

# Lezione 4:

# Monitoraggio ambientale

## Programmazione a blocchi

Drones4STEM © 2023 di Drones4STEM è concesso in licenza con licenza CC BY-NC-SA 4.0. Per visualizzare una copia di questa licenza, visitare <https://creativecommons.org/licenses/by-nc-sa/4.0/>

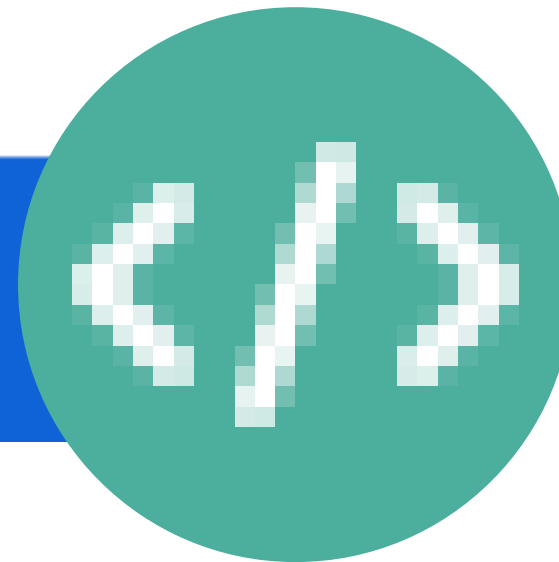


Co-funded by  
the European Union

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.

# Obiettivi della lezione

Imparare le basi della  
programmazione a blocchi



SCRATCH

Imparate a usare Scratch e Mind+ per programmare  
la missione!

Mettetevi alla prova programmando una missione  
di un drone!



# Introduzione alla programmazione a blocchi

## Cos'è la programmazione a blocchi?

**Definizione:** Metodo di codifica visuale per la programmazione senza sintassi, collegando le istruzioni sotto forma di blocchi.

**Scopo principale:** essere un approccio alla programmazione adatto ai principianti, concentrandosi sui concetti fondamentali di struttura, logica e risoluzione dei problemi.



# Perché utilizzare la programmazione a blocchi?

## 1 Semplifica la codifica

- Permette ai principianti di scrivere programmi perfettamente completi senza preoccuparsi delle specificità della sintassi e delle altre complessità della programmazione testuale.

## 2 Apprendimento visivo

- Consente ai programmatori di qualsiasi età di comprendere visivamente la struttura e le connessioni della programmazione attraverso la rappresentazione a blocchi.

## 3 Incoraggia la sperimentazione e la creatività

- Con uno strumento così facile da usare, i codificatori inesperti sono rapidamente liberi di sperimentare nuove idee e di mettere alla prova le loro capacità di problem solving.

# Strumenti per la programmazione a blocchi

## 1 Scratch

- Un popolare linguaggio di programmazione a blocchi con un proprio ambiente di sviluppo. Gratuito da usare, adatto ai principianti, con molte applicazioni già pronte che si possono trovare sul sito ufficiale!

## 2 Mind+

- Mind+ è un software di programmazione gratuito basato su Scratch 3.0 che supporta vari hardware open-source come Arduino, micro: bit, mPython-esp32 e tutti i tipi di schede di controllo.
- Mind+ supporta decine di librerie di moduli di estensione di uso comune, che lo rendono adatto alla programmazione di droni.

# Tipi di blocchi

1

## MOTION

It allows you to position and move the sprite within the work area.

2

## LOOKS

It allows you to change the appearance and let the sprite speak.

3

## SOUND

It allows you to introduce sound effects.

4

## EVENTS

It allows to start the program according to the different events and manages the messages.

5

## CONTROL

It allows you to make loops and insert conditional blocks.

6

## SENSING

It allows you to manage interactions.

7

## OPERATORS

It allows you to perform mathematical and comparison operations.

8

## VARIABLES

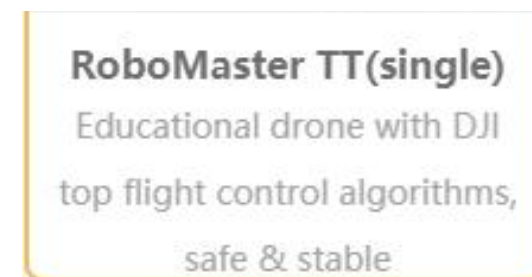
It allows you to create and manage variables.

# Programmazione a blocchi e droni

La programmazione a blocchi può essere utilizzata per programmare i dispositivi dei droni, per fargli prendere diverse traiettorie di volo, eseguire decolli e atterraggi, utilizzare sensori, scattare foto e altro ancora.

Per programmare il drone Robomaster TT in Mind+ è necessario prima installare l'estensione corretta:

Scegliere quindi



In Mind+, i blocchi utilizzati per la programmazione dei droni sono

e si trovano nella sezione



dopo aver installato l'estensione.

# Semplice esempio di programmazione di un drone basato su

I blocchi si collegano tra loro come un puzzle.

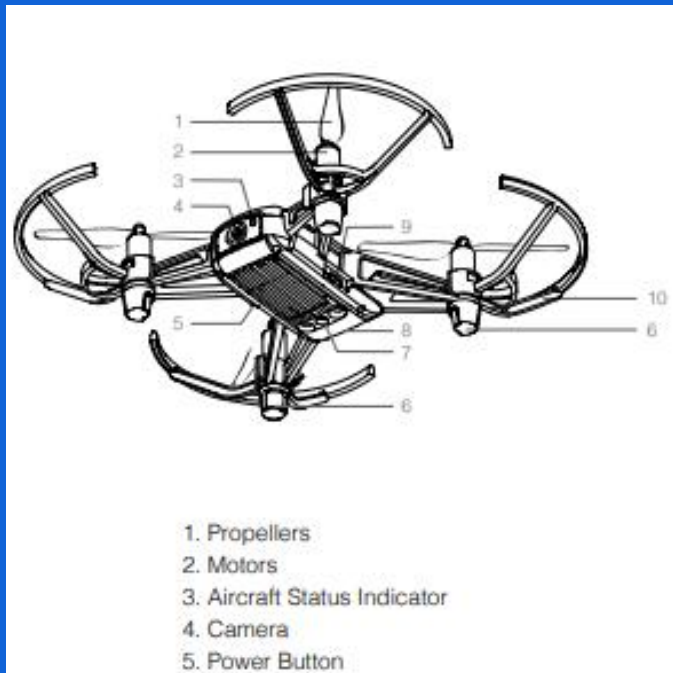
Il programmatore può trascinare e rilasciare i blocchi per creare il proprio programma.

Questo semplice esempio mostra la facilità con cui i blocchi possono essere utilizzati per programmare un drone:

decollare, volare a sinistra per 50 cm, volare a destra per 50 cm, ripetere 10 volte e poi atterrare.



# Caso d'uso 2 Missione: Collegare il drone



## Passo 1

Premere il  
pulsante di  
accensione del

Collegare il  
drone tramite  
Wifi, alla rete  
TELLO o RMTT.



## Passo 2

## Passo 3

Andare su



in Mind+ e  
premere il tasto

▶ RoboMaster TT(single) !

Collegarsi al  
dispositivo  
tramite Mind+



## Passo 4

# Caso d'uso 2 Missione: Programmazione di droni e sensori

## Scopo:

L'obiettivo della missione è di:

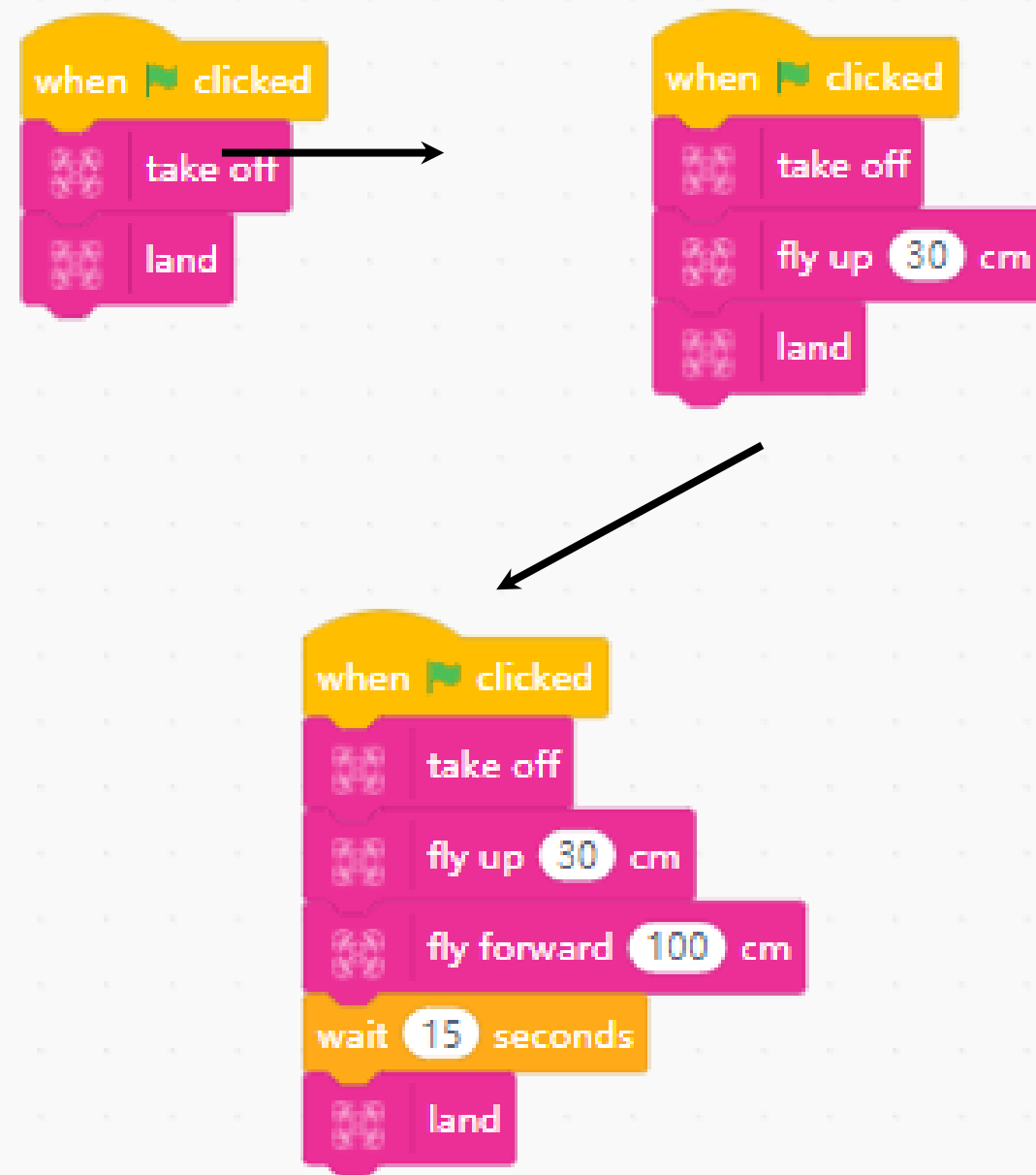
- Insegnare la programmazione di droni a blocchi
- Insegnare l'uso dei sensori con i droni
- Insegnare la raccolta e l'analisi dei

## In questa lezione di programmazione ci

- occuperemo di: drone  
Robomaster TT utilizzando Mind+.
- Raccogliere i dati dai sensori interni del drone
- Organizzare i dati in elenchi per facilitarne l'analisi

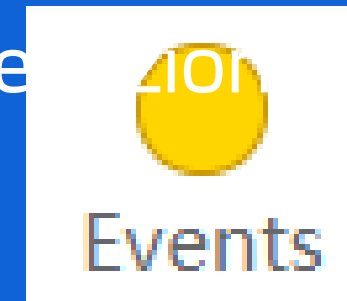
# Caso d'uso 2 Missione: Basi del volo del drone

(1)



## Iniziare

Utilizzando semplici comandi che si trovano nelle



e

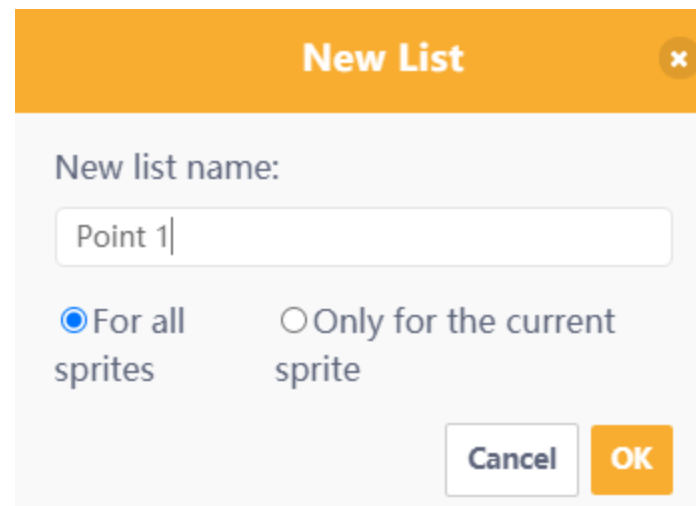


costruiamo lentamente un semplice programma per il drone:

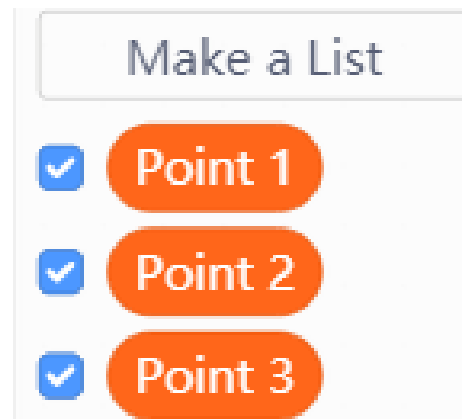
- decollare, quindi volare verso l'alto di 30 cm
- Volare in avanti di 100 cm, quindi

# Caso d'uso 2 Missione: Creazione di elenchi

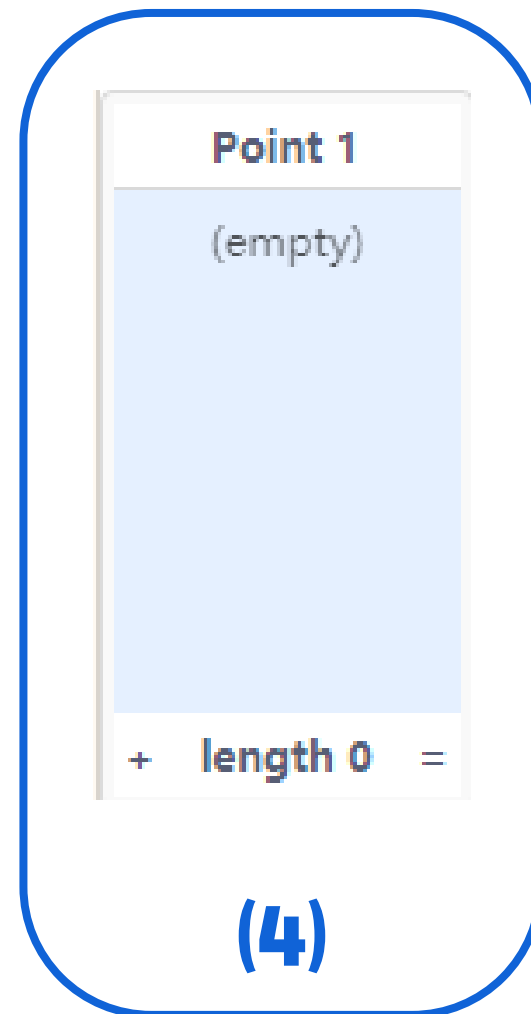
(2)



(3)

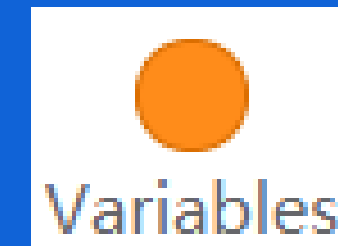


(4)



## Creazione di elenchi

Prima di utilizzare gli elenchi per memorizzare i dati, è necessario crearli.



Nella sezione

Make a List

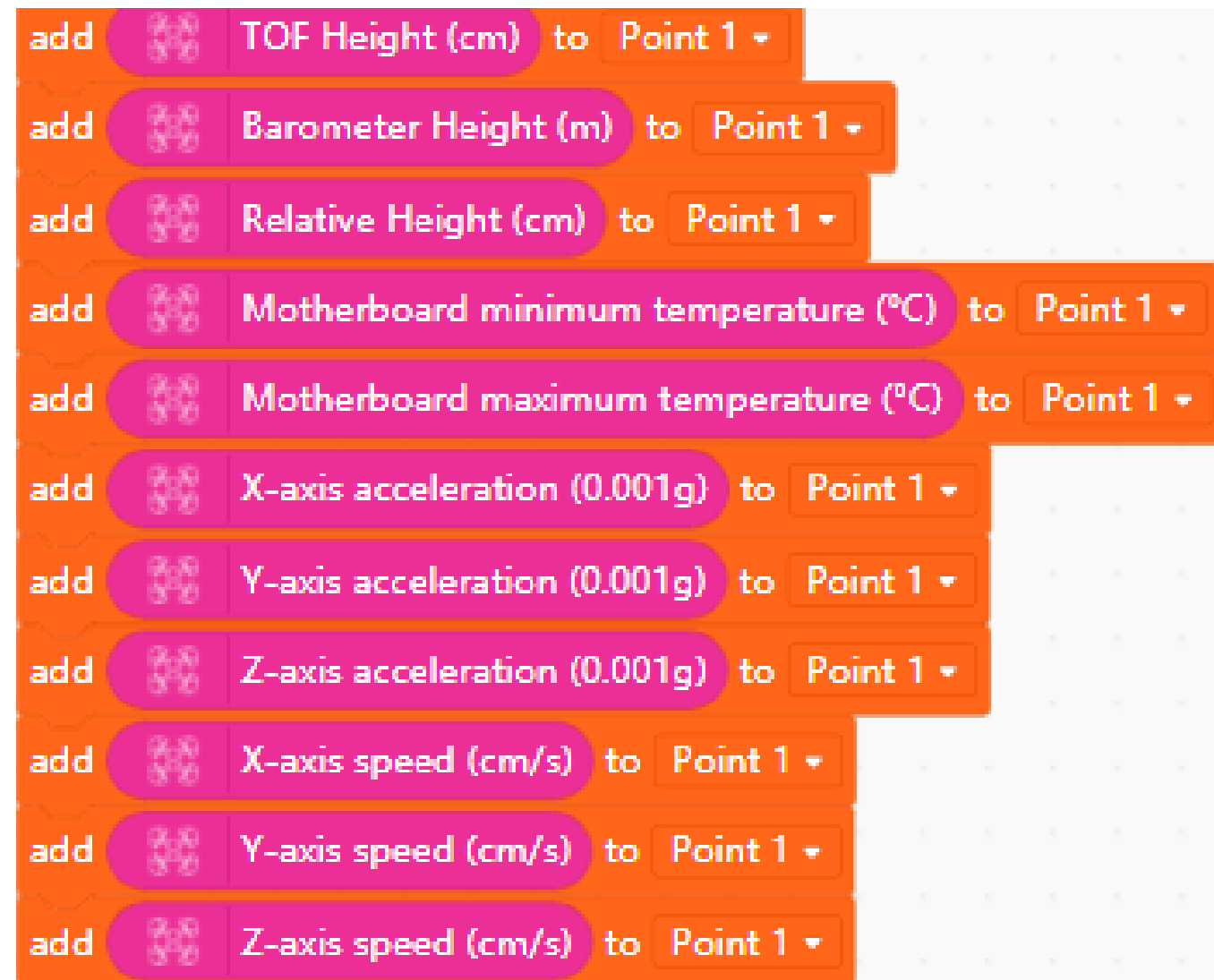
troviamo il comando che provoca il pop-up (2).

Utilizziamo questo comando per creare tre elenchi: Punto 1, Punto 2 e Punto 3, visibili in (3).

Questi elenchi appaiono vuoti nella sezione degli sprite, come si vede in (4).

# Caso d'uso 2 Missione: Riempire gli elenchi con i dati

(5)



The screenshot shows a mission editor interface with a list of data points to be added to 'Point 1'. The list includes:

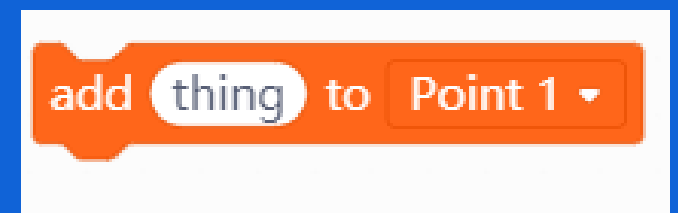
- add TOF Height (cm) to Point 1
- add Barometer Height (m) to Point 1
- add Relative Height (cm) to Point 1
- add Motherboard minimum temperature (°C) to Point 1
- add Motherboard maximum temperature (°C) to Point 1
- add X-axis acceleration (0.001g) to Point 1
- add Y-axis acceleration (0.001g) to Point 1
- add Z-axis acceleration (0.001g) to Point 1
- add X-axis speed (cm/s) to Point 1
- add Y-axis speed (cm/s) to Point 1
- add Z-axis speed (cm/s) to Point 1

## Dati negli elenchi

Dopo aver creato gli elenchi, nella sezione



si trova il comando , che può essere utilizzato per inserire diverse variabili negli elenchi.

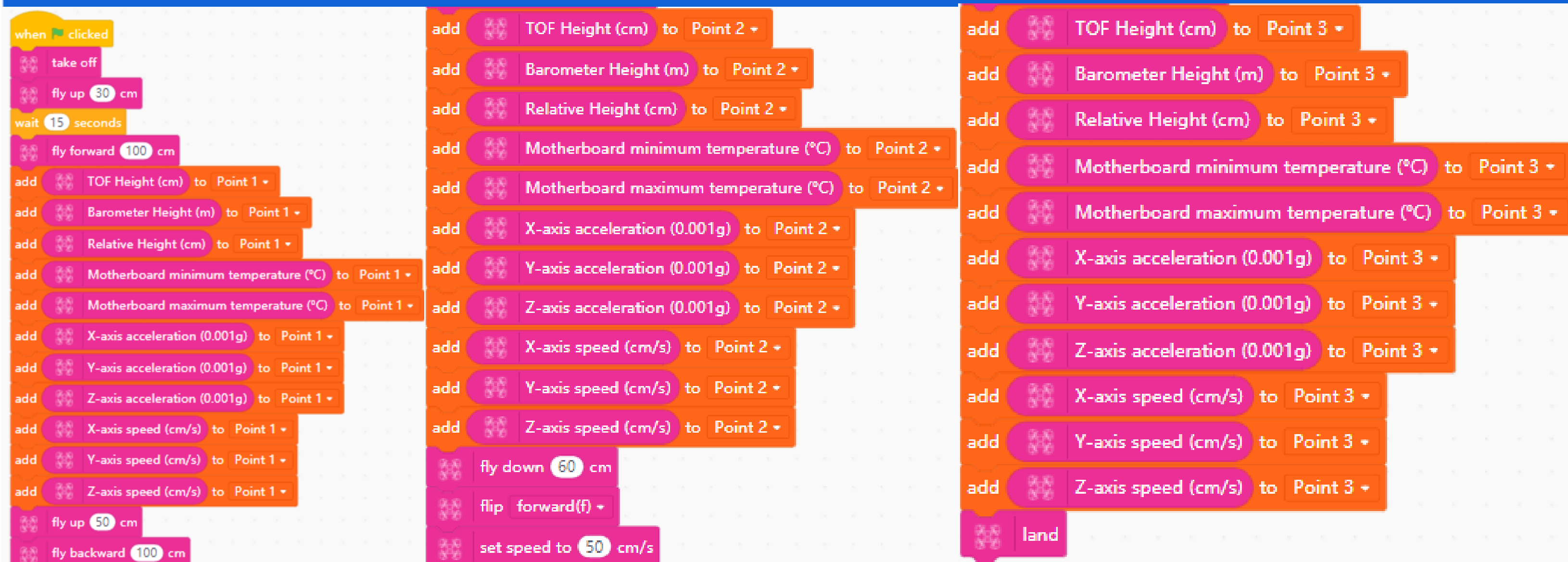


Nella sezione, troviamo variabili come



come , che utilizziamo in (5) per inserire nell'elenco Punto 1 i dati raccolti dai sensori interni del drone.

# Caso d'uso 2 Missione: Programma finale



## Combinazione

Ripetiamo il procedimento in (5) per gli elenchi Punto 2 e Punto 3. Poi combiniamo questi comandi con il nostro semplice programma in (1).

Tra una misurazione e l'altra, aggiungiamo diverse azioni del drone per garantire la varietà dei dati raccolti!

# Caso d'uso 2 Missione: Risultato finale

## Fine del programma

Dopo aver eseguito il programma, abbiamo 3 diversi elenchi con le misurazioni provenienti dai sensori del drone.

Utilizzeremo questi dati nelle fasi successive della missione.

Il processo può essere replicato utilizzando sensori esterni anziché interni, per raccogliere dati relativi all'ambiente!

| Point 1       |        | Point 2       |        | Point 3       |        |
|---------------|--------|---------------|--------|---------------|--------|
| 1             | 107    | 1             | 220    | 1             | 154    |
| 2             | 37.84  | 2             | 38.22  | 2             | 38.02  |
| 3             | 80     | 3             | 130    | 3             | 120    |
| 4             | 71     | 4             | 71     | 4             | 69     |
| 5             | 74     | 5             | 73     | 5             | 72     |
| 6             | -32.00 | 6             | -32.00 | 6             | -34.00 |
| + length 11 = |        | + length 11 = |        | + length 11 = |        |

# Caso d'uso 2 Missione: Bonus



## Utilizzo di sensori esterni

Per utilizzare i sensori esterni, aggiungeremo una sezione bonus al Caso d'uso. Questo richiede:

- Un sensore di temperatura e umidità DHT11
- Una scheda di sviluppo ESP32
- Una scheda breakout per Micro SD Card Mini
- Una batteria (Li-Po - 3,7v 800mAh)
- Un caricabatterie (1A USB-C - TP4056)
- Un interruttore (Mini SPDT - 1A/250VAC)
- Una scheda microSD (SDSQUA4-032G-GN6MA)

# Caso d'uso 2 Missione: Bonus

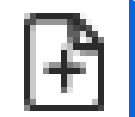


## Programmazione

Per utilizzare un sensore esterno, ci sposteremo nella sezione Codice di Mind+ per scrivere il codice Python.

Il file main.py del caso d'uso è quello che utilizzeremo.

Per creare un nuovo file Python, fare clic sull'icona accanto a "File nel progetto", quindi scrivere o incollare il codice.



Il file main.py programmerà la raccolta dei dati dal sensore.

# Caso d'uso 2 Missione: Bonus

```
import machine
import dht
import time
import os
from machine import SPI, Pin
import sdcard

# DHT22 Sensor Setup
DHT_PIN = 4
dht_sensor = dht.DHT22(machine.Pin(DHT_PIN))

# SD Card Configuration
CS_PIN = 5 # Confirm this matches your wiring
spi = SPI(1,
          baudrate=400000, # Reduced baudrate for stability
          sck=Pin(13),
          mosi=Pin(15),
          miso=Pin(14))

try:
    sd = sdcard.SDCard(spi, Pin(CS_PIN))
    os.mount(sd, '/sd')
    print("SD card mounted successfully")
    print("Root directory:", os.listdir('/'))
    print("SD card contents:", os.listdir('/sd'))
except Exception as e:
    print("SD card initialization failed:", e)
    machine.reset()
```

```
# File Configuration
FILE_PATH = '/sd/data.csv'

# Test file operations
try:
    with open(FILE_PATH, 'a') as f:
        f.write('Test entry\n')
        print("File write test successful")
except Exception as e:
    print("File operation failed:", e)

def log_data(temp, hum):
    try:
        timestamp = "{:02d}::{:02d}::{:02d}".format(*time.localtime())
        data = f"Temp {temp:.1f},Hum {hum:.1f}\n"
        with open(FILE_PATH, 'a') as f:
            f.write(data)
        print(f"Data logged: {data.strip()}")
    except Exception as e:
        print("Logging failed:", e)

while True:
    try:
        dht_sensor.measure()
        temp = dht_sensor.temperature()
        hum = dht_sensor.humidity()
        log_data(temp, hum)
    except OSError as e:
        print("Sensor read error:", e)

    time.sleep(10)
```

## Il codice

Il codice python in main.py si occupa delle seguenti operazioni:

- Impostazione del sensore DHT11
- Configurazione della scheda SD
- Creazione di un file .csv per i dati
- Registrare le misure del sensore nel file .csv

# Caso d'uso 2 Missione: Bonus

## File in the project



 main.py

 sdcard.py

## Libreria SD

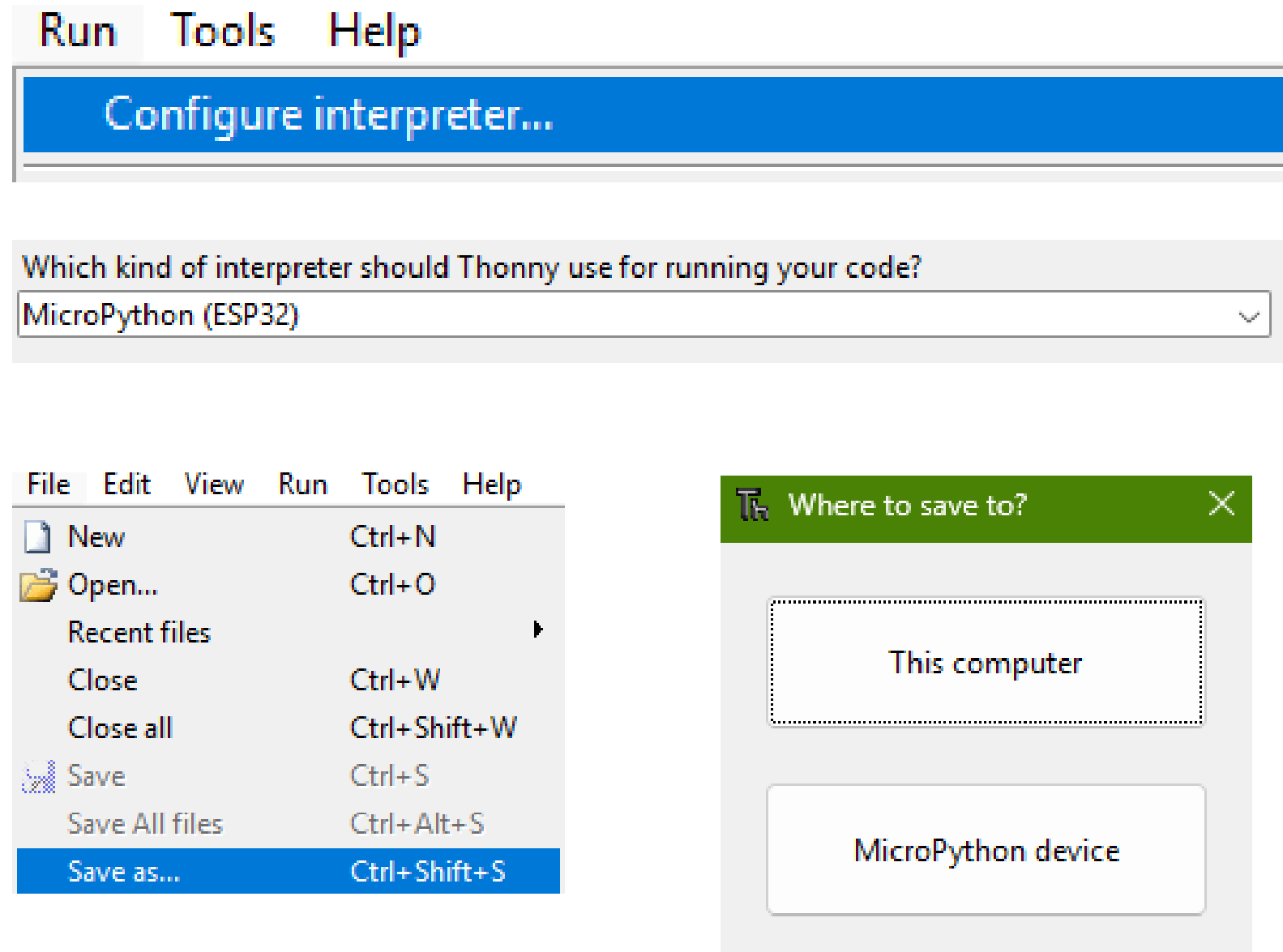
Affinché il codice in main.py funzioni, è necessaria una libreria aggiuntiva.

La libreria sdcard.py si trova a questo link:

<https://github.com/micropython/micropython-lib/blob/master/micropython/drivers/storage/sdcard/sdcard.py>

Copiare il codice e incollarlo in un nuovo file python chiamato sdcard.py.

# Caso d'uso 2 Missione: Bonus



## Caricamento del codice nella scheda di sviluppo

Un modo per salvare i file `main.py` e `sdcard.py` nella scheda di sviluppo è l'IDE Thonny, attraverso i seguenti passaggi:

- Passo 1: aprire il file `main.py` con Thonny.
- In Thonny, fare clic su Esegui, quindi su Configura Inteprete e scegliere MicroPython ESP32. Premere OK.
- Quindi, fare clic su File, Salva con nome e scegliere l'opzione MicroPython.
- Ripetere l'operazione per il file `sdcard.py`.

Ora il codice si trova nella scheda di sviluppo e verrà eseguito automaticamente ogni volta che si accende l'interruttore!

# Domande

- In che modo la programmazione a blocchi rende la programmazione più accessibile?
- Perché il drone ha dei sensori interni?
- In che modo la programmazione a blocchi di un drone può essere utilizzata per il monitoraggio ambientale?

Domande?



# Risorse aggiuntive

## Ulteriori letture

<https://mindplus.dfrobot.com/RMTT>

## Video

[Introduzione al software di programmazione Mind+ : A Sctrach 3.0](#)



Grazie per la vostra  
attenzione!



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.



