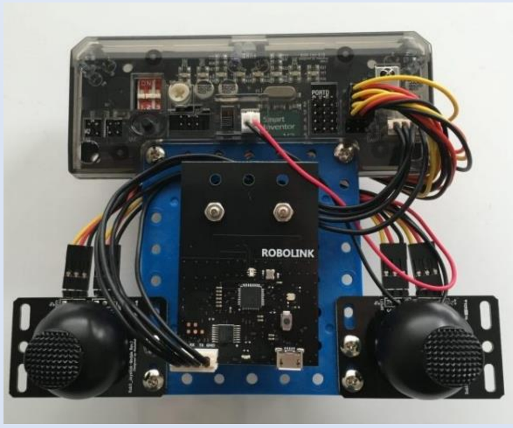
Assembly: Building the CoDrone Remote-Control



4. Use the following resources to guide you through the build process:
   - Short video: https://youtu.be/mDDdWtQXyF4
   - Detailed video: https://youtu.be/HAXq1gBO3I8
   - Pictorial guide: https://learn.robolink.com/lesson/codrone-arduino-build-the-remote/

   It is advisable to look at all the resources to avoid build complications.

5. Here are the basic steps:
   - Attach the SIB to the blue board,
   - Attach the joysticks to the blue board,
   - Attach the battery case to the blue board,
   - Attach the Bluetooth board to the blue board,
   - Wire the battery,
   - Wire the joysticks,
   - Wire the serial connection,
   - Turn on and test.



5.2 Features of the CoDrone Remote-Control

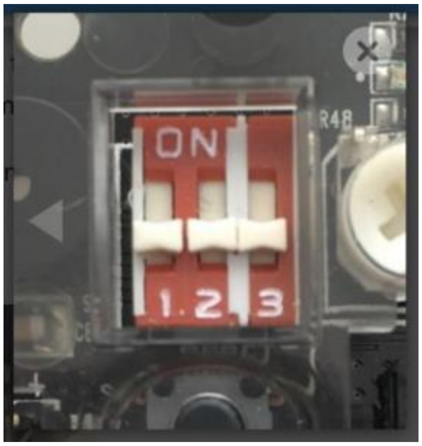
Now that the remote-control is assembled, check out the following resources to learn how to pair the CoDrone to the remote, the special functions of the remote, and how to fly using the remote.

Flying the CoDrone with the Remote-Control (Video): https://youtu.be/Ujkp8QSI5J4

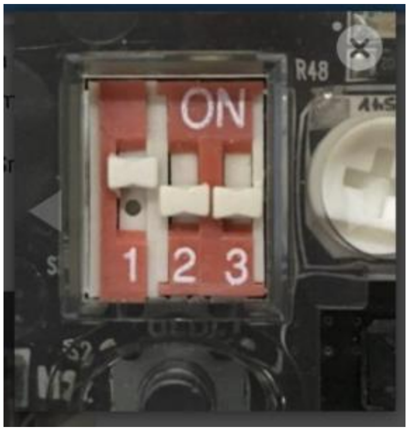
The following information is from the Robolink Basecamp on the features of the remote-control and can be found here: https://learn.robolink.com/lesson/codrone-arduino-using-the-remote/

DIP Switches:

The three small switches on the lefthand side of your Smart Inventor board on the remote are called DIP switches.



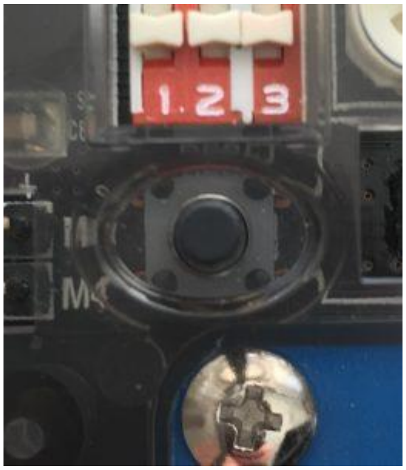
- OFF is at the bottom of the red panel, and ON is at the top of panel. Hint: it says on!
- If you want to run your code, all three DIP switches should be off. This is called run mode.



- If you want to upload your code, put switch 1 in the on position and switches 2 and 3 in the off position. This is called upload mode.

To upload code, you must make sure that both the Smart Inventor board and the Bluetooth module are in 'Upload Mode'.

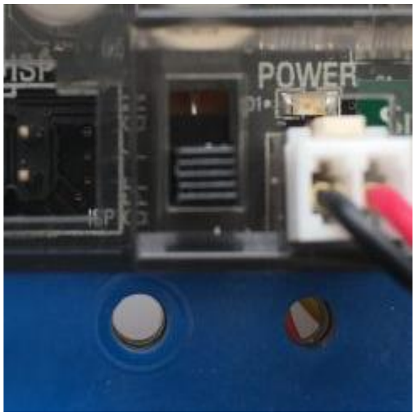
We will NOT upload code to the remote for the AIR Challenge, but it is important to check to be sure the switches are in the right location.



Reset Button:

The reset button is also on the lefthand side of the Smart Inventor Board, right underneath the DIP switches. If this button is pressed and all the DIP switches are off, the Smart Inventor Board will automatically start whatever program was last uploaded. Just as a warning, it does not reset the Bluetooth module!

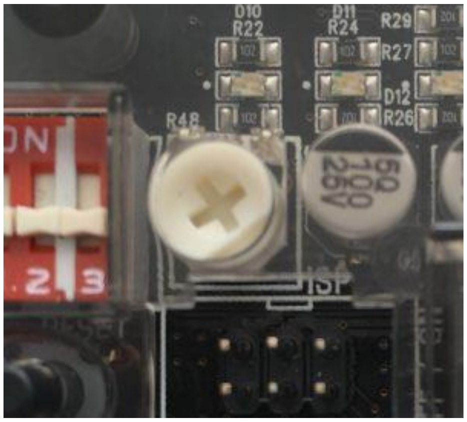
Power Switch:



The power switch is in the middle of the Smart Inventor Board. When this switch is flipped on, it connects the battery pack to the Smart Inventor Board. If you don't have AA batteries for your battery pack, you can plug the USB cable into your computer to power the remote.

Make sure you turn your remote off when you're switching between upload mode and run mode! Just flip the power switch to off and disconnect your USB cable.

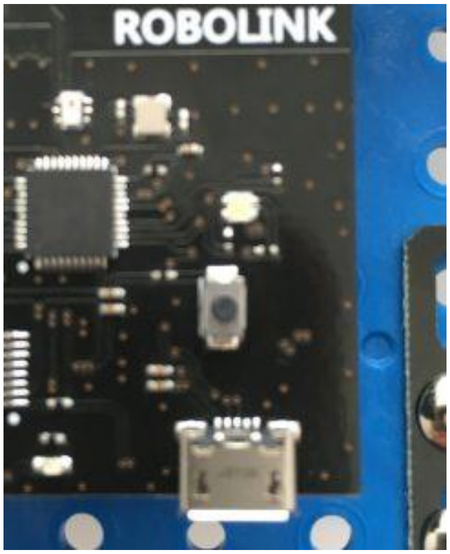
Sensor Sensitivity:



Next to the DIP switches is a white knob that is a perfect fit for a Phillips screwdriver. This is the potentiometer (adjustable resistor) and it controls the sensitivity of the IR sensors on the Smart Inventor Board. Turning the potentiometer clockwise will increase sensitivity and turning it counterclockwise will decrease sensitivity.

Warning: the potentiometer needs to be turned very carefully! You should be able to feel the upper and lower limits when you're turning the potentiometer, and it cannot make a full 360-degree turn. If the potentiometer is turned too far, it will break, and the IR sensors will not work correctly. Ask for assistance.

Bluetooth Module:



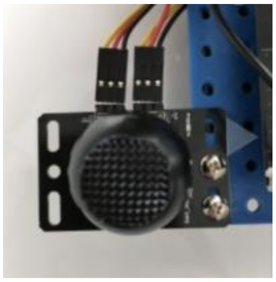
The Bluetooth module on your remote control's blueboard turns the code you write into wireless Bluetooth signals that the CoDrone can receive.

The small black button on the Bluetooth module is its reset button. Pressing this puts the Bluetooth module into upload mode. If you would like to upload your program to the Smart Inventor Board, make sure it's in upload mode (DIP switch 1 on, DIP switches 2 and 3 off)! If this all works correctly, one blue LED on the Smart Inventor Board should be flashing.

Joysticks:

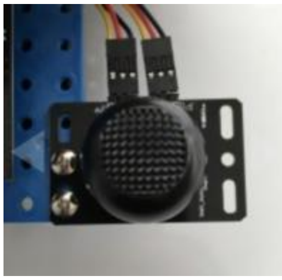
The joysticks use the movements of your fingers to output a signal which the Smart Inventor Board can read. Inside each joystick are two potentiometers: one for the x (horizontal) direction and one for the y (vertical) direction.

Left joystick



- Located on the left side of the controller.
- Vertical movement controls your CoDrone's throttle, or how high it goes.
- Horizontal movement controls your CoDrone's yaw, or which way it turns.

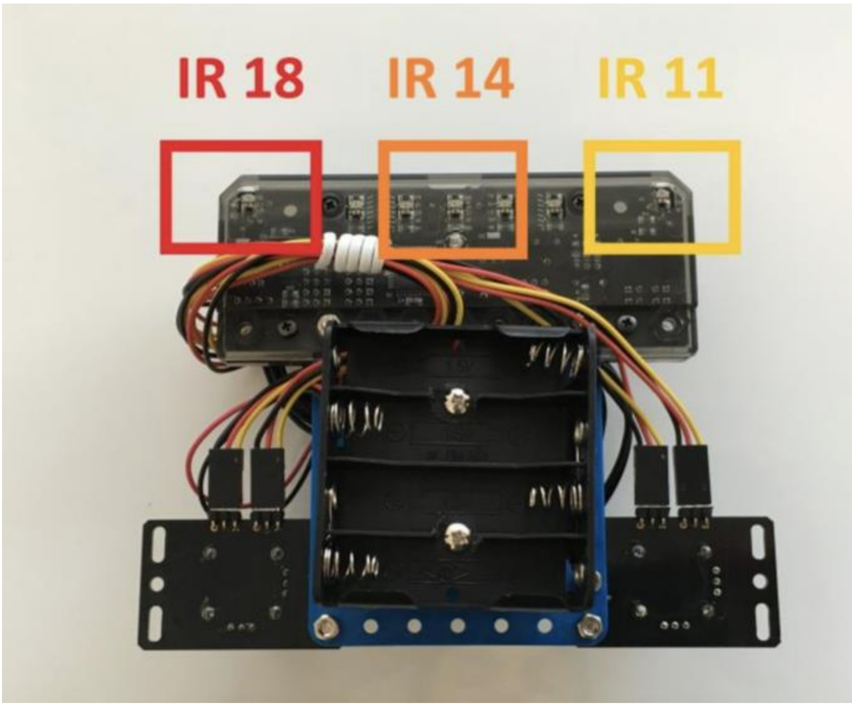
Right Joystick



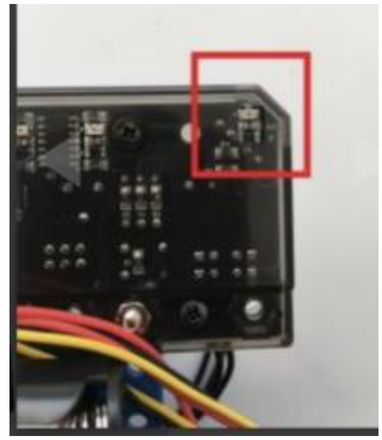
- Located on the right side of the controller.
- Vertical movement controls your CoDrone's pitch, or its movement forwards and backwards.
- Horizontal movement will control your CoDrone's roll, or its side-to-side movement.

Infrared Sensors:

The Smart Inventor Board uses infrared light to detect your fingers! When you cover one of the infrared sensors, the infrared light bounces off your fingers and shines on the sensor. Sunlight can also activate the sensors, so be careful in areas with a lot of direct sunlight.



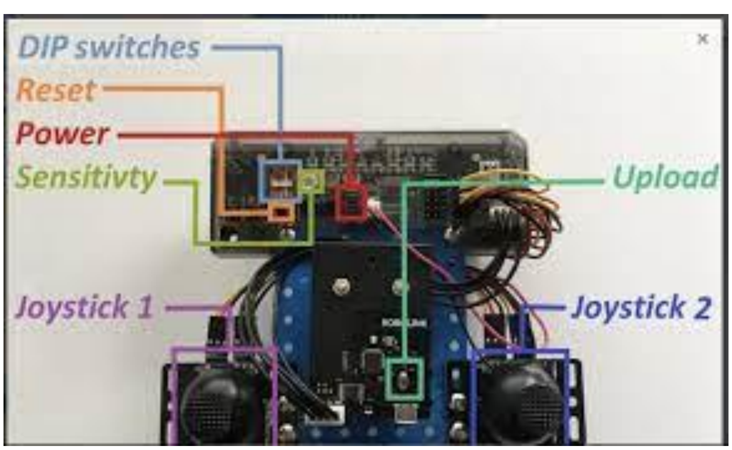
- The bottom left IR sensor is labeled IR 11 and is connected to Smart Inventor Board digital pin 11. In the default flight program, covering this sensor immediately stops your CoDrone's motors so it stops mid-air. **This is your emergency kill switch!**



- The bottom center IR sensor is labeled IR 14 and is connected to Smart Inventor Board digital pin 14. In the default flight program, this sensor doesn't do anything.

- The bottom right IR sensor is labeled IR 18 and is connected to Smart Inventor Board digital pin 18. **In the default flight program, covering this sensor initiates the landing command so your CoDrone slowly descends and lands.**

Sunlight may trigger the infrared sensors and accidentally trigger commands. The bottom IR sensors' sensitivity can be adjusted with the white potentiometer.



5.3 Star Ship Communication (Flying with Remote Control)

Now it is time to test out your remote-control skills and practice flying the CoDrone with the remote. Where the autonomous flying relies completely on coding skills, flying the drone with the remote relies on your dexterity and spatial awareness of the course.

Before we fly with the remote, **the drone must be paired with your remote**, but the process is a bit different then when we paired the drone with the computer.

The pairing process is automatic, but we need to be careful. When we do it the first time, we need to ensure that you know what to look for and what to do when things do not look right.

The following steps are to be performed in order by each pilot – **One by One!**

Read through all the instructions first:

1. Insert a charged battery in the CoDrone.

2. Place the CoDrone on the Drone Target Sheet. (The rear green LED should flash twice)

3. Turn on the remote-control:

- Initially the LED on the Bluetooth board blinks orange, then red.

- Initially the row of blue LEDs shows some kind of "inward motion".

- The drone (if on and close enough to the remote-control) should bind: the rear LED should become solid green, the LED on the Bluetooth board should flash somewhat greenish (with other colors in between), and the blue LEDs should "stop moving, two blue LED should remain solid. This indicates the drone is paired to the remote-control.

- If the drone does not bind, ask somebody for help.

Communication Protocol:

Air Traffic Controller (ATC)	Pilot
Crew 1A power drone!	Battery is in, 1A
Crew 1A power remote-control	RC ON, 1A
	RC Paired Successfully, 1A
	(or, if it doesn't Malfunction, 1A)
(Next Crew)	

We will be using this communication protocol for the CoDrones as well as for the Star Wars drones. In the classroom, we may be doing this only once, but in the gym, it is important to quickly perform the operation – otherwise nobody can fly!

4. **Be prepared:**

   - Have your thumbs touch the joysticks – but do not move the joysticks, yet, until you have read the last bullet point in this list!
   - Keep your other fingers from being below the "front" of the remote-control: there are the sensors for the emergency stop!!
   - Be ready any time you fly to move the pointing finger of your left hand underneath the front left corner of the remote-control. This will stop the propellers in an emergency.
   - Remember once you want to land the drone, just move the pointing finger of your right hand underneath the front right corner of the remote-control.

5. Take off: move gently (using your thumb) the left joystick forward: That is the throttle – do you remember this lifts the drone off the ground … and later when you pull back on the joystick cause the drone to descend.

6. Correct any drift with the right joystick.

7. Use the left joystick for yaw: rotate the drone by gently moving the left joystick left … and then right: always observe the drone! Do not look at the remote control!

8. Land. – Remember there are two ways to land the drone using the remote – use the soft landing when it is not an emergency.

How was your first remote controlled flight?

- Did the drone pair?
- Did it take off?
- Was the drone steady hovering without you touching the joysticks?
- Could you move the drifting drone back over the landing pad?
- Did you land successfully?
- Did you turn the motors off?
- Did you take the battery out from the drone after completing your flight tests?
- Did you turn off the remote-control after completing your flight tests?

Flight Practice 3: Remote Controlled Flight

Time to practice flying with the remote in an open area (gym). You will practice some of the same flight courses you did earlier but this time using a remote to control the drone's flight instead of flying autonomously.

From Earth to Mars:

1. Activate your CoDrone.
2. Place the drone on the pad with Earth's image.
3. Make sure nobody else gets ready: **For pairing you must make sure you bind to your drone, not to somebody else's.**
4. Turn on the remote-control. Watch and check on the binding process.
5. If successfully paired, take off, fly to Mars and land on it.
6. How close did you drone land to Mars? (Inside the hoop or outside?)

To Mars and Back:

1. Activate your CoDrone.
2. Place the drone on Earth.
3. Make sure nobody else gets ready: **For pairing you must make sure you bind to your drone, not to somebody else's.**
4. Turn on the remote-control. Watch and check on the binding process.
5. If successfully paired, take off, fly to Mars and hover for 2 seconds over it.
6. Rotate the drone by 180 degrees.
7. Fly back to Earth and land on it.
8. Did you rotate over Mars?
9. How close did you drone land to Earth? (Inside the hoop or outside?)

Orbit an Asteroid and Return to Earth (Competition Event #5)

1. Ask somebody to replace Mars with an asteroid.
2. Activate your CoDrone.
3. Place the drone on Earth.
4. Make sure nobody else gets ready: **For pairing you must make sure you bind to your drone, not to somebody else's.**
5. Turn on the remote-control. Watch and check on the binding process.
6. If successfully paired, take off, fly to the left side of the asteroid.
7. Fly around the asteroid – behind it – until the drone faces you at the right side of the asteroid.
8. Fly back to Earth and land on it.
9. Did you make it successfully around the asteroid?
10. How close did you land to Earth? (Inside or outside the hoop?)

The goal is to fly to and orbit an asteroid and fly back to Earth. Accuracy and time are measured.

5.4 Optional Flight Simulator Contest: Amazon Trek

Can you conquer a flight to the Amazon with hidden dangers and obstacles? Jump into this optional activity suited students who are looking for an additional challenge. It can be done at any time during the Summer Institute. The FSX Simulator Mission Midwest Fly-In lesson is a prerequisite for this activity.

Directions for the Amazon Trek:

1. On one of the PC laptops start Steam.

2. Login as stemsealsxx with password stemseals1xx.

3. If you forgot how it works, go back to the Midwest Fly-In instructions in this book.

4. Before you fly let’s recap the Flight Simulator commands you learned when you flew the Piper Cup in the Midwest Fly-In mission:

- F9 key: cockpit view
- F11 key: spot plane view
- P: pause simulation
- G: gear up / down
- “.”: brakes, release / brake
- arrow keys on Num Pad: change view
- “5” key on Num Pad: center view
- lever behind joystick: throttle

5. Start the Flight Simulator.

6. Choose the “Missions” tab at the upper left.

7. Choose “Amazon Trek” (scroll down!).

8. Click “Go to Briefing.”

9. **Read all!** It is important. This is a challenging mission on many levels. Take notes if you see it appropriate.

10. After reading the “Overview” click on the “Details” tab and read carefully!

11. Click the “Maps & Charts” tab and look at the map: To the lower right is the airport at Quicemil. Note its altitude. At the upper right, there is a small lake – this is where we go first, and a long, large lake, which looks more like a large river. Remember this for later.

12. Go one more time over the briefing information. You need a few more flight simulator commands in addition to the general commands to successfully fly a floatplane:

- Press G on the keyboard to lower or raise the gear.
- Shift-6 displays the overhead panel to lower or raise the wingtip floats.

How to control the engine:

- E1 (press the E key and after the number 1): activate differential thrust: left engine only.
- E2 (press the E key and after the number 2): activate differential thrust: right engine only.
- E12: deactivate differential thrust: the thrust level now affects both engines equally.

How to operate the flaps:

- F5 (pressing the F5 key on the keyboard) sets the flaps to zero degrees (meaning no flaps).
- F7 increases the flap angle by 30 degrees.
- F6 decreases the flap angle by 30 degrees.
- Pressing F8 increases the flaps to maximum. We don't need this command on this mission.

13. Ready? If yes, then click the "Fly!" button.

Cockpit view of the Grumman Goose. The four flight instruments you see here are sufficient for this mission: Flaps, Airspeed, Attitude Indicator, and Altimeter.

14. You are on a runway without any markings. Before you take off look at your plane (F11). The Grumman Goose is pretty much a flying boat, which means be careful how you fly! Take-off requires to pull the stick real hard toward you. Switch back to cockpit view (F9) and take off!

15. Make a left turn – Do you see the compass at the upper left with the mission pointer?

16. The weather is bad. The ceiling (the underside of the clouds) is about at 8000 feet altitude. It is a good idea to stay below to see a little better.

17. Did you remember to raise the gear after take-off?

18. Did you set the flaps to zero degrees once the plane had enough airspeed?

19. Those first 25 nautical miles to the base camp seem boring. Use the time to get familiar with the aircraft:

- Stay on course!
- Hold your desired cruising altitude.
- 4000 feet is a good value for this mission.
- About 130 knots is a good cruising speed for this aircraft.

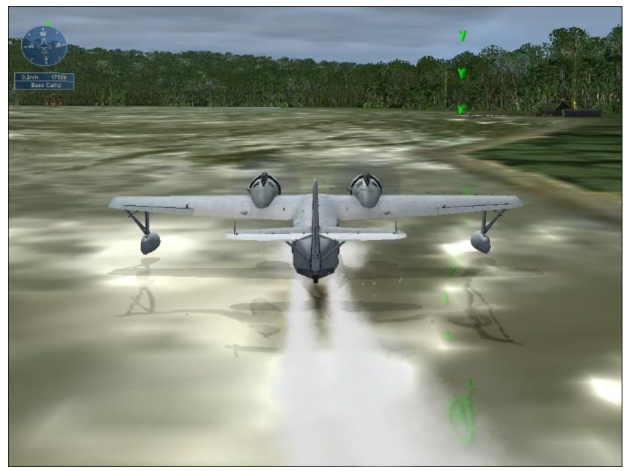
20. At about 7 nautical miles from the base camp, you should be able to see the target pointer in the clouds (if you are low enough!). Aiming a little right of the target pointer is a good idea – there is a tall waterfall on the left!

21. Land on the lake (use 30-degree flaps).

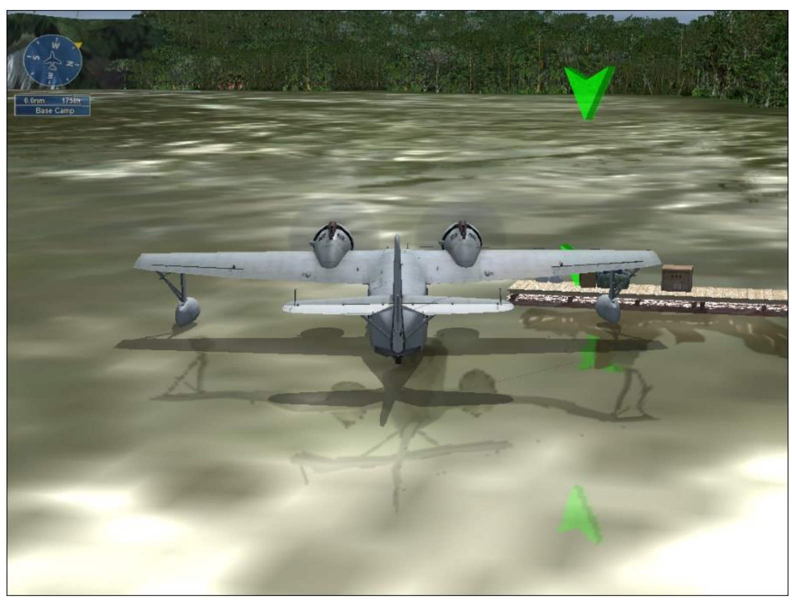
22. Lower the wingtip floats (shift 6).

23. And taxi to the dock.

24. Approach the dock from the side. Remember you are on water, and airplanes don't have a reverse. You must be able to leave the dock without crashing into it. You do not need to worry about the wingtip floats (in the simulator).



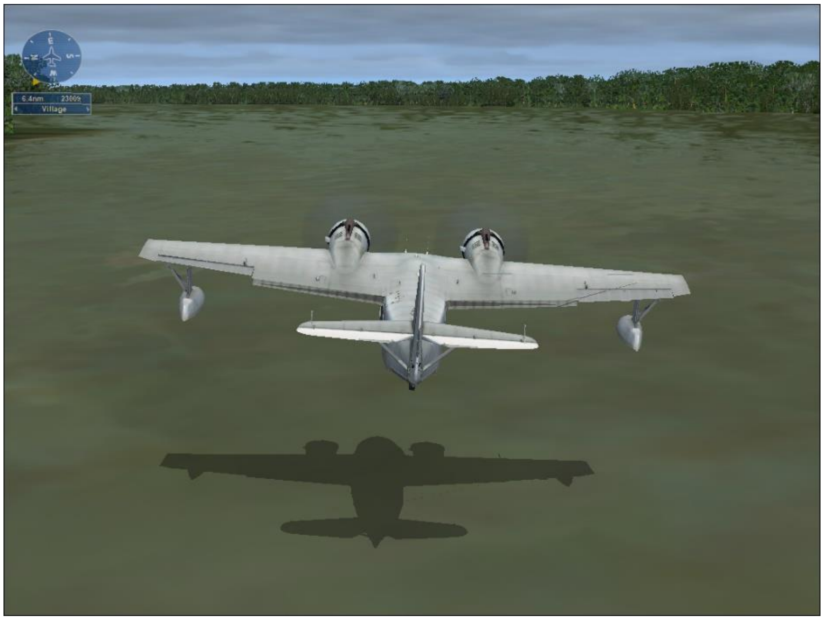
25. You must be very close to the dock to successfully pick up the baggage.



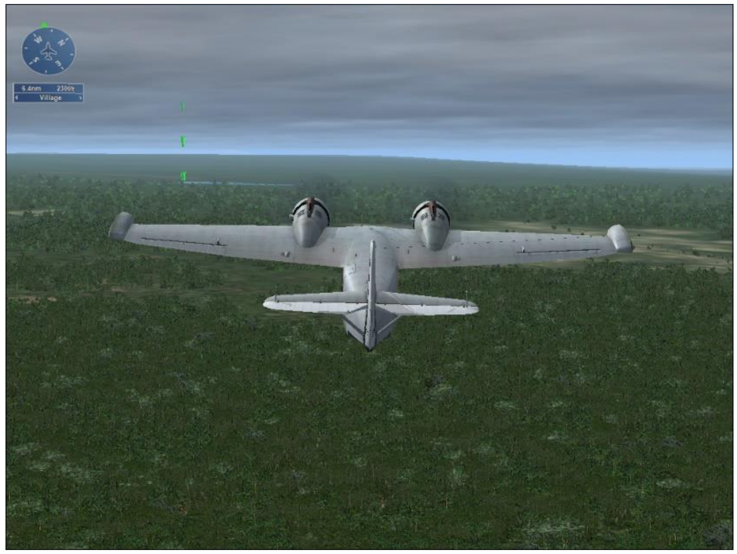
26. Once the baggage is not on the dock anymore, you are ready to take off again. Listen to the pilot.

27. Taxi to the end of the lake (the end straight ahead of you), turn around…..

and take off.



28. The lake at the village is much smaller than the previous lake. You need to approach again from the right to have the full length of the lake available for landing.



29. Land along the long direction of the lake – you need the space to get down.

30. Taxi past the dock to approach it again alongside the shore. This enables you later to continue to prepare take-off.

31. Before take-off you must decide. If you are up to going to the Temple of the Anaconda.

32. Taxi again to the end of the lake ahead of you, turn around and take off.

33. Make a turn to the SW. It is only 8 nautical miles to the temple.

34. Follow the directions for a while … But – Oops! – the directions were not accurate!

35. It rains heavier. Try to stay under 4000’ to be able to see.

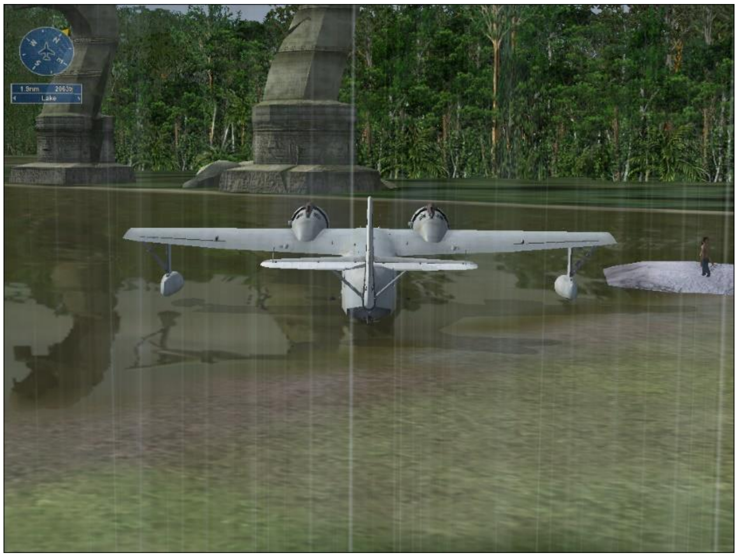
36. When the archeologist says: “Try further to the south!”, do so (and don’t follow the mission pointer)! Shortly after you will see the lake, which looks like a large river.

37. Fly to the river and follow it towards the south. Lower your altitude to prepare for landing.

38. It is difficult at first to see the entrance to the temple. Fly low along the river and look for something unusual on the left shore.

39. Once you see it there is no doubt: Giant snake statues! Land on the river and taxi to the entrance.

40. Taxi between the statues towards the temple. Note: In front of the temple is a flat rock in the water. The entrance is tight for a floatplane. It is possible to turn the plane in the water. But wait: the Grumman Goose has wheels! There is another way to prepare for take-off: Taxi right up to the shore to the right of the flat rock. Lower the gear and taxi on land.



41. Once on land it is much easier to turn the plane on wheels! Be careful with the throttle. Make a tight turn until you face the water and the passageway out. Stop!

42. The archeologist will go on shore. Listen carefully!

43. She comes running back … and you need to take off immediately! You can look back at the temple. But don't waist too much time – the journey back will be hard!

44. Taxi out of the temple entrance, turn to the right on the river and take off with no delay.

45. Follow the mission pointer and fly back to the airport. Listen carefully what happens. You need to keep the plane in the air, keep a safe altitude, and you need to keep course. It will get worse, and there is no time to waste!

46. With instrument failures and one engine with an oil leak keep course towards the airport!

47. Don’t trust the instruments! Every time you get a glimpse of the target marker: this is the direction to fly. Do your best, and don’t give up! Be prepared for the worst to happen.

48. The direction of the runway is towards the mountains on the right. Approach by aiming towards the wooded hill on the left of the target marker.

49. With a lost right engine, the plane still flies. Increase throttle and compensate with the joystick to keep course.

50. Close to the runway, with the left engine on fire, the situation has become scary. Do not give up!

51. When landing: do not use flaps! They also don’t work right. You can try but set them back to zero degrees immediately when the plane becomes unstable.

52. Now aim to align with the runway. You need to land but you also don’t want to crash. Do the best you can – and you can make it. Don’t bother with details!

53. Over the runway reduce throttle all the way down, keep the plane level and put it gently on the ground.

54. Brake! (“.” or the front button on the joystick) The aircraft needs to stop.

Did you make it to the end?

If you did or did not, answer the following questions:

- Did you pick up the baggage at the base camp?
- Did you pick up the archaeologist at the village?
- Did you find the entrance to the Temple of the Anaconda?
- Did the archaeologist go on land to the temple?
- Did she come back?
- Could you take off again after visiting the temple?
- Did you make it halfway back to the airport?
  - Three quarters?
  - All the way?
- Yes: Great! You are a champion!
- If you answered any of the questions above with no, try again to fly the mission. Before you do, revisit all flight commands needed, and find somebody to ask questions what to do in the situation, which caused the mission to fail.

Module 6: Flying Other Drones

Now that you have learned about the CoDrone, how about trying out your flying skills on another drone. The CoDrone is a very light mini drone and should only be flown indoors. Drones come in many sizes and have many uses.

Think about It: What are drones used for?

How many different uses, applications, or careers can you think of using drones?

What kind of training do you think it requires to fly drones professionally?

What kind of issues do you think could arise as the use of drones increase in society?

Do you think you would enjoy a career that involves drones?

6.1 Learning About Careers Using Drones

To learn more about drone careers, Check out these internet resources:

Drone Careers:

- UAV Coach website: https://uavcoach.com/uav-jobs/

7 Unexpected Drone Career Paths:

- The Drone Girl website: https://www.thedronegirl.com/2020/08/10/drone-career-paths/

Is Drone Pilot a Good Career?

- Drone's Gator website: https://dronesgator.com/drone-pilot-career-demand/#:~:text=Becoming%20a%20professional%20drone%20pilot,opening%20up%20for%20drone%20enthusiasts.

6.2 The STAR WARS Drones

While there are many types, sizes, and uses for drones for careers, sometimes it is fun just to fly drones for fun! It is time now to check out another cool drone developed solely for recreational purposes.

The STAR WARS drones have a bit more power and are larger than the CoDrone but to minimize drone retrieval, we will fly in the gymnasium. However, these drones can be flown outside but it is recommended to only do so on a calm day with little wind.

As with any new device, always read over the instructions that come with it.

There is an instruction booklet included with the STAR WARS drone.

Anytime you fly a different drone, you should look at these things:

Are they faster?

Are they easier to fly?

Can they do something else than just fly?

However, no matter what type of drone you fly, it is most important to always look over the instructions. Treat all drones with care, especially the STAR Wars drones as they are now collectible drones and not easily replaced.

6.3 The STAR Wars Remote Control

Let’s first look at the remote control of the STAR Wars Drone. It is different than the CoDrone remote.

Note: You need to learn the following about the remote control. Do NOT operate your drone or remote-control until you are told to do so!

This is the remote for the STAR Wars drones. The black remote is for the Tie Fighter and the X-Wing Fighter remote is white, but they operate in the same way.

There are manuals that come with the drones to inform you of all of the features and details. Here we will walk through a few of the basic details.

The power button is the big red button.

- Press and hold 2 seconds to turn on.
- Press and hold 2 seconds to turn off.

When powered on, the remote-control will vibrate, light up the LEDs and play a Star Wars soundtrack. If nothing happens ask somebody to check the batteries.

There are two joysticks:

Throttle: Left joystick forward and back

Yaw: Left joystick left and right

Pitch: Right joystick forward and back

Roll: Right joystick left and right

Sync to Ship is needed after power on.

This is similar to pairing with the CoDrone.

1.) Turn on remote-control,
2.) turn on star ship,
3.) left joystick up: listen for one beep,
4.) left joystick down: listen for another two beeps:

The LEDs around the power button will light up: this indicates the selected speed mode.

There are four arc-shaped buttons between the two joysticks.

Calibrate Drone

This is needed after a hard landing or crash, or you may just calibrate the gyro to ensure the drone does not drift on its own.

With the ship on a flat surface with its motors stopped, press, and hold the upper right arc-shaped button for 2 seconds to calibrate.

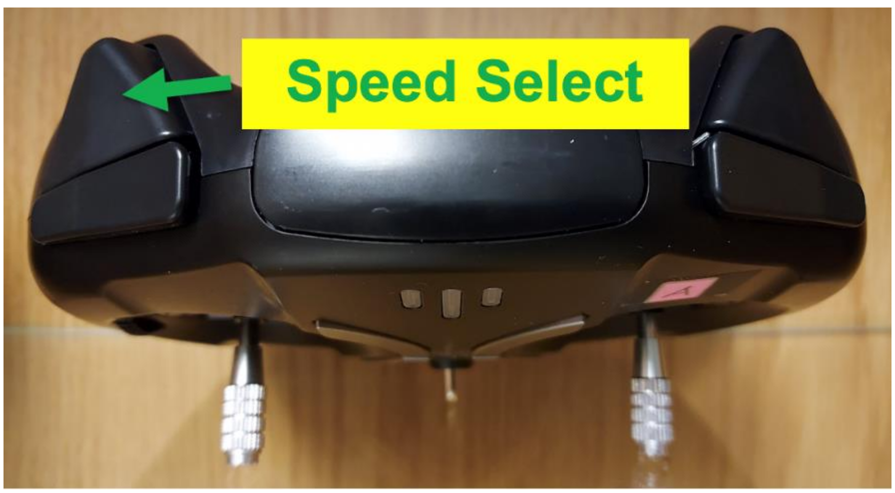
Stabilization of the drone's altitude can be adjusted with the small wheel to the left of the left joystick. It is called Repulsor Lift. When the Repulsor Lift is on, the star ship uses a barometric pressure sensor to stabilize its altitude simulating the anti-gravity effect of repulson engines. You can then adjust the altitude incrementally.

It is easier to fly with Repulsor Lift on, but for battle and stunts the star ship is more agile with Repulsor Lift off.

Repulsor → ON

For now, we will be flying with Repulsor ON.

Turn the small wheel all the way toward you.



The Star Wars remote-control has four more buttons at the front. Take the remote-control in both hands as if you would be using the joysticks and rotate the device towards you such you can see the front – and the joysticks point downwards.

There are two large buttons:
- Speed Select: left
- Fire: right

and two small buttons:
- Flip left: left
- Flip right: right

Enter T-Mode

Press and hold the Speed Select button (left large) for 3 seconds. (Repulsor Lift must be on!)

If all lights around the Power button turn blue you are in T-Mode. This means the drone will not fly higher than 2 meters, will not touch the ground and not go further than 100 feet. Press (and don't hold!) the Speed button if you want to turn T-Mode off.

6.4 Star Wars Drone Communication

Like what you already learned about pairing of the CoDrone, radio communications between the Star Wars drone and a remote controller require what is called "syncing".

We will use the same communication protocol we used for the CoDrones to ensure that your drone is synchronized to your remote-control and not to somebody else's.

Let's first look at the communication protocol and then sync one drone after the other:

Communication Protocol:

Air Traffic Controller (ATC)	Pilot
Crew 1A power drone!	Battery is in, 1A
Crew 1A power remote-control	RC ON, 1A
	RC Paired Successfully, 1A
	(Or, if it doesn't Malfunction, 1A)
(Next Crew)	

Flight Practice 4: Star Wars Drones

Re-read the instructions above about the remote-control. You will need them all the time! You don't yet know everything, about the drone's capabilities but we are ready now for a test flight in the gym.

Let's use some of the same courses to practice flying the Star Wars Drones.

Note: There are not enough Star Wars drones for each student. Therefore, you may end up flying either of the two drones: Star Wars X-Wing Fighter or the Star Wars Tie Fighter. During flight practice it is encouraged you try out both types.

From Earth to Mars: (Read through directions first)

1. Place the two disks with the Earth and Mars images the same way as we did when we tested the CoDrone. Place a hoop around both disks.
2. Turn the star ship on (insert the battery).
3. Place it in the center of the Earth, facing the direction where Mars is.
4. Turn the remote-control on (push the red button).
5. Sync the star ship (left joystick up, then down).
6. Optional: Calibrate (press the upper right arc-shaped button for 2 sec).
7. Activate Repulsor Lift (small wheel towards you).
8. Enter T-Mode (press Speed Selector/left large button and hold for 3 sec).
9. Be ready: thumbs on both joysticks!
10. One more button to know: **Auto take-off and land:** The lower (towards you!) left arc-shaped button initiates an automatic take-off or landing procedure. That's what we need now - wait: You first mission goal is to fly to Mars and land of it. Here we go:
11. Auto take-off: press the lower left arc-shaped button.
12. Fly straight to Mars.
13. Land in the middle of it.

Did you crash?
Did the star ship find Mars?
How far are you off the middle?

To Mars and Back:

Let us see if you can also fly to Mars and back to Earth.

1. Place drone on Earth.
2. Power on drone.
3. Turn on RC.
4. Sync
5. Activate Repulsor Lift.
6. Enter T-Mode.
7. Fly to Mars and back to Earth. Land.

Did you crash?
Did you land on Earth successfully?
How far off the middle did you land?

To Mars and Back (Faster)

Let us see if you can perform the same flight faster.
Look at the LEDs around the Power button! There are three speed levels. Pressing the button increases the speed level, pressing the button in Speed 3 mode cycles back to Speed 1. (Red LED lights mean remote-control is synced to star ship.)

1. Place drone on Earth.
2. Power on drone.
3. Turn on RC.
4. Sync.
5. Activate Repulsor Lift.
6. Press the Speed Selector button (left large).
7. Fly to Mars and back to Earth. Land.

Did you crash?
Did you land on Earth successfully?
How far off the middle did you land?
Did you go faster?

You may want to try a different speed level and fly again.

Turn Repulsor Lift off (small wheel forward) and fly again.

What is better, Repulsor Lift on or off?

Orbit an Asteroid and Return to Earth

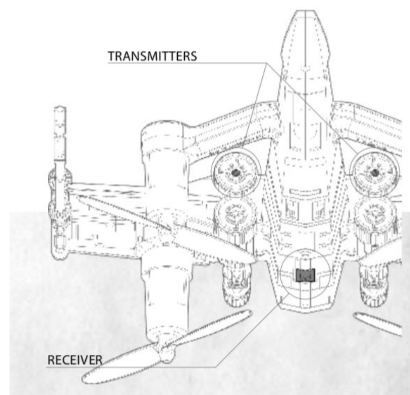
Let us see if you can fly to an asteroid and come back home.

1. Replace the Mars disk with an asteroid.
2. Place your drone on Earth.
3. Power on drone.
4. Turn on RC.
5. Sync.
6. Choose a Speed Mode.
7. Fly to the left of the asteroid, orbit around behind it and return to Earth. Land.

Did you crash?
Did you land on Earth successfully?
How far off the middle did you land?
Could you go faster?

Did you use Repulsor Lift?
You may want to try a different speed level and fly again.

Star Wars Battle



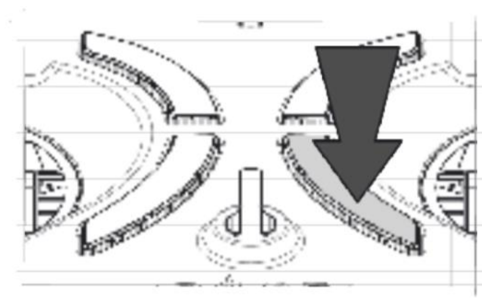
The Star Wars drones are equipped with infrared (IR) transmitters capable of shooting invisible light at opponents' drones. They are located at the front of the star ship. They transmit a wide signal beam.

Underneath the ship are IR receivers, which allow simulation of getting hit by another drone. If you aim anywhere beneath your target ship from a range of less than two to three meters, you will most likely score a hit.

When battling indoors, IR signals can bounce off walls, especially white walls. You can sometimes accomplish event even if you were not directly aiming at your opponent if you fire using the correct angle.

When IR battling, you will not be able to battle with any Star Wars ships that are set to use auxiliary weapons.

Auxiliary Weapons Activation



The lower (closer to you) right arc-shaped button of the remote-control- toggles aux weapons on or off. Press and hold for 4 seconds to activate or deactivate auxiliary weapons. When auxiliary weapons are activated, the IR transmitter is disabled, and you will not be able to participate in battle with your star ship.

Note: If you only press the lower right arc-shaped button for one second, the white LEDs of you star ship can be turned on or off.

Weapons Fire

The weapons fire button is the large right button in front of the remote-control. When this button is pressed, the weapons of the star ship fire (IR transmitters fire when Aux Weapons are deactivated) and you will hear a sound from the controller's speaker. You must press the button again to place a second shot.

When a star ship is hit by enemy fire, it can withstand two hits; the third hit disables the spacecraft. When a ship is hit the first time, it will rock back and forth two times, the controller will vibrate, play a sound, and indicate the ships status with the Life Status Indicator LEDs. Also, the player who has struck your ship will receive a notification from his/her controller of a successful hit.

When a star ship is hit the second time, it will rock more violently, the controller will vibrate, play a sound, and change the Life Status Indicator LED display to only one LED illuminated.

The third hit is the final hit in battle. The ship will auto-land in a downward spiral. The throttle stick will be disabled for 10 seconds. All other joystick motions remain enabled allowing for controlled landing. After 10 seconds on the ground, the drone will be reset and re-enabled for battle.

Aircraft Battle Maneuvers

Before you enter battle, you would like to be informed about some aircraft battle maneuvers, either offensive maneuvers for attack or defensive maneuvers to protect your star ship. Some clever maneuvers combine both, defense to get out of the fire, but at the same time getting in a better position for attack.

The maneuvers can be found in the manual for the Star Wars drone. These will be available to review. Some of the maneuvers are:

Circular Strafe (Offensive)

Drop Back and Attack (Defensive)

Reverse Throttle Hop (Defensive, enabling offense)

Smuggler's reverse (Offensive)

Tallon Roll (defensive)

You may wish to practice these a bit before practicing the battle.

Star Wars Battle: (Practice for Competition Event # 6)

1. You will form teams with X-Wing Fighters and Tie Fighter drones. This way you can easily distinguish between friend and foe.
2. Ensure you have a fully charged drone battery at hand.
3. Power on your drone.
4. Place your drone in the center of the Earth at your station. The other members of your squadron and the opponent team will do the same.
5. Turn on RC.
6. Sync.
7. Choose Speed Mode.
8. Wait until all pilots are ready and you received the signal for opening battle.
9. Take off and search for enemy drones.
10. The more enemy drones you shoot down the more you help your team win.
11. **A drone can be hit three times before it is disabled and will land. Once your drone lands, you must exit the floor.**
12. At the end of 6 minutes (approximate drone fly time), the team with the most drones still fighting- wins!
13. At the signal for the end of the battle fly your drone back to Earth (the disk in front of your station) and land.

On competition day, squadrons will compete against each other through a process of elimination. The type of drone you fly will be determined by a draw.

Module 7: Designing a Search and Rescue Mission

Did you notice many differences in the two drones- the CoDrone and the Star Wars drone? While the Star Wars drone was developed solely for recreation the CoDrone was developed with educational purposes allowing the user to fly it autonomously by writing their own code and to even write code for the remote control.

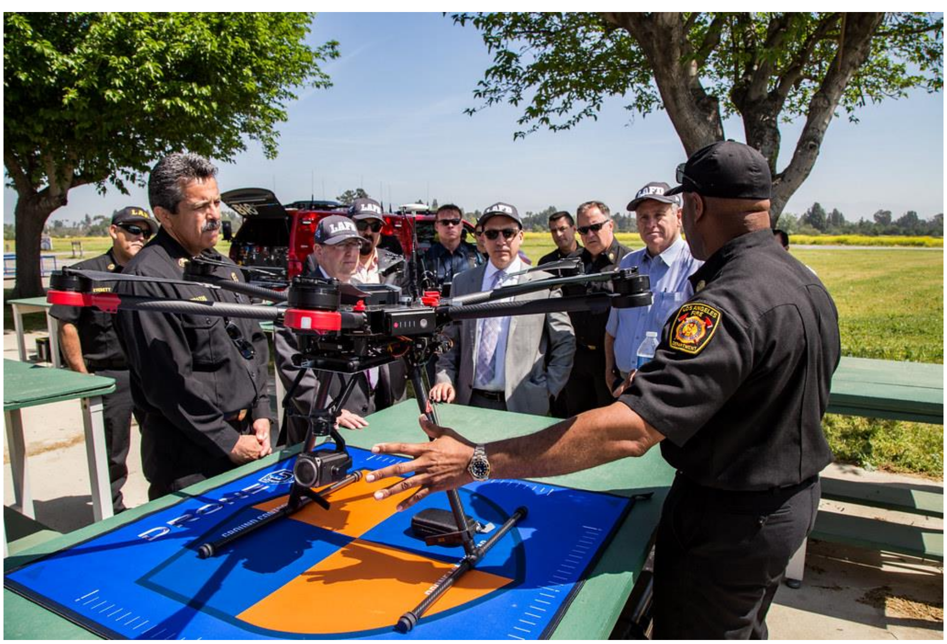
Let's see if we can design and carry out a special purpose for one of these drones.

Do you think it would be possible to create a search and rescue for one of the drones? Which drone do you think would be easier to adapt for a search and rescue mission?

Did you know drones are very beneficial for search and rescue missions? Check out the resources below.

Check out these resources on Search and Rescue with drones:

- https://www.outsideonline.com/outdoor-adventure/exploration-survival/drones-search-rescue/
- https://www.dronefly.com/blogs/news/drones-for-sar-search-and-rescue/
- https://uavcoach.com/search-and-rescue-drones/



7.1 Build a Drone Attachment for the Rescue Mission

The next mission is to find a new task for a drone to do. We want to find out if we can rescue an astronaut, who got stranded on an asteroid.

Just imagine, like any other technological devices, a star ship may show some malfunction and as a result, the astronaut cannot return home safely. He chooses to land on an asteroid to figure out if he can repair the star ship but finds out he does not have the parts.

If his radio equipment still functions, he will send a message home saying he needs help. You receive this message and promptly prepare a rescue mission.

Think about how you would pick up an astronaut … or anything with a drone!

What we will do is

- attach a passive capture device to the astronaut to assist picking him up.
- and attach another capture device to the drone able to grab something, e.g., the astronaut.

3D printed Astronaut

The capture device for the drone will be a fishhook; and the device for the astronaut will be a loop large enough to become hooked to the drone.

Both, including the astronaut, need to be light enough for the drone to be able to provide enough additional lift. All devices, including the astronaut, are 3D printed.

3D printed fishhook

7.2 Testing the Drone Attachment

Let's see if we can create a search and rescue drone using the CoDrone.

1. Attach the attachment frame to the CoDrone. It slips over the center of the body, with the loop underneath the drone. Be careful not to twist it too much that it doesn't break.

CoDrone with attachment frame

2. Slide the plastic band through the loop such that it does not slip back out too easily. It should slip out when the forces are too large – this will avoid a crashing drone when the hook gets entangled to somethings else than the astronaut.

3. Slide the other end of the plastic band through the loop at the fishhook. Ensure the band is secure.

4. A drop of plastic glue may secure the band better but test out your equipment first! Every excess weight reduces performance of the drone – and you want to be sure that your loops are not too large, and the total length of the band is not too long. However, too short is not good either because the down draft from the propellers hitting the payload will reduce performance as well. Find the best compromise!

5. Test fly the drone first just with the bare hook attached.

6. Test fly the drone with an astronaut hooked on to it.

7. Look for hover stability.

8. Look for how the drone reacts to pitch, yaw, and roll.

How to attach the fishhook

9. Is the band long enough to allow safe landing?

10. Can you land and then successfully take off after with the fishhook attached?

11. Lengthen or shorten the band if needed. Measure how the long the distance is from the loop at the frame to the loop at the fishhook:

____________ cm

Flight Practice 5: Rescue Mission

Rescue Mission to an Asteroid (Practice for Competition Event #7)

Now we are prepared for the rescue mission.

1. Place the astronaut on the asteroid at your station in the gym. Take care that the capture loop sticks out well enough to be picked with the fishhook, but also that the astronaut is not stuck too tight to the asteroid.
2. Wait for ATC authorization. Follow communication protocol!
3. Turn on your CoDrone.
4. Place the drone in the center of the Earth.
5. Turn on the remote-control.
6. Sync.
7. Choose a slow Speed Mode first.
8. Take off.
9. Fly to the asteroid.
10. Try picking up the astronaut and bring him home.
11. Land.
12. Turn off drone and remote-control.

Did you crash?

Did you land at the center of the Earth?

… with the astronaut?

It wasn't easy even if you succeeded perfectly. Try again!

What makes it difficult?

Is this a problem?

Discuss with your Crew partner!

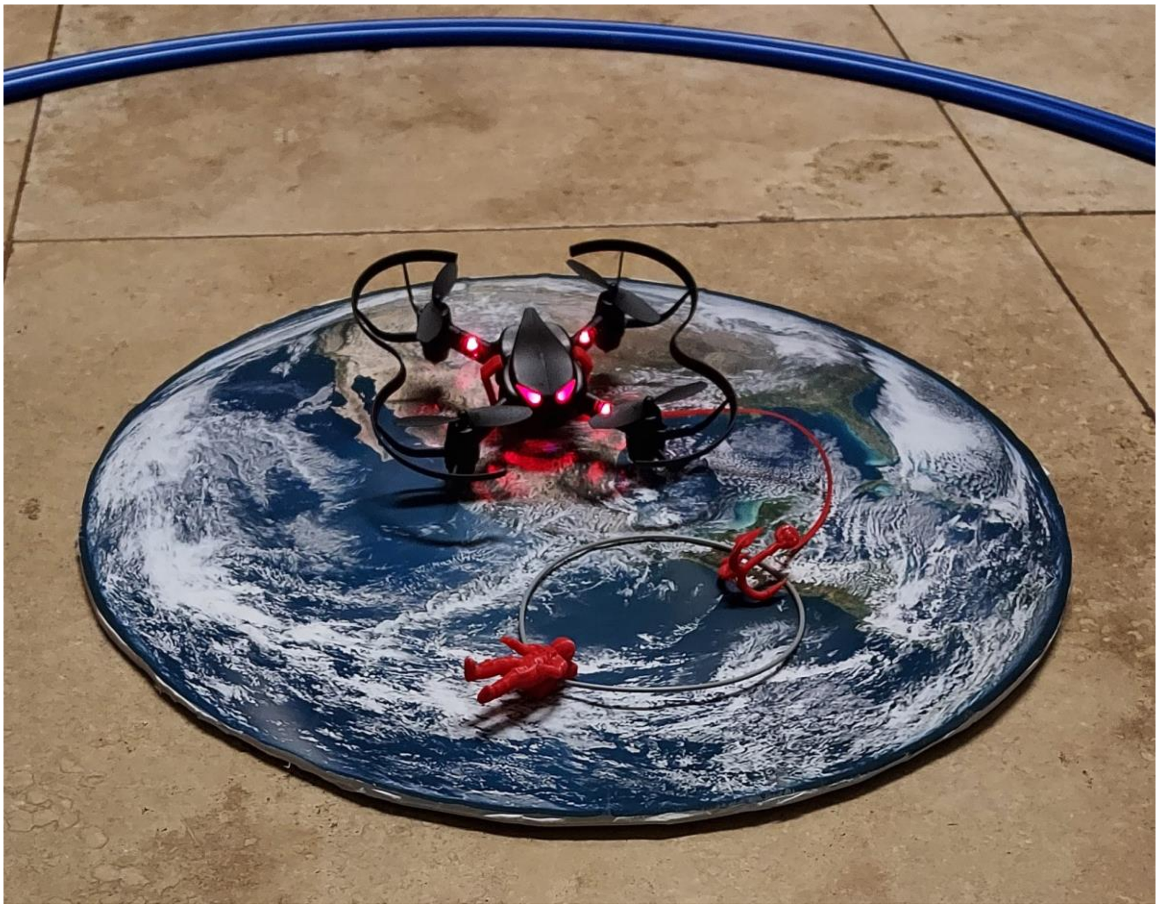
The goal of the rescue mission is to get to the asteroid, pick up the astronaut, and land him safely on the Earth.

Time matters! You need to get ready with

- your drone,
- a battery for it, and
- the remote-control.

Then you wait until ATC instructs you to power up and sync. The timer starts during synchronization.

The you are on your own, getting the mission accomplished. The timer stops when the engines of the drone are off or … when you crash.



Module 8: Competition Practice

The highlight of the AIR Challenge is getting to showcase all that you have learned and practiced through out the week in a friendly competition event. This event will be attended by family and friends. You will be assigned one of four stations to compete at – meaning there will be a rotation of eight students in each station for each event for all but the Star Wars Battle Competition.

Each student will be given one practice run at the start of each event. The students at each station will first complete a practice run and after all eight have completed the practice they will then go through the official run for the competition. Facilitators at each station will be the judges.

Here are the competition events:

Earth to Mars: (CoDrone / Autonomous)

To Mars and Back: (CoDrone/ Autonomous)

Asteroid Mission: (CoDrone / Autonomous)

Orbit the Asteroid and Return to Earth: (CoDrone / Autonomous)

Orbit the Asteroid and Return to Earth (CoDrone / Remote-Controlled)

Rescue Mission (CoDrone / Remote-Controlled)

Star Wars Battle (Star Wars Fighters / Remote-Controlled)

Refer to the descriptions in the Flight Practices to know what is expected of each event.

Competition Practice Notes:

Use these blank pages to write down notes for competition. Refer to the Flight Practices for what is expected in in event.

Your Drone: PETRONE ________________

Your Name: ____________________________________

Your Group, Wing, Squadron and Crew:

____

Resources:

Page 12:

- Image 1: Bee - This Photo by Unknown Author is licensed under CC BY-NC
- Image 2: Drone - https://get.pxhere.com/photo/drone-camera-isolated-background-helicopter-technology-remote-surveillance-aerial-white-sky-control-fly-propeller-digital-drones-aircraft-copter-photo-robot-modern-military-flight-aviation-force-army-transport-vehicle-airplane-plane-airshow-war-video-blue-helicopter-rotor-rotorcraft-tiltrotor-bell-boeing-quad-tiltrotor-radio-controlled-helicopter-1446059.jpg

Page 13:

- Image 3: Drone photography - https://pixabay.com/photos/bride-and-groom-wedding-celebration-2718863/
- Image 4: Drone (real estate)- This Photo by Unknown Author is licensed under CC BY-SA-NC
- Image 5: Drone (utilities)- This Photo by Unknown Author is licensed under CC BY-SA-NC
- Image 6: Drone (agriculture)- This Photo by Unknown Author is licensed under CC BY-NC
- Image 7: Drone (police)- This Photo by Unknown Author is licensed under CC BY-SA-NC

Page 14:

- Image 8 & 9: CoDrone- https://www.robolink.com/products/codrone-pro
- Image 10 & 11: https://www.amazon.com/stores/Propel+Toys/page/58F5408E-5317-49D4-8C24-8812CD021ADB?ref_=ast_bln

Page 15:

- Image 12: FAA - This Photo by Unknown Author is licensed under CC BY
- Image 13: Flight navigation: Image courtesy of Federal Aviation Administration (FAA)

Page 16:

- Wikipedia contributors. “Compass Rose.” *Wikipedia*, 13 Jan. 2023, en.wikipedia.org/wiki/Compass_rose.
- MooMooMath and Science. “What Are Cardinal Directions and a Compass Rose?” *YouTube*, 13 Nov. 2020, www.youtube.com/watch?v=lkfq37bMVBY.

Page 22:

- “Principles of Flight.” *Science Learning Hub*, www.sciencelearn.org.nz/resources/299-principles-of-flight.
- Pilot Institute. “Principles of Flight – the 4 Flight Forces Simply Explained.” *Pilot Institute*, 1 June 2022, pilotinstitute.com/principles-of-flight.
- “How to Fly a Drone: A Beginner’s Guide to Multirotor Systems.” *UAV Coach*, 29 Nov. 2022, uavcoach.com/how-to-fly-a-quadcopter-guide.

Page 35:

- Piper's Club Owner's Manual: https://stpeteair.org/wp-content/uploads/cub_owners_manual.pdf

Page 44:

- *Maintenance – Robolink Basecamp*. (n.d.). https://learn.robolink.com/lesson/codrone-arduino-maintenance/

Page 55:

- Compass Rose: https://pngimg.com/uploads/compass/compass_PNG25581.png

Page 88:

- *LEDs – Robolink Basecamp*. (n.d.). https://learn.robolink.com/lesson/codrone-blockly-sr-leds/

Page 95:

- *1C: Variables – Robolink Basecamp*. (n.d.). https://learn.robolink.com/lesson/variables/
- *Loops – Robolink Basecamp*. (n.d.). https://learn.robolink.com/lesson/codrone-blockly-sr-loops/

Page 104:

- Robolink Inc. (2018, August 1). *Building the CoDrone Pro Remote Time Lapse* [Video]. YouTube. https://www.youtube.com/watch?v=mDDdWtQXyF4
- Inc, R. (2018, July 31). *Building the CoDrone Pro Remote* [Video]. YouTube. https://www.youtube.com/watch?v=HAXq1gBO3I8&feature=youtu.be
- *Building the Remote – Robolink Basecamp*. (n.d.). [Video]. https://learn.robolink.com/lesson/codrone-arduino-build-the-remote/

Page 105:

- Inc, R. (2018a, February 13). *Flying CoDrone* [Video]. YouTube. https://www.youtube.com/watch?v=Ujkp8QSI5J4&feature=youtu.be
- *Using the Remote – Robolink Basecamp*. (n.d.). https://learn.robolink.com/lesson/codrone-arduino-using-the-remote/

Page 125:

- *UAV / Drone Jobs | Build a Career in the Drone Industry*. (2022, October 27). UAV Coach. https://uavcoach.com/uav-jobs/
- French, S. (2020, July 21). *7 unexpected drone career paths if you want to work with UAVs*. The Drone Girl. https://www.thedronegirl.com/2020/08/10/drone-career-paths/
- Posea, P. (2022, February 23). *Is Drone Pilot A Good Career? (7 High Demand Industries In 2022)*. Dronesgator - Drone Review, Buyer Guides and Best Drones Lists. https://dronesgator.com/drone-pilot-career-demand/

Page 136:

- Schwartz, D. (2021, November 10). *Why Drones Are the Future of Outdoor Search and Rescue*. Outside Online. https://www.outsideonline.com/outdoor-adventure/exploration-survival/drones-search-rescue/
- Hayes, T. (2022, November 8). *DRONES FOR SAR (SEARCH AND RESCUE)*. https://www.dronefly.com/blogs/news/drones-for-sar-search-and-rescue/
- Dukowitz, Z. (2021, July 14). *Drones in Search and Rescue: 5 Stories of Drones Helping Save Lives*. UAV Coach. https://uavcoach.com/search-and-rescue-drones/
- Image Drone rescue- This Photo by Unknown Author is licensed under CC BY-NC-ND