



# CoDrone Mini

---

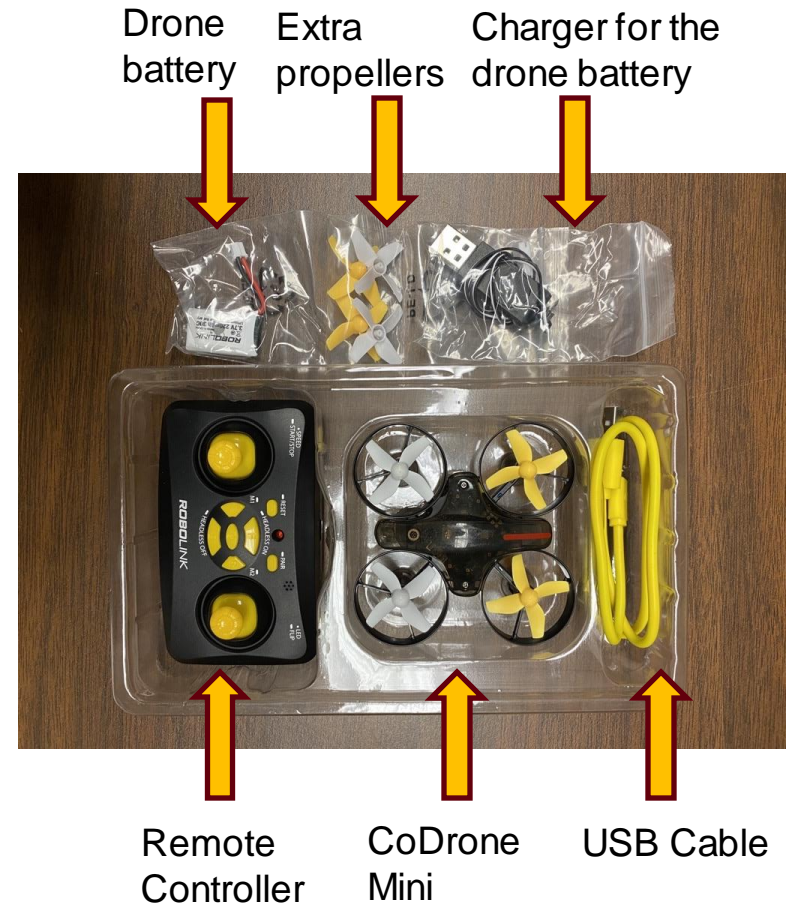
CODERS CS Team

Summer 2023

# Introducing CoDrone Mini

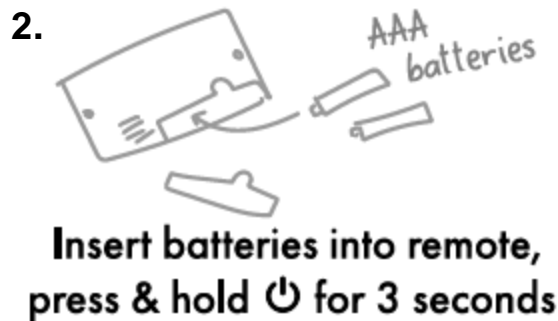
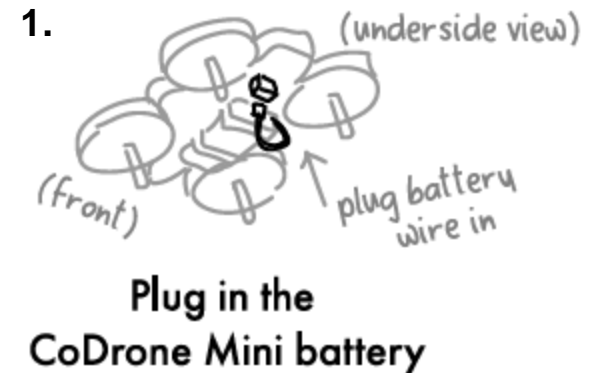
<https://learn.robotlink.com/product/codrone-mini/>

- Can be programmed with block coding in Blockly or with Python
- Comes with a remote controller. Uses radio frequency to connect the remote and the drone
- We can code to make the drone to fly around, flash colors, do flips, etc.
- **Note:** The **red** line on the drone indicates the front



# Powering the drone and the remote

1. Plug in the drone battery wire underneath the drone.
2. Insert two AAA batteries into the controller (batteries are not included in the packaging) **OR** connect the controller with a computer using a micro-USB cable.



**OR**



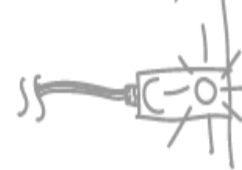
# Charging CoDrone Mini

- With the provided batteries, CoDrone Mini's flight time is approximately **5 minutes**.
- The drone's battery can be charged by the provided charger.
- We can connect the provided charger to a USB port of a computer.
- The charger status can be recognized by the lights on it - light on means it is still charging and light off means charging is complete.
- Charging time is approximately 45 minutes.



## Low battery

Drone LED blinks a single color when the battery is low



## Charging

Light on means still charging



## Charge battery

Charge battery by plugging it into the USB battery charger

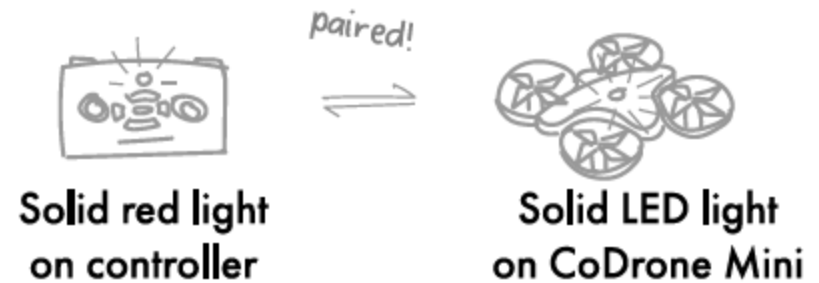


## Charged

Light off means fully charged

# Pairing the controller and the drone

- The drone and controller are automatically paired when both are powered ON.
- A blinking light on the drone will turn into a solid light.
- If not already paired, then unplug the battery, reconnect the battery, and shake the drone gently. The light will turn into a flashing blue from a flashing red.
- Press and hold the “pair” button on the controller.





# Controlling the drone using the remote

## Left shoulder

**SPEED:** Press the button to cycle between 3 speeds. You can tell which speed you're on based on how many beeps you hear.

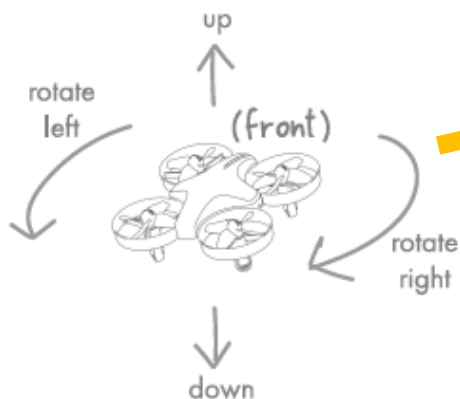
**START/STOP:** Press and hold for take off and landing

## Right shoulder

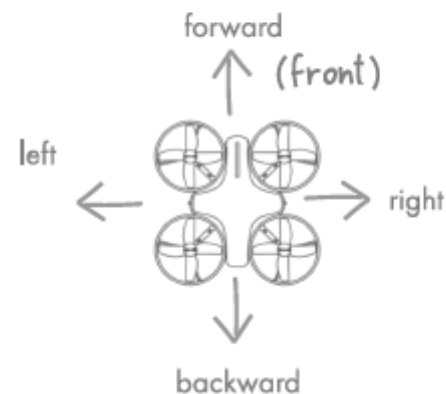
**LED:** Press to change the drone's LED color

**FLIP:** During flight, press and hold until you hear a beep. Then use the right joystick to flip in any 4 directions.

## Left joystick



## Right joystick

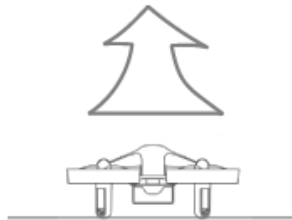
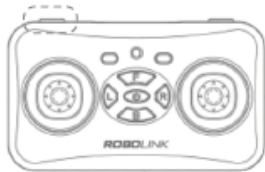


Power button for the remote

# Controlling using remote – Contd.

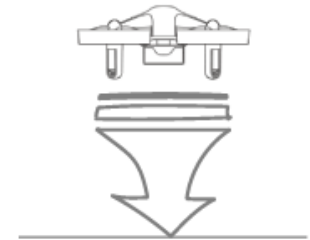
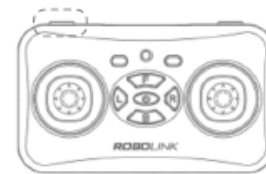
## How to take off

1. Place the CoDrone Mini on a flat surface.
2. Press and hold the left shoulder button on the controller. CoDrone will take off and then hover!



## How to land

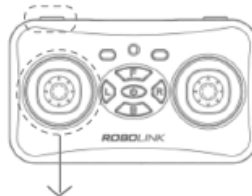
Press the left shoulder button on the controller during flight for more than 3 seconds to do a soft landing.



## Emergency stop

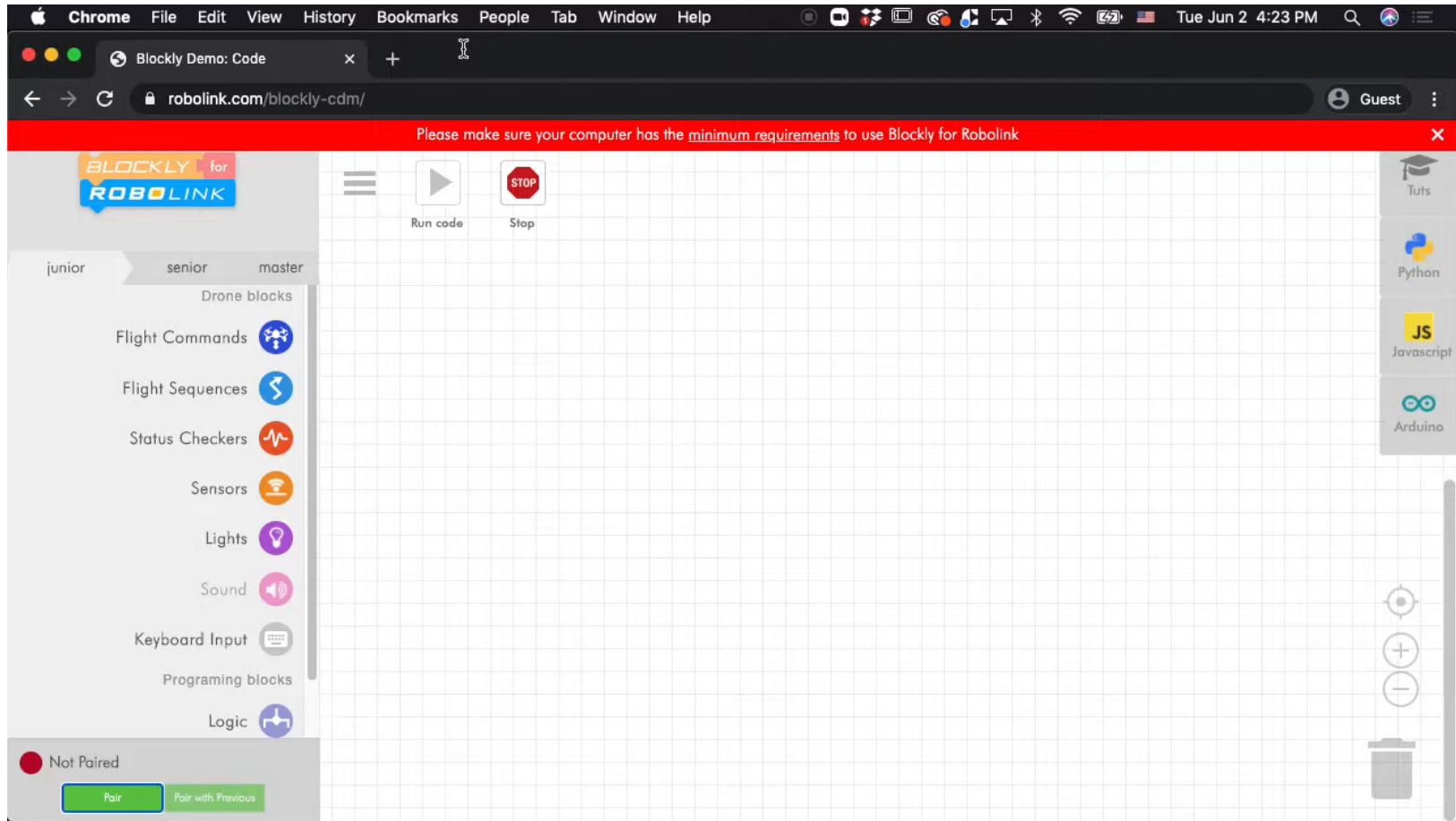
In the event of an emergency during the flight, push the left joystick down while holding down the left shoulder button. This will immediately stop the motors in the air. Use this if you need the CoDrone Mini to stop suddenly.

**NOTE:** The controls below are in the default Mode 2. If you're flyign in Mode 1, you'll push the RIGHT joystick down instead while holding down the left shoulder button.



# Coding the drone using Blockly

<https://youtu.be/kC8vzVzlhE8>





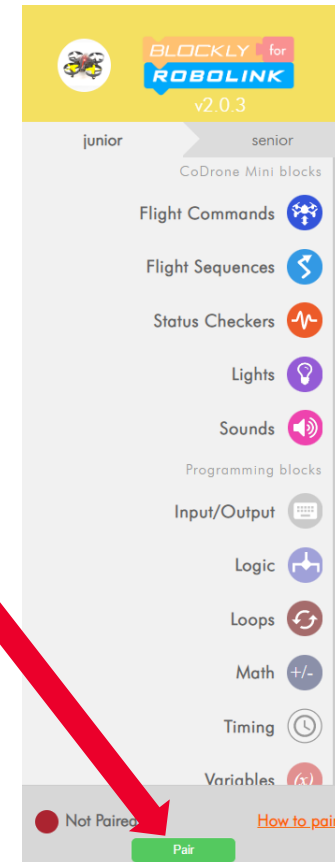
# BLOCKLY Pairing Steps

1. Open the Chrome browser
2. Search for <chrome://flags>
3. Enable “Experimental Web Platform features”
4. **Relaunch** the Chrome browser
5. Search for <http://www.robolink.com/blockly-cdm>
6. Click on “Pair”
7. Select the remote

## Experimental Web Platform features

Enables experimental Web Platform features that are in development. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros

[#enable-experimental-web-platform-features](#)



# Flight Commands

Junior	Senior

Click "Run Code" to execute the commands

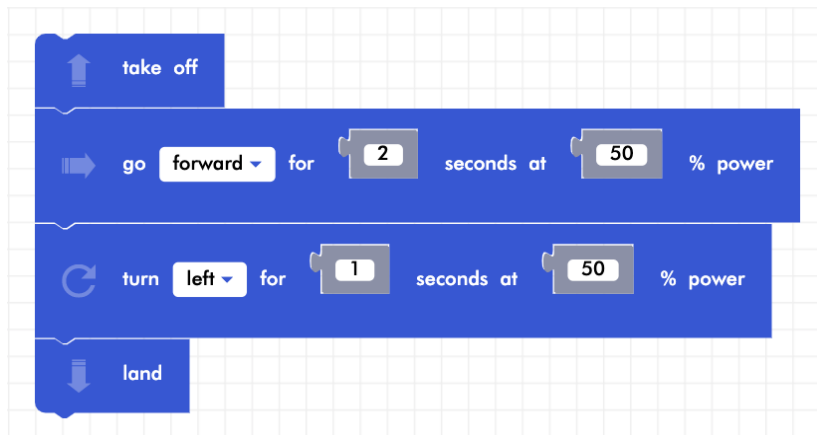
**Note:** Complete the pairing steps from slides 5 & 9 first.

# Flight Directions



# Flight Directions in BLOCKLY

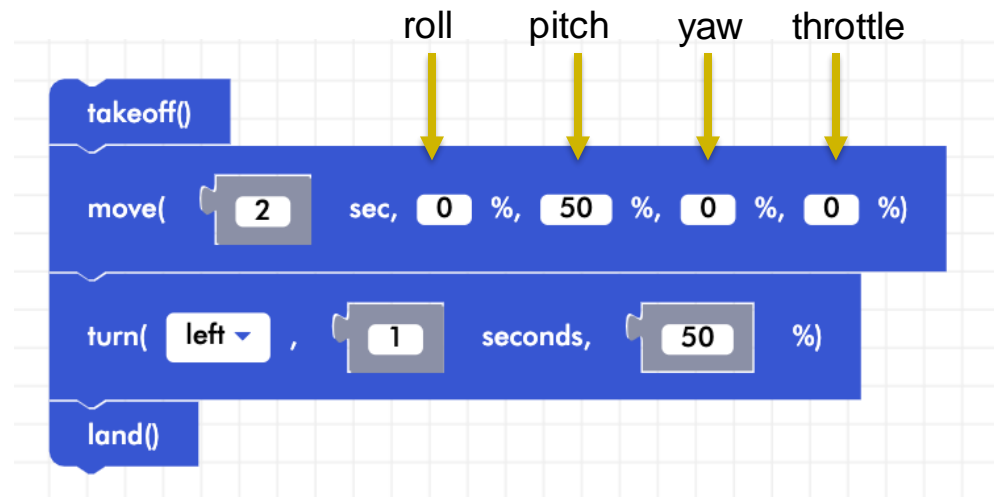
Junior



```
take off
go forward for 2 seconds at 50 % power
turn left for 1 seconds at 50 % power
land
```

The Junior level code consists of four blocks: a 'take off' block, a 'go forward' block with a duration of 2 seconds and 50% power, a 'turn left' block with a duration of 1 second and 50% power, and a 'land' block.

Senior

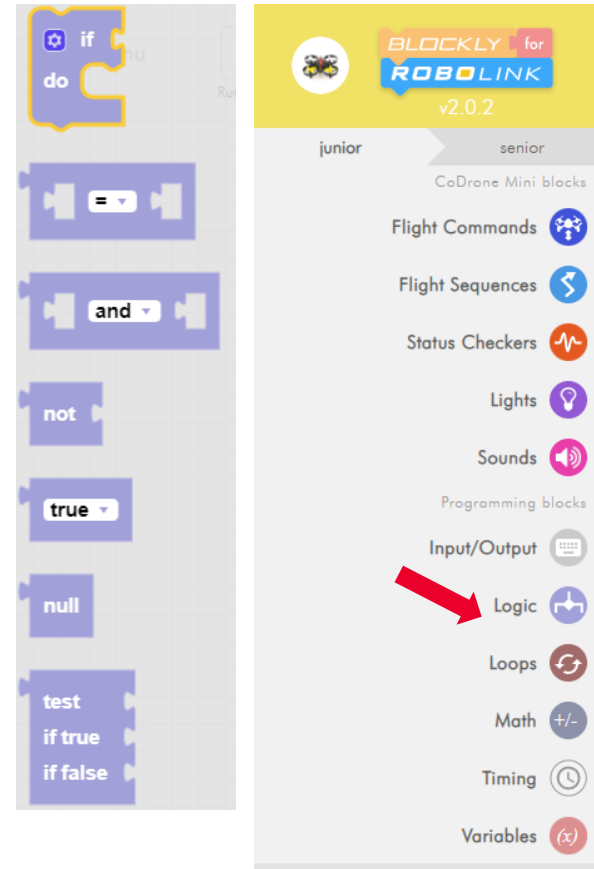
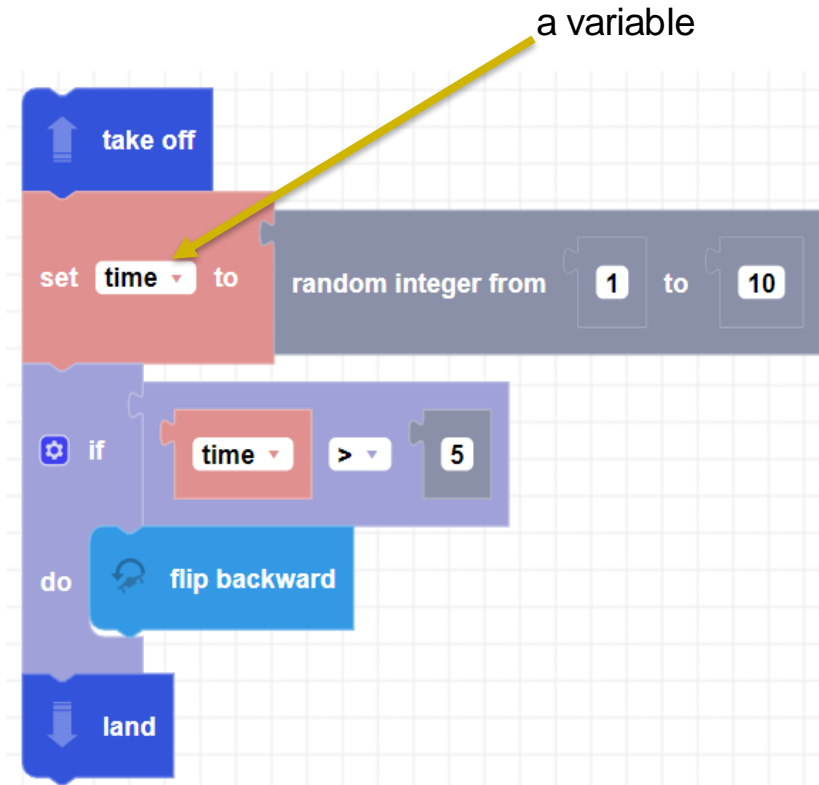


```
takeoff()
move( 2 sec, 0 %, 50 %, 0 %, 0 %)
turn( left , 1 seconds, 50 %)
land()
```

The Senior level code uses more detailed blocks: 'takeoff()', 'move()' with parameters for seconds (2), roll (0%), pitch (50%), yaw (0%), and throttle (0%), 'turn()' with parameters for direction (left), seconds (1), and power (50%), and 'land()'. Yellow arrows point from the labels 'roll', 'pitch', 'yaw', and 'throttle' to the corresponding percentage values in the 'move()' block.

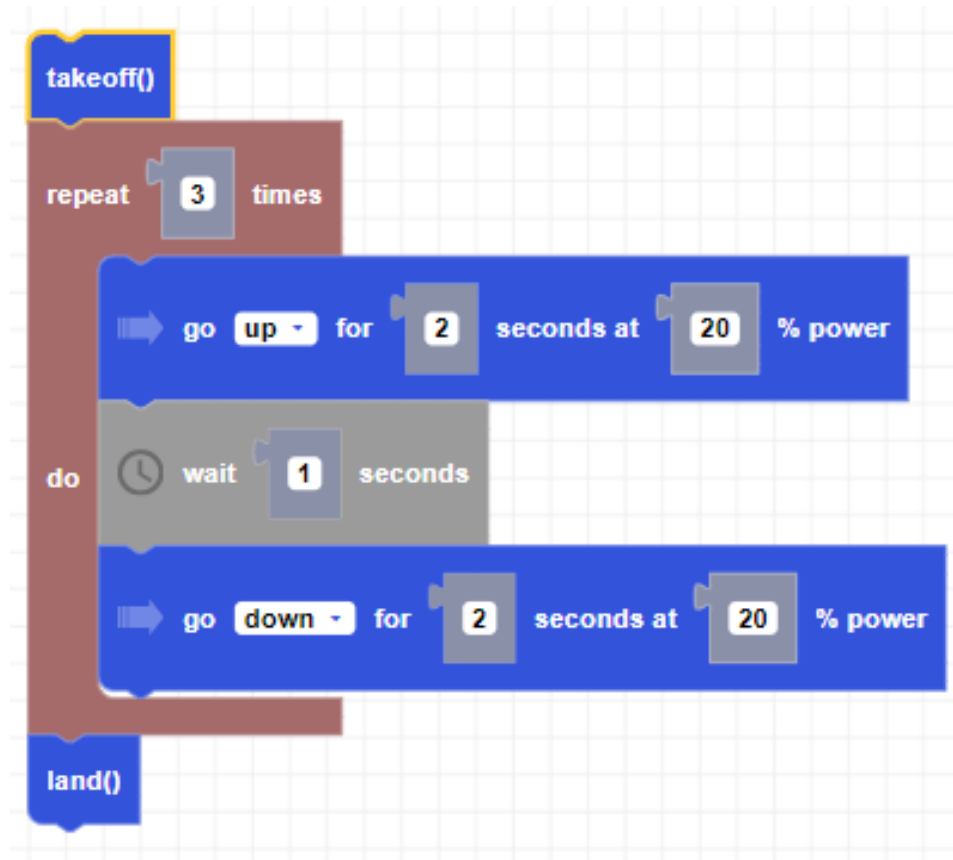
# Activity-1: Logic

**Goal:** Flip the drone when a certain condition is met.



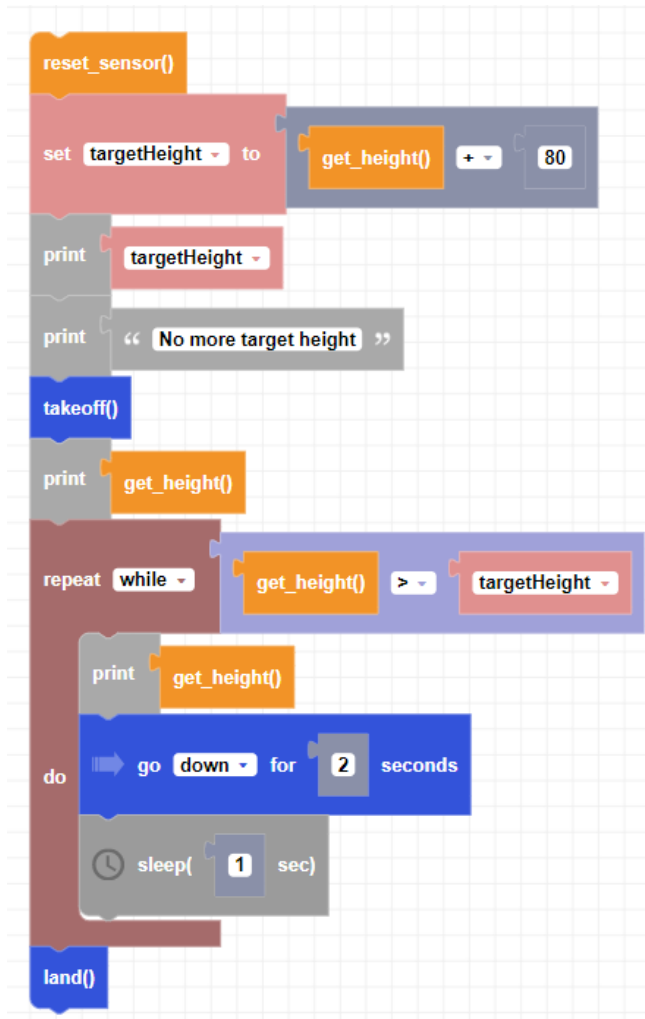
# Activity-2: Loop

**Goal:** Fly the drone up and down for 3 times.



# Activity-1: Autopilot

**Goal:** Fly the drone by maintaining a certain height



```
reset_sensor()
set targetHeight to get_height() + 80
print targetHeight
print " No more target height "
takeoff()
print get_height()
repeat while get_height() > targetHeight
  print get_height()
  do
    go down for 2 seconds
    sleep( 1 sec)
land()
```

The code is written in a Scratch-style block-based language. It starts with a 'reset\_sensor()' block. Then, a 'set targetHeight to' block is connected to a 'get\_height()' block, followed by a '+' sign and a '80' block. Below this is a 'print targetHeight' block, followed by a 'print' block with the text " No more target height ". The next block is 'takeoff()', followed by 'print get\_height()'. A 'repeat while' loop is then used, with the condition 'get\_height() > targetHeight'. Inside the loop, there is a 'print get\_height()' block, a 'do' block containing 'go down for 2 seconds' and 'sleep( 1 sec)', and finally a 'land()' block at the end.



# Thank You

---

Any Questions?