

Lesson 3:

Technical parts

Sensing and connectivity

Drones4STEM © 2023 by Drones4STEM is licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>



Co-funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

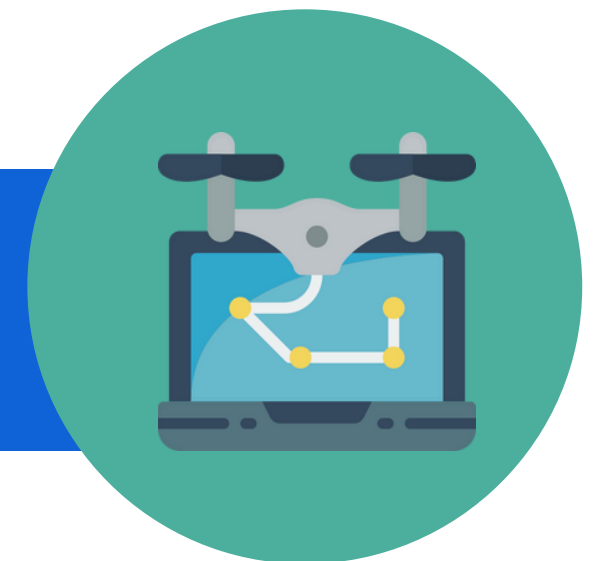
Lesson Objectives

Understand the fundamentals of programming drones to collect environmental data during autonomous flights.

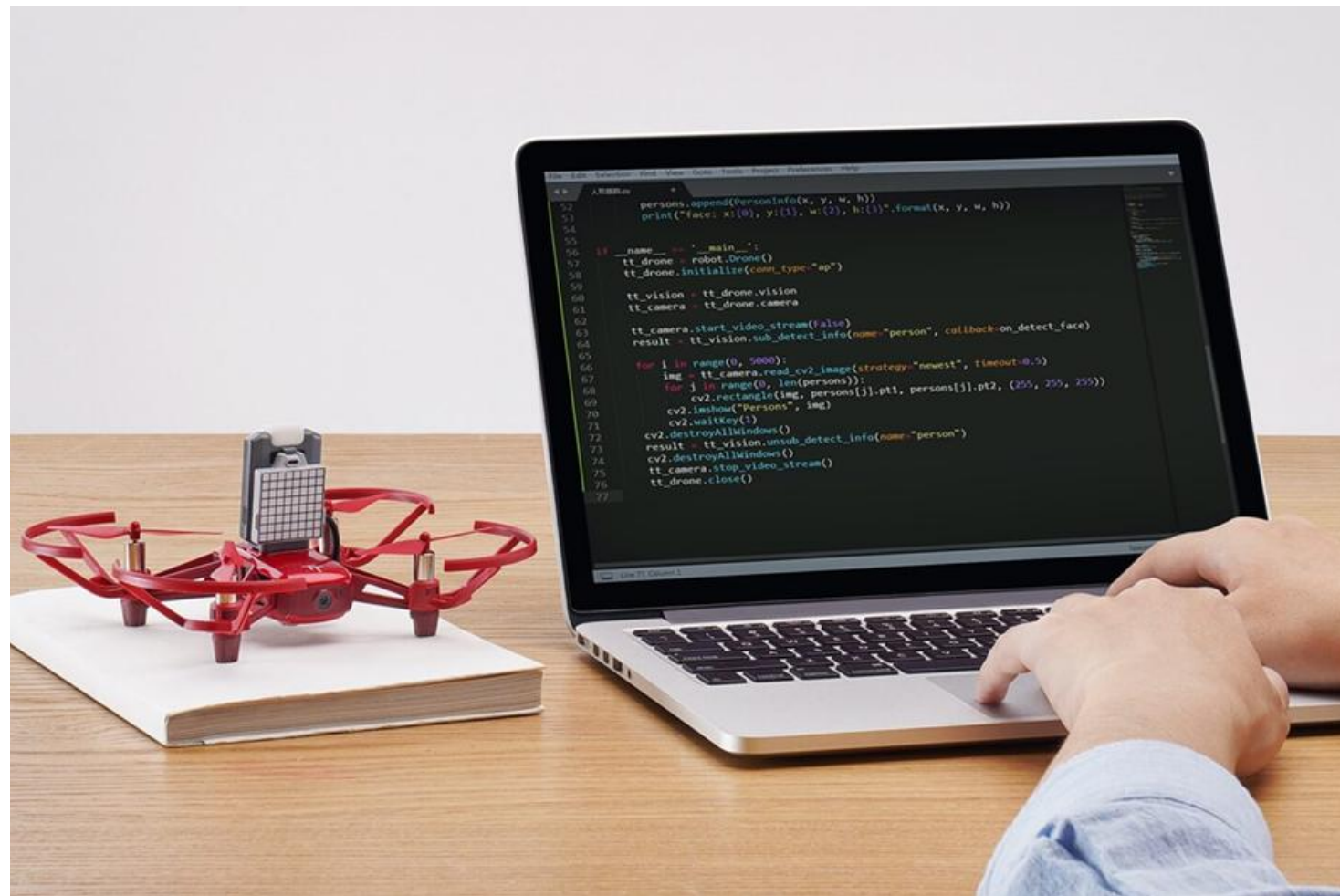


Ensure accurate and synchronized
data collection

Learn about swarm mode flights.



Materials Needed



HARDWARE

DJI Robomaster TT Drone

Ensure it's charged, and sensors functional.

Drones4STEM FloorMat

PC

SOFTWARE

MIND+ Scratch GUI Platform

Test connectivity between drones and Scratch

Basics for Flight Planning

0

Before You Start...

- Define Mission Objectives: Environmental parameters to be measured and select the sensors.
- Prepare Equipment. Ensure all drones are charged, and sensors are functional. Test connectivity between

1

Planning and

What Are Waypoints?

Predefined coordinates (x, y, z) the drone must follow during the mission.



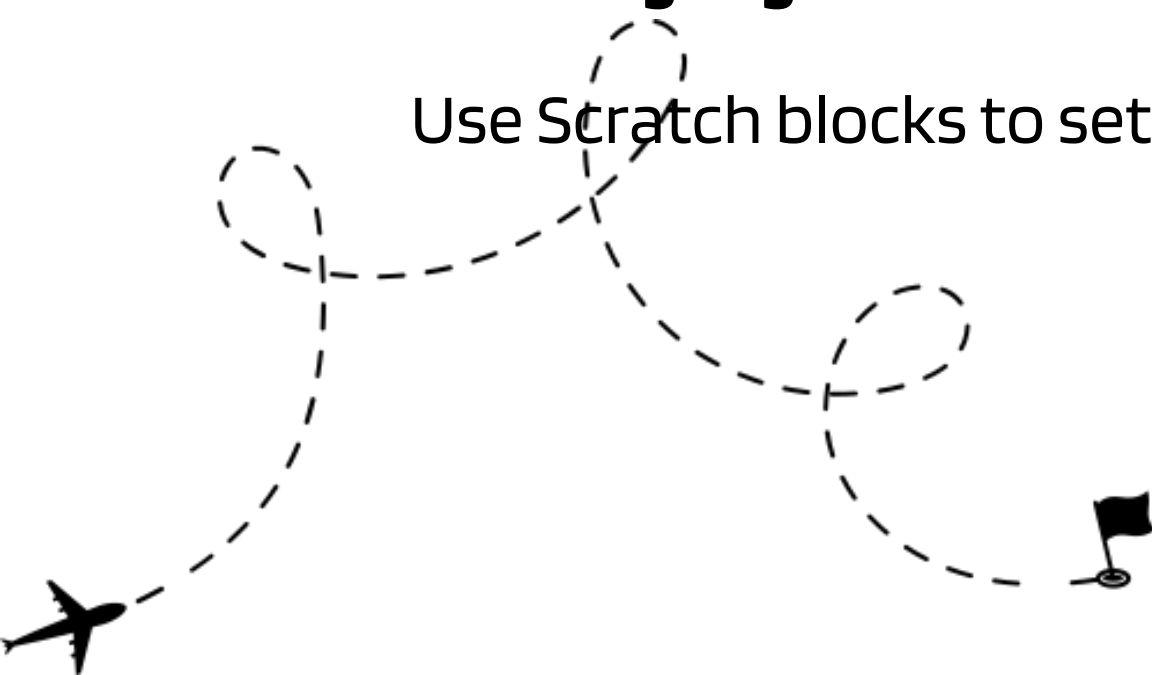
Planning Flight Path:

Use Scratch blocks to set waypoints and define the altitude and horizontal position of each

Example:

- `"Fly to x: 50 y: 30 z: 10"`
- `"Fly to x: 100 y: 50 z: 20"`

Add "wait" block between waypoints to control the flight duration



Using Measurement Sensors

Internal sensors for Tello Robomaster TT drone:

Infrared Time-of-Flight (ToF) Sensor: Facilitates obstacle detection and avoidance, as well as environmental awareness.

Barometer: Detects air pressure changes to assist with altitude estimation and stable hovering.

Vision Positioning System (VPS): Utilizes a downward-facing camera and infrared sensors to enable precise indoor hovering and positioning.

5 MP HD Camera: Captures high-resolution images and streams 720p video.

LED Indicator and 8×8 Dot-Matrix Display: Programmable visual outputs for signaling and interactive applications.

ESP32-Based Extension Module: Provides interfaces (I2C, SPI, UART, GPIO) for connecting additional sensors. Supports programming in Python, Arduino, and MicroPython.

Using Measurement Sensors

External sensors: Integrating and Calibrating Sensors on Drones

- **Type of Sensors:**

- Temperature sensors: Measure ambient temperature.
- Humidity sensors: Measure moisture in the air.
- Barometric pressure sensors: Measure atmospheric pressure.
- Air quality sensors: Measure pollutants and particulate matter.

- **Choosing the Right Sensors:**

- For temperature: Use a sensor with a broad temperature range.
- For humidity: Choose a sensor with good accuracy in varied conditions.

Using Measurement Sensors

Integrating Sensors on Drones

- **Attach and connect sensors to the drone's main system:**
 - Sensors need to be physically mounted on the drone. Mounting brackets or custom enclosures are often used to secure the sensors, ensuring they are stable during flight. The sensors are then connected to the drone's onboard system, typically via ports or interfaces such as I2C, UART, or SPI, depending on the sensor type. Proper integration ensures that sensors can communicate seamlessly with the drone's microcontroller or flight controller.

Using Measurement Sensors

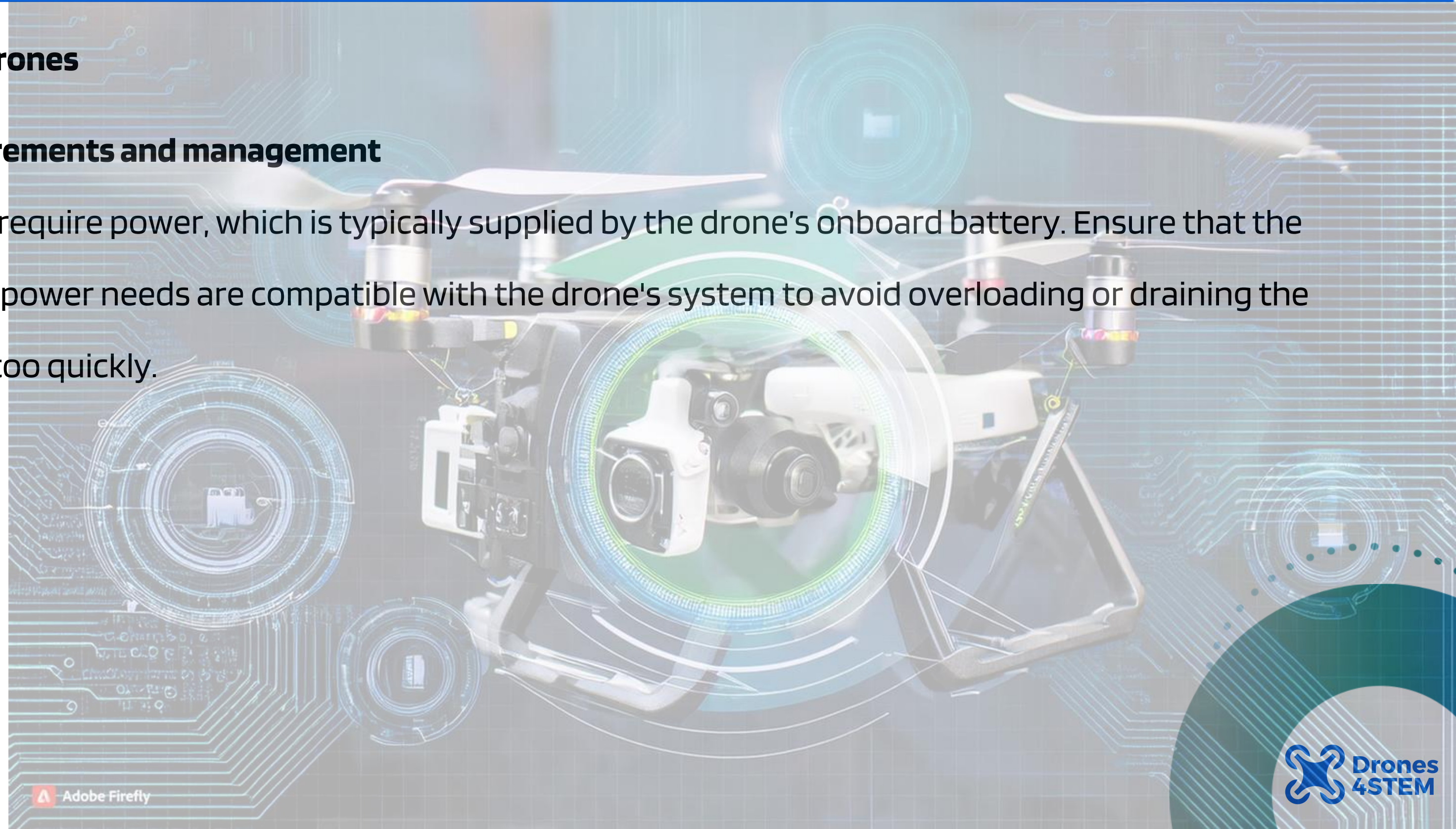
Integrating Sensors on Drones

- **Configure sensor communication protocols**
 - After physical integration, configure the sensors to communicate with the drone's system. This may involve setting up protocols such as I2C or serial communication, ensuring that the data is read accurately. The drone's software must be programmed to recognize and interpret the incoming data from the sensors.

Using Measurement Sensors

Integrating Sensors on Drones

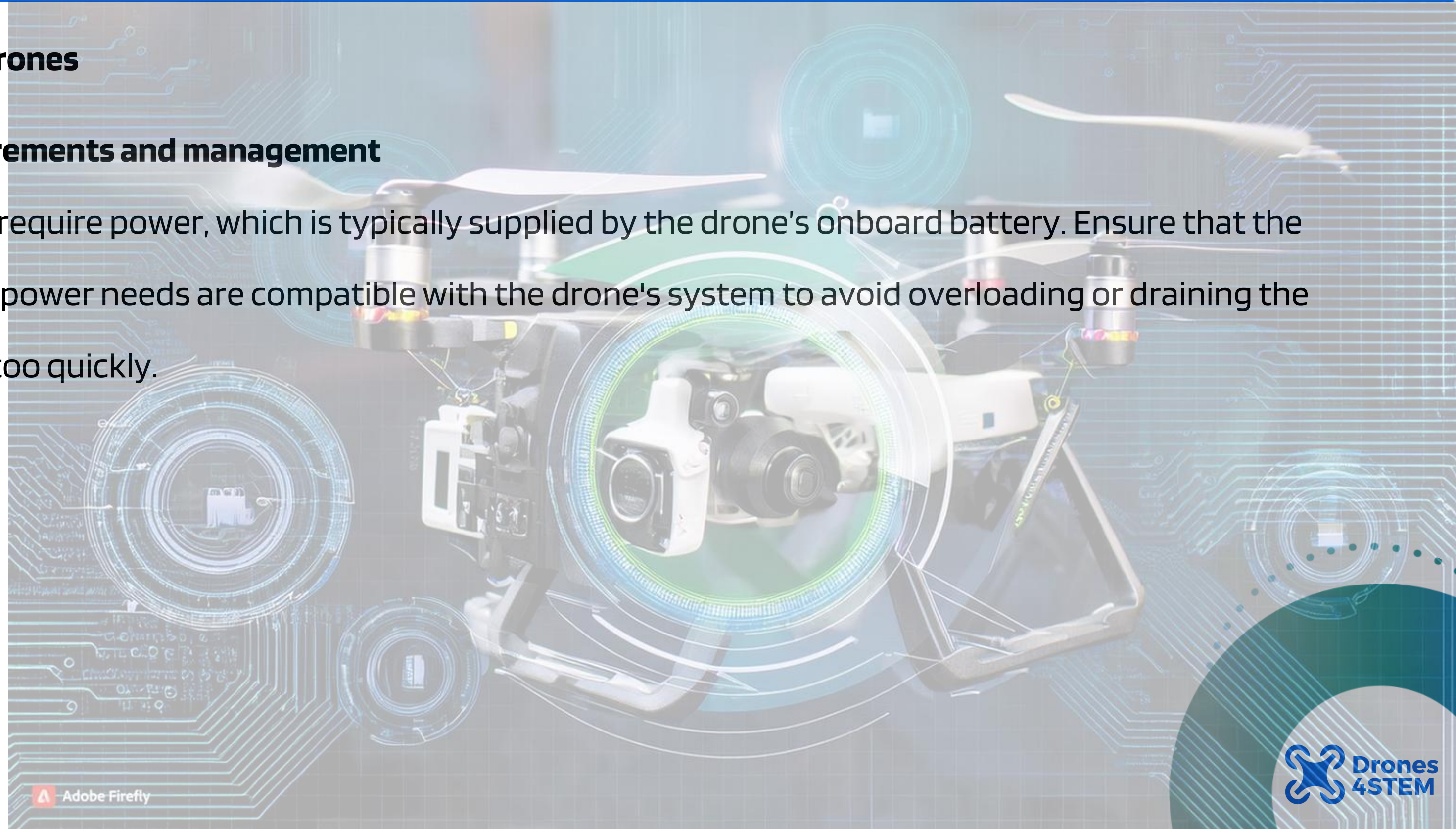
- **Power requirements and management**
 - Sensors require power, which is typically supplied by the drone's onboard battery. Ensure that the sensors' power needs are compatible with the drone's system to avoid overloading or draining the battery too quickly.



Using Measurement Sensors

Integrating Sensors on Drones

- **Power requirements and management**
 - Sensors require power, which is typically supplied by the drone's onboard battery. Ensure that the sensors' power needs are compatible with the drone's system to avoid overloading or draining the battery too quickly.



Using Measurement Sensors

Calibrating Sensors on Drones

- **Why calibration is important**

- Calibration ensures that the sensors provide accurate and reliable readings. Environmental factors, such as temperature and humidity, can affect sensor accuracy, so calibration is critical for consistent performance.

- **Calibration steps:**

- **Reference Measurement:** Compare the sensor's readings against a known standard or reference value, such as a calibrated thermometer for temperature sensors.
- **Adjustment:** Adjust the sensor's output using software or hardware tools until its readings match the reference values.
- **Recalibration Schedule:** Sensors should be recalibrated periodically, especially after extended use or

Sensor Data Collection



Environmental monitoring relies heavily on the collection of accurate and reliable data. By integrating sensors into drones and programming them through Scratch, you can measure key environmental factors like temperature, humidity, and air quality in real time.

Learn in the next slides how to add sensor blocks in Scratch and use them effectively to collect essential environmental data

Sensor Data Collection

Adding Sensor Data Blocks

Scratch extensions like "Micro:bit", "TT" or "Arduino" allow you to integrate physical sensors

Sensor blocks can collect real-time data such as:

- **Temperature:** Readings from digital temperature sensors (e.g., **DHT11** or DS18B20).

Block Script:

- Connect the temperature sensor to the microcontroller (e.g., Arduino).
- In Scratch, use a block like set [temperature] to (sensor reading) to fetch data.
- Display the temperature in real-time using the say [temperature] block or a dynamic sprite.

Sensor Data Collection

Adding Sensor Data Blocks

Temperature: Readings from digital temperature sensors (e.g., **DHT11** or DS18B20).

Temperature measurement example: *Display the current temperature on the screen*

```
when green flag clicked
forever
  set [temperature] to (sensor reading from DHT11)
  say (join [Temperature: ] (join (temperature) [°C]))
  wait (1) seconds
end
```

The temperature sensor sends its reading to the variable temperature.

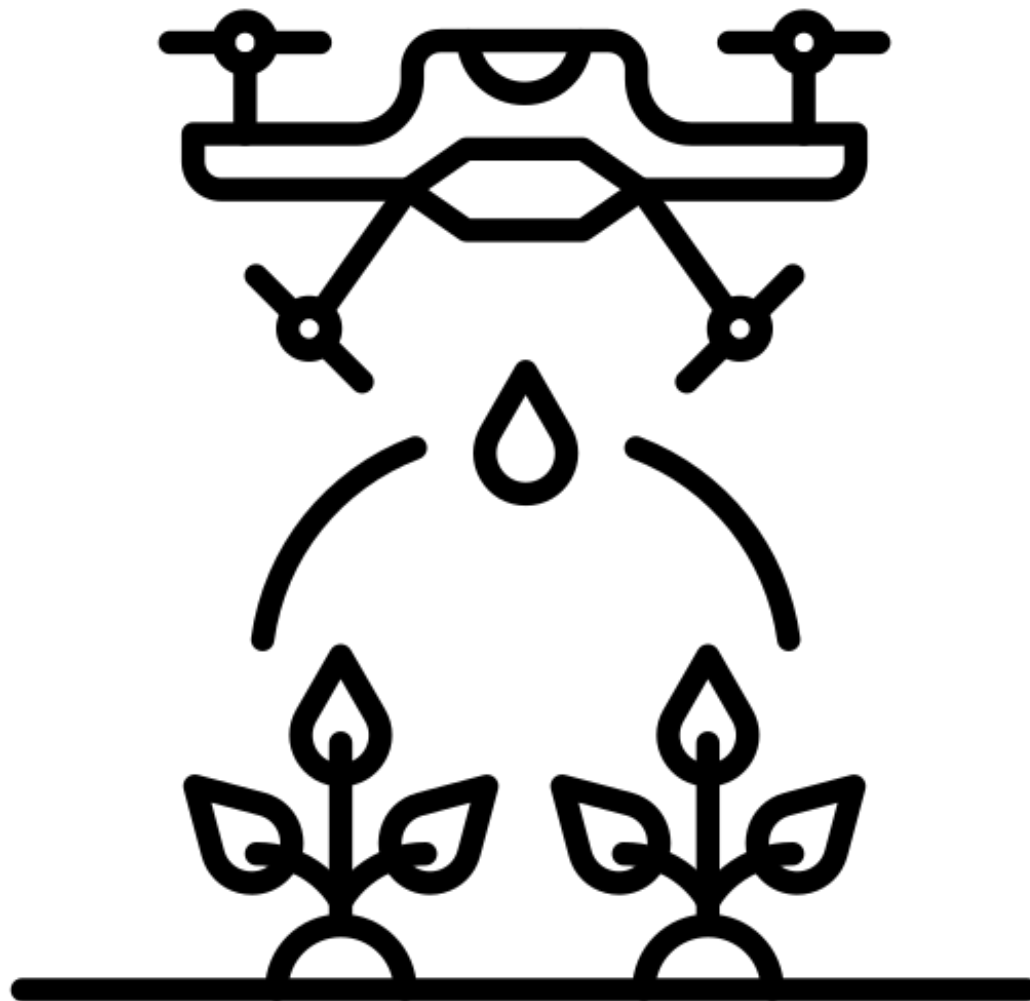
The say block displays the temperature in real time.

The loop updates the reading every second.

Sensor Data Collection

Adding Sensor Data Blocks

- **Humidity:** Data from humidity sensors (e.g., **DHT11** or AM2302).



Block Script:

- Connect the humidity sensor and ensure data transmission.
- Use if-then conditions to trigger actions based on humidity levels, like alerting users when humidity exceeds a threshold

Sensor Data Collection

Adding Sensor Data Blocks

- **Humidity:** Data from humidity sensors (e.g., DHT11 or AM2302).

***Example:** Alert if humidity exceeds a specific threshold.*

```
when green flag clicked
forever
  set [humidity] to (sensor reading from DHT11)
  if <(humidity) > [60]> then
    say [High Humidity Detected!]
  else
    say (join [Humidity: ] (join (humidity) [%]))
  end
  wait (1) seconds
end
```

The sensor updates the humidity variable in real time.

If humidity exceeds 60%, an alert is displayed.

If not, the current humidity is shown as a percentage.

Sensor Data Collection

Adding Sensor Data Blocks

- **Air Quality:** Use Air Quality Sensors (e.g., MQ135) to Measure Pollutants

Example: *Change the screen color based on air quality levels.*

```
when green flag clicked
forever
  set [air_quality] to (sensor reading from MQ135)
  if <(air_quality) < [50]> then
    set [background color] to [green]
    say [Good Air Quality]
  else
    if <(air_quality) < [100]> then
      set [background color] to [yellow]
      say [Moderate Air Quality]
    else
      set [background color] to [red]
      say [Poor Air Quality]
    end
  end
  end
  wait (1) seconds
end
```

The air_quality variable gets updated by the MQ135 sensor.

Depending on the air quality value, the background color changes:

- *Green: Good air quality.*
- *Yellow: Moderate air quality.*
- *Red: Poor air quality.*

A message indicating the air quality is displayed.

.

Sensor Data Collection

Connectivity

- **How to connect sensors and drones?**

Example: Connecting temperature sensor DH11 to Tello Robomaster TT drone.

What do we need?

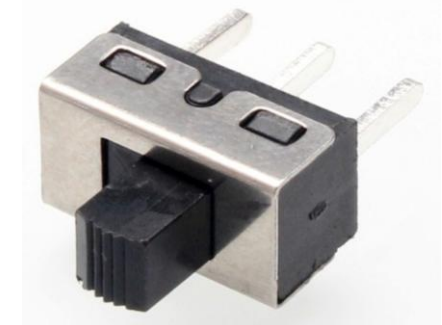
- DH11 sensor.



- MicroSD CARD + ADAPTER



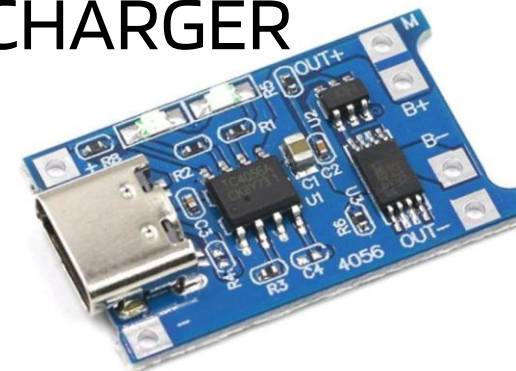
- Slide Switch Mini SPDT



- ESP32



- BATTERY + CHARGER



Sensor Data Collection

Connectivity

- **How to connect sensors and drones?**

Example: Connecting temperature sensor DH11 to Tello Robomaster TT drone.

How to connect it:

DH11 to ESP32:

Black cable -> GND

Red cable -> 3V3

Green cable -> P4

MicroSD Card Adapter to ESP32:

CS -> P5

SCK -> P13

MOSI -> P15

MISO -> P14

VCC -> 5V

GND -> GND

Battery Charger to ESP32:

OUT+ (Put Switch here) -> 5V

OUT- -> GND

Battery to Battery Charger:

Black cable -> B-

Red cable -> B+

Sensor Data Collection

Connectivity

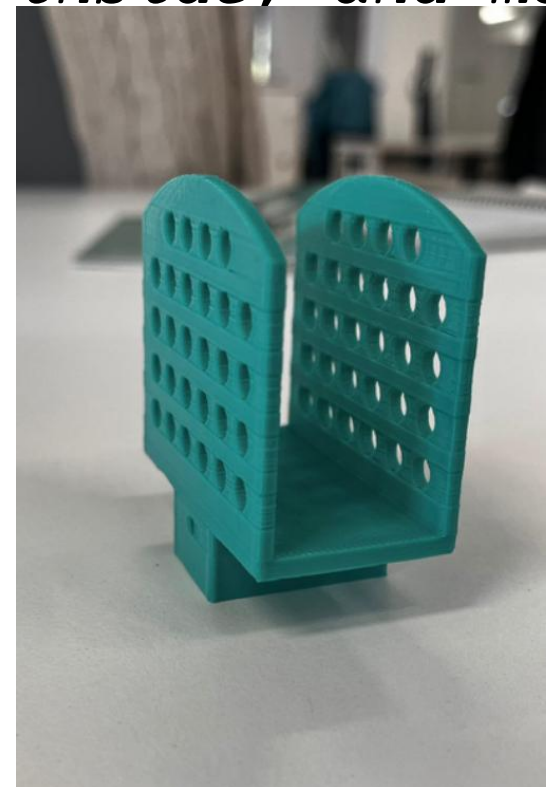
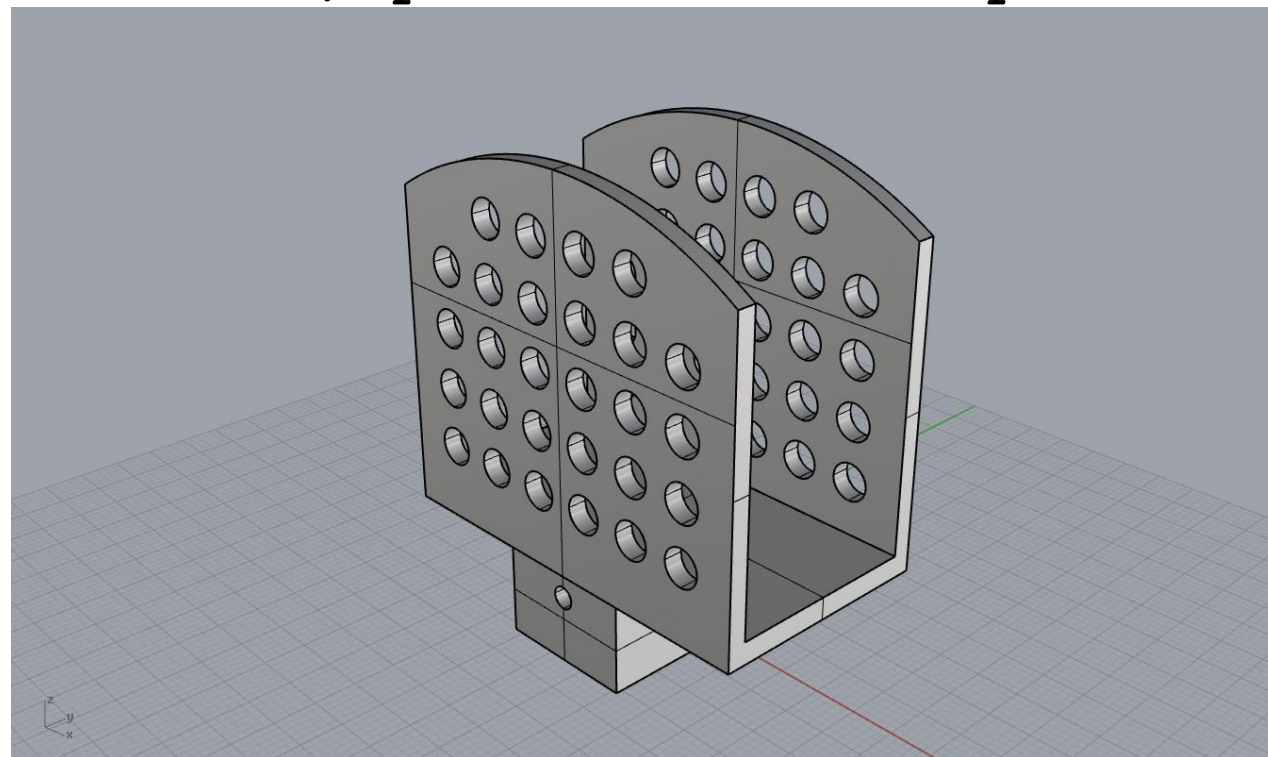
- **How to mount the sensor on the drone?**

Example: Use a 3D-printed base.

In the folder UC2 Environmental Monitoring/3D Printing Sensor Base:

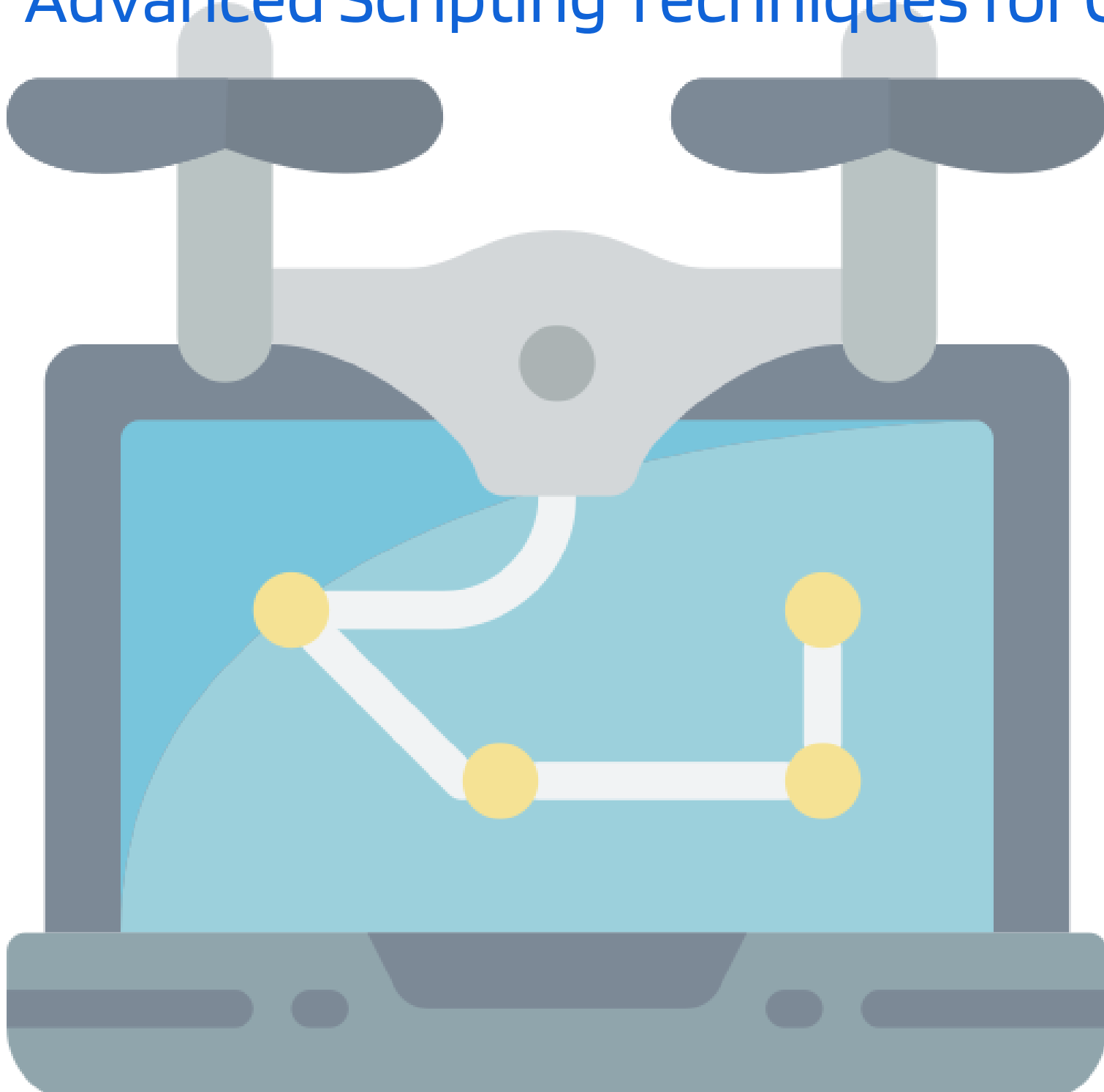
Use the .stl or the .dxf file to print your base.

Then, put all the components inside, and mount it on the drone!



Sensor Data Collection

Advanced Scripting Techniques for Complex Flight Paths



Efficient drone operation in environmental monitoring often requires advanced flight planning.

By leveraging loops and conditional programming in Scratch, we can create dynamic and adaptable flight paths.

Let's explore how to programmatically adjust a drone's movement based on sensor data, allowing for more precise and efficient environmental monitoring.

Sensor Data Collection

Advanced Scripting Techniques for Complex Flight Paths



- Using Loops and Conditions:
 - Loops can be used to repeat certain actions, like scanning the environment or measuring sensors.
 - Use the repeat until or forever block to automate actions like scanning areas or taking measurements.

Sensor Data Collection

Advanced Scripting Techniques for Complex Flight Paths

- **Example:**

when green flag clicked

repeat until $\langle \text{Temperature} \rangle > (30) >$

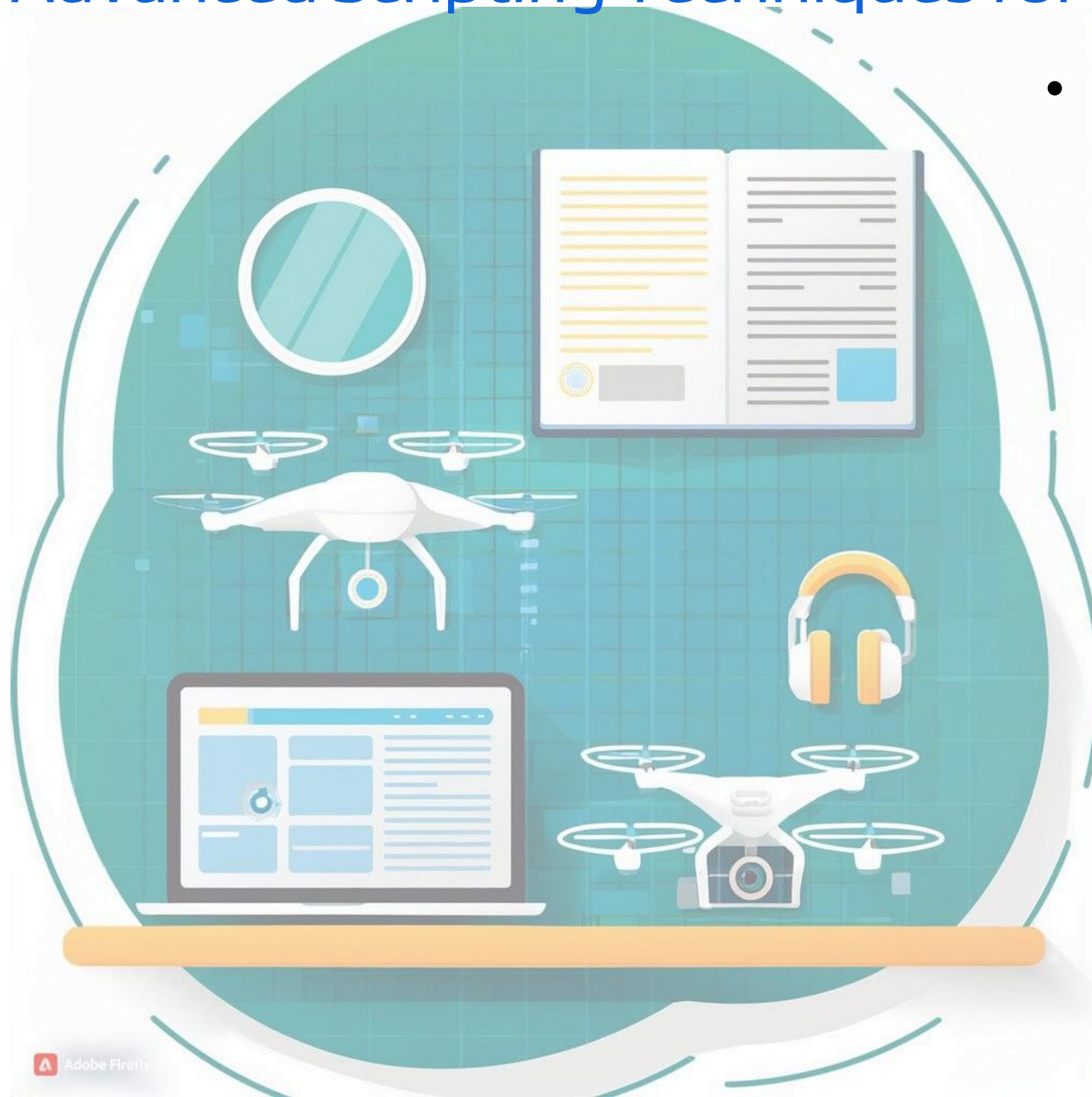
fly to x: 50 y: 30 z: 10

wait (2) seconds

set [Temperature v] to (read temperature sensor)

end

land



Sensor Data Collection

Adding Conditions to Adjust Flight Paths:

Adjusting a drone's flight path ensures more efficient and precise environmental monitoring.

By integrating sensors like temperature, altitude, or distance sensors with programming tools **drones can make automated decisions during their flight.**

Using conditional blocks in Scratch, these adjustments are easily programmed, creating dynamic and adaptive flight paths.

Examples:

- *If a drone detects a temperature above a certain threshold, it can increase altitude to avoid overheating or hazardous conditions.*
- *Altitude sensors can help the drone navigate uneven terrain by maintaining a safe distance from obstacles.*

Sensor Data Collection

Adding Conditions to Adjust Flight Paths:

Example: The drone adjusts altitude based on a temperature threshold. If the temperature exceeds 30°C, the drone will ascend by 10 meters. Otherwise, it will continue its current flight path

```
when green flag clicked
take off
forever
  set [temperature] to (sensor reading from temperature sensor)
  if <(temperature) > [30]> then
    change altitude by (10)
    say [High Temperature: Adjusting Altitude!]
  else
    say [Temperature Normal: Continuing Flight]
  end
  wait (1) seconds
end
```

The temperature variable receives data from the temperature sensor.

An “if-else” block checks whether the temperature exceeds 30°C.

If true, the drone ascends by 10 meters and displays a message.

If false, the drone continues as usual with a status update.

The forever loop ensures continuous monitoring the flight.

Introduction To SWARM Mode

Concepts and Advantages

What is Swarm Mode?

Swarm mode involves multiple drones working together, with each drone performing coordinated tasks

Advantages

Large Area Coverage: Drones can work in parallel to cover more ground.

Data Redundancy: Multiple drones collect the same data, improving accuracy.

Improved Efficiency: Tasks like mapping or data collection can be done faster.

Introduction To SWARM Mode

Communication Protocols

- In drone swarms, communication ensures synchronized movements and tasks.
- Using **broadcast** messages in Scratch, a leader drone sends commands (such as movement coordinates), to follower drones.
- This approach enables real-time coordination, allowing all drones to receive the same instructions simultaneously.

Introduction To SWARM Mode

Communication Protocols

Example:

Leader Drone (Broadcasts Movement Command):

when green flag clicked broadcast [Move to x: 50 y: 30 z: 10 v]

Follower Drone (Responds to Broadcast and Moves):

when I receive [Move to x: 50 y: 30 z: 10 v] fly to x: 50 y: 30 z: 10

Introduction To SWARM Mode

Communication Protocols

Why Use Broadcasts?

Broadcast messages in Scratch simplify communication between drones in a swarm.

Instead of addressing each drone individually, the leader sends a single message to all followers.

This approach minimizes complexity and ensures synchronized execution, especially useful in scenarios requiring coordinated movements or simultaneous tasks, such as covering a large area

Introduction To SWARM Mode

Setting Up Multiple Drones for Swarm Operations



Coordinating Drones:

Set up specific roles:

Leader and Followers: In a swarm operation, each drone has a specific role to play. The leader drone takes on the responsibility of guiding the swarm, while the follower drones simply follow the leader's commands. This structure mimics natural swarm behaviors, like those seen in bird flocks or bee colonies, where coordination is achieved through leadership and follower systems.

Introduction To SWARM Mode

Setting Up Multiple Drones for Swarm Operations



Leader Drone: Sends coordinates or commands:

The leader drone is responsible for setting the direction of the swarm. It sends commands to the follower drones, such as the next waypoint or actions.

Allows for coordinated movement across larger areas and ensures the swarm performs tasks efficiently.

The leader may also adjust commands based on environmental factors or sensor data, such as changing the flight path to avoid obstacles or hazards.

Introduction To SWARM Mode

Setting Up Multiple Drones for Swarm Operations



Follower Drone: Listens for commands and acts accordingly:

The follower drones monitor the leader's broadcasts for movement commands and react by executing the appropriate action.

They follow the leader's instructions to maintain formation and synchronization. They ensure the swarm behaves cohesively by adjusting their flight paths based on the leader's actions, allowing for tasks like environmental monitoring, surveying, or mapping to be carried out systematically.

Introduction To SWARM Mode

Synchronizing Data Collection Across the Swarm

Ensure all drones collect the same environmental data (by synchronizing)

Synchronizing data collection across a swarm is critical to ensure consistency and accuracy. All drones should be equipped with identical sensors, and their sensors should be calibrated before deployment.

This ensures that the data collected by each drone is comparable and reliable.

Synchronization also minimizes the risk of missing data or



Introduction To SWARM Mode

Synchronizing Data Collection Across the Swarm

Broadcast commands to start data collection simultaneously

Using Scratch, a leader drone can broadcast a "start data collection" message to all followers.

This ensures that every drone in the swarm begins data collection at the same time, avoiding gaps or overlaps in the recorded information.



Introduction To SWARM Mode

Synchronizing Data Collection Across the Swarm

Assign specific areas to each drone for efficient coverage

To avoid duplication, each drone can be assigned a distinct area within the environment.

The leader can broadcast these assignments or include them as part of pre-programmed tasks.

This ensures comprehensive and efficient data coverage across the entire monitored region.



Introduction To SWARM Mode

Synchronizing Data Collection Across the Swarm

Real-time data transmission or onboard storage:

The drones can transmit collected data to a central hub in real time or store it onboard for retrieval after the mission.

Real-time synchronization allows the swarm to respond dynamically to changes in the environment, such as detecting pollutants or temperature spikes.



Introduction To SWARM Mode

Synchronizing Data Collection Across the Swarm

Example:

If a swarm is tasked with monitoring air quality across a large area, synchronized sensors ensure that drones collect data simultaneously at different points.

The data can then be stitched together to provide a complete picture of air quality



Additional Resources

Further Reading

<https://unece.org/environmental-monitoring>

<https://www.pem.tuc.gr/en/index.php?>

Videos

[Environmental Monitoring](#)

[Environmental Monitoring and IoT](#)

[Mr Copernicus: an introduction to Europe's environment monitoring programme](#)



Thank you for your
attention !



Co-funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.