

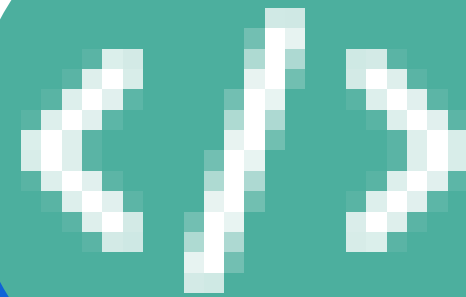
Lekcija 4:

Spremljanje okolja

Programiranje z bloki

Cilji lekcije

Učenje osnov programiranja z bloki



SCRATCH

Nauči se uporabljati Scratch in Mind+
za programiranje misije!

Preizkusi se s programiranjem misije
drona!



Uvod v programiranje z bloki

Kaj je programiranje z bloki?

Definicija: Vizualna metoda kodiranja za programiranje brez sintakse, z povezovanjem ukazov v obliki blokov.

Glavni namen: Prijazen uvod v programiranje, osredotočen na osnovne koncepte strukture, logike in reševanja problemov.



Zakaj uporabljati programiranje z bloki?

1

Poenostavi kodiranje

Omogoča začetnikom pisanje popolnih programov brez skrbi glede sintakse in drugih zapletenosti programiranja na osnovi besedila.

2

Vizualno učenje

Omogoča programerjem vseh starosti, da vizualno razumejo programsko strukturo in povezave s pomočjo predstavitve blokov.

3

Spodbuja eksperimentiranje in ustvarjalnost

Zaradi enostavne uporabe lahko začetniki hitro preizkušajo nove ideje in svoje sposobnosti reševanja problemov.

Orodja za programiranje z bloki

1

Scratch

Priljubljen jezik programiranja z bloki z lastnim razvojnim okoljem. Brezplačen za uporabo, prijazen do začetnikov, z mnogimi že pripravljenimi aplikacijami na uradni spletni strani!

2

Mind+

Brezplačen programski paket, ki temelji na Scratch 3.0 in podpira različne odprtokodne strojne naprave, kot so Arduino, micro:bit, mPython-esp32 in številne krmilne plošče.

Mind+ podpira več deset pogosto uporabljenih knjižnic razširitvenih modulov, kar ga naredi primerne za programiranje dronov.

Vrste blokov

1

GIBANJE

Dovoljuje premike in položaj za delo znotraj delovnega področja

2

IZGLED

Omogoča ti, da spremeniš videz in da lik spregovori.

3

ZVOK

Omogoča spremembe zvočnih efektov.

4

DOGODKI

Omogoča spreminjanje in aktiviranje na podlagi različnih ukazov.

5

NADZOR

Omogoča gibanje nadzor na podlagi različnih blokov.

6

ZAZANAVANJE

Omogočanje in upravljanje interakcij

7

OPERACIJE

Omogoča izvajanje matematičnih operacij.

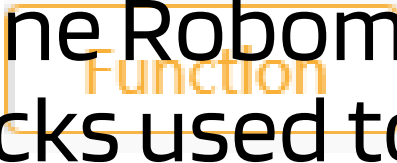
8

VARIABLE

Omogoča ustvarjanje in uporabo novih variabel

Programiranje z bloki in droni

Block-based programming can be used to program drone devices, directing them to take different flight paths, perform take-offs and landings, use sensors, take pictures and more.

To program the drone Robomaster TT in Mind+ you must first install the correct extension:
In Mind+, the Blocks used to program drones are
Choose  then  then 

and can be found in the



section after installing the extension.

Programiranje z bloki in droni

Programiranje z bloki se lahko uporablja za programiranje dronov – usmerjanje v različne letalne poti, izvajanje vzletov in pristankov, uporabo senzorjev, fotografiranje in še več.

Za programiranje drona Robomaster TT v Mind+ je potrebno najprej namestiti pravo razširitev:



V Mind+ so bloki za programiranje dronov  in jih najdemo v določenem razdelku  po namestitvi razširitve.

The text explains that in Mind+, there are blocks for programming drones, represented by a pink rounded rectangle labeled 'Fuchsia'. These blocks are found in a specific section, indicated by a 'Function' icon (a grey square with a green wrench and screwdriver), after installing the extension.

Preprost primer programiranja drona z bloki

Bloki se povezujejo med seboj kot sestavljanica.

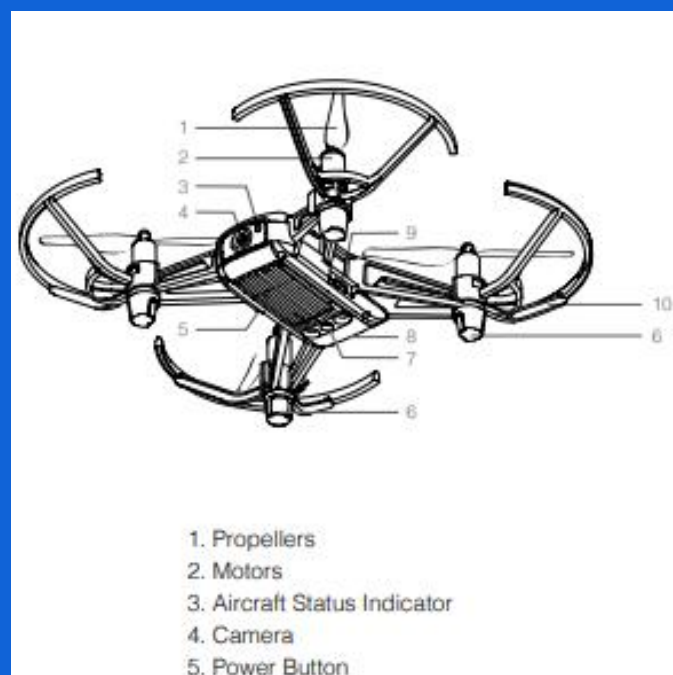
Programer lahko povleče in spusti bloke, da ustvari svoj program.

Ta preprost primer prikazuje, kako enostavno se lahko dron sprogramira za:

- vzlet,
- let levo za 50 cm,
- let desno za 50 cm,
- ponovitev tega 10-krat,
- nato pristanek.



Uporaba primera 2: Povezovanje drona



Korak 1: Pritisni
gumb za vklop
drona.

Korak 2: Poveži
se z omrežjem
TELLO ali RMTT
preko WiFi.



Korak 3:

Pojdi na



V Mind+ in
pritisni



▶ RoboMaster TT(single)



Korak 4: Poveži
se z dronom
preko Mind+.



Uporaba primera 2: Programiranje drona in senzorji

Cilj misije je:

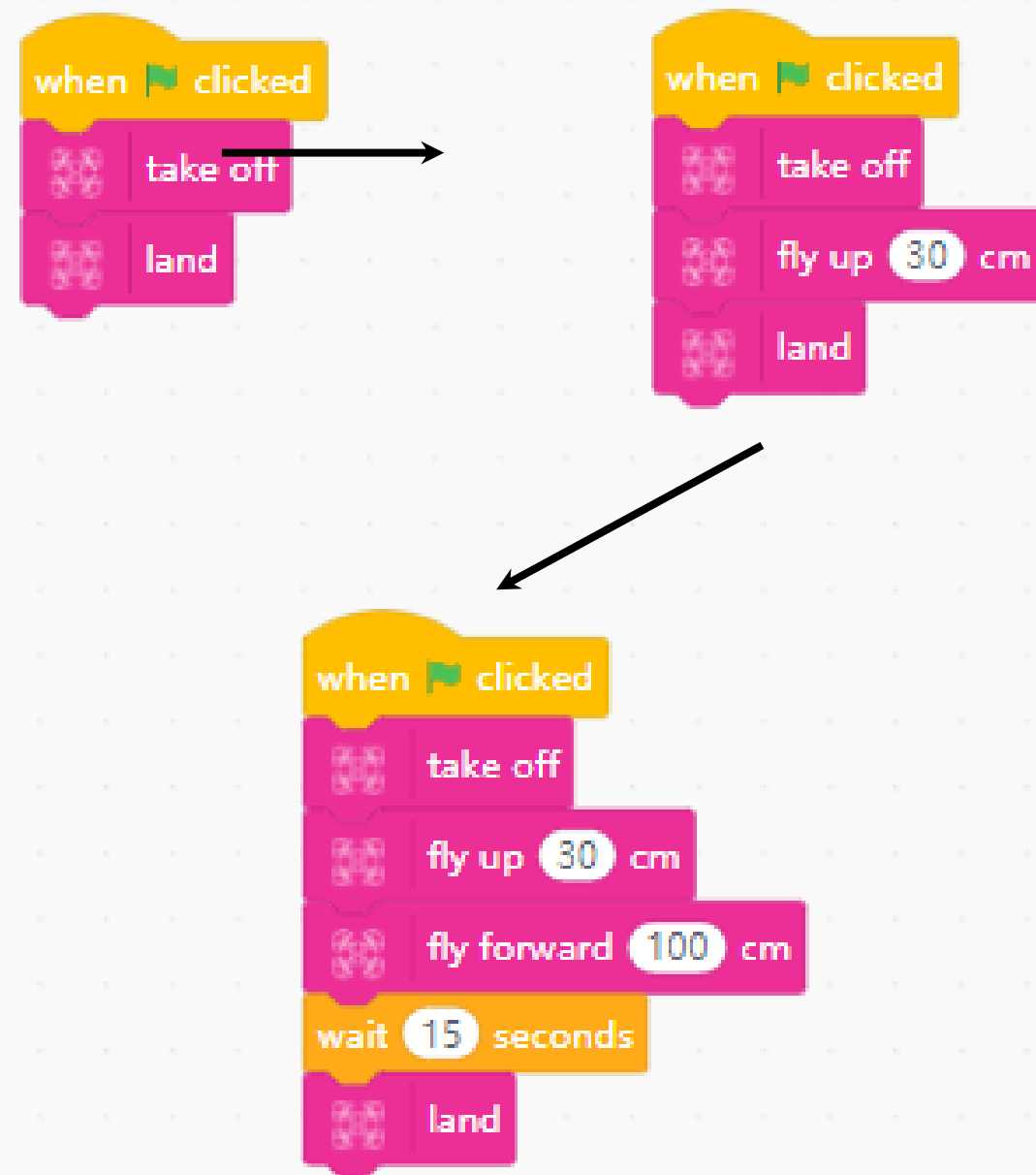
- Učiti blokovsko programiranje dronov
- Učiti uporabo senzorjev z droni
- Učiti zbiranje podatkov in analizo podatkov s senzorjev

V tej lekciji bomo:

- Programirali dron Robomaster TT z uporabo Mind+
- Zbirali podatke iz notranjih senzorjev drone
- Organizirali podatke v sezname za lažjo analizo

Uporaba primera 2: Osnove letenja z dronom

(1)



Začnemo z enostavnimi ukazi in postopoma gradimo program:



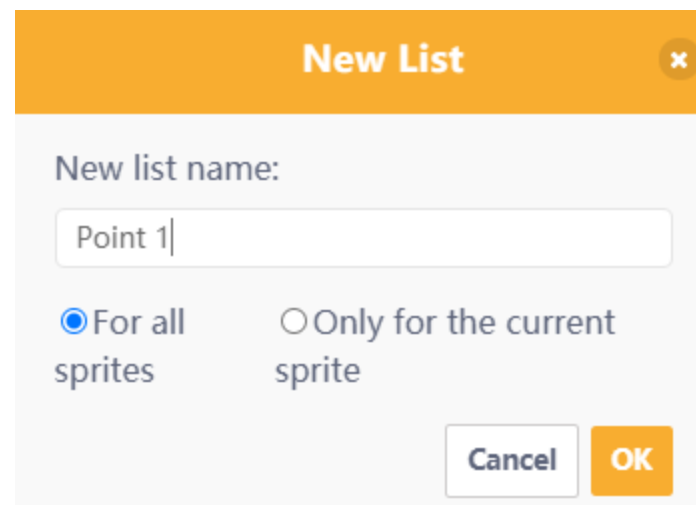
in



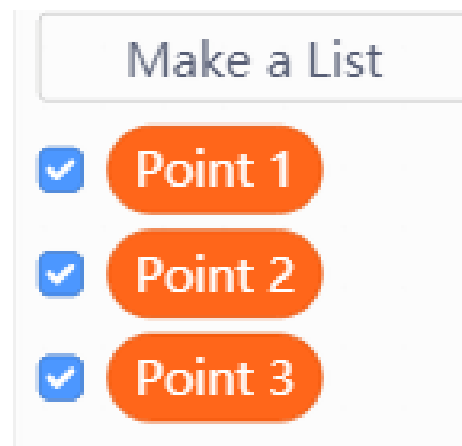
- Vzlet, nato let navzgor 30 cm
- Let naprej 100 cm, čakanje 15 sekund
- Pristanek

Uporaba primera 2: Ustvarjanje seznamov

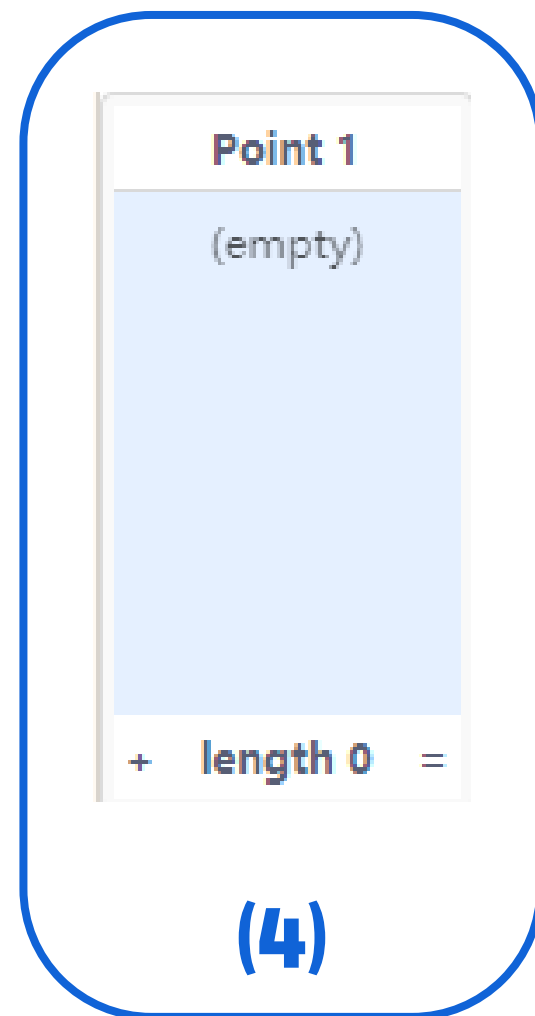
(2)



(3)



(4)



Preden uporabimo sezname (Lists) za shranjevanje podatkov, jih moramo ustvariti.

V razdelku



najdemo ukaz

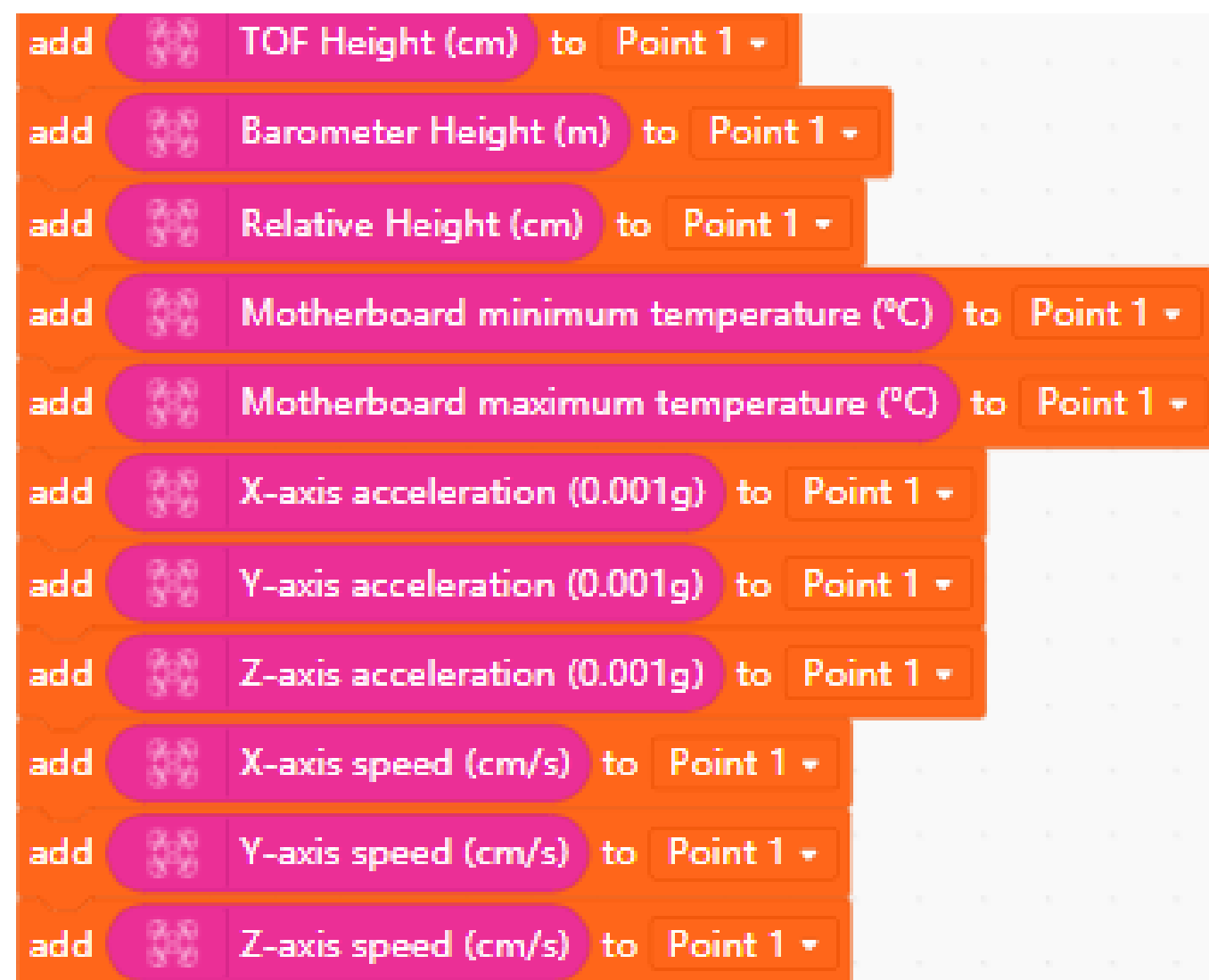


ki odpre pojavno okno (2).

Ta ukaz uporabimo za ustvarjanje treh seznamov: Point 1, Point 2, Point 3, kot je prikazano v (3). Ti sezname se prikažejo prazni v

Uporaba primera 2: Polnjenje seznamov s podatki

(5)



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 ▾

Ko ustvarimo sezname, v razdelku



najdemo ukaz



ki ga lahko

uporabimo za vstavljanje različnih spremenljivk v sezname.

V razdelku



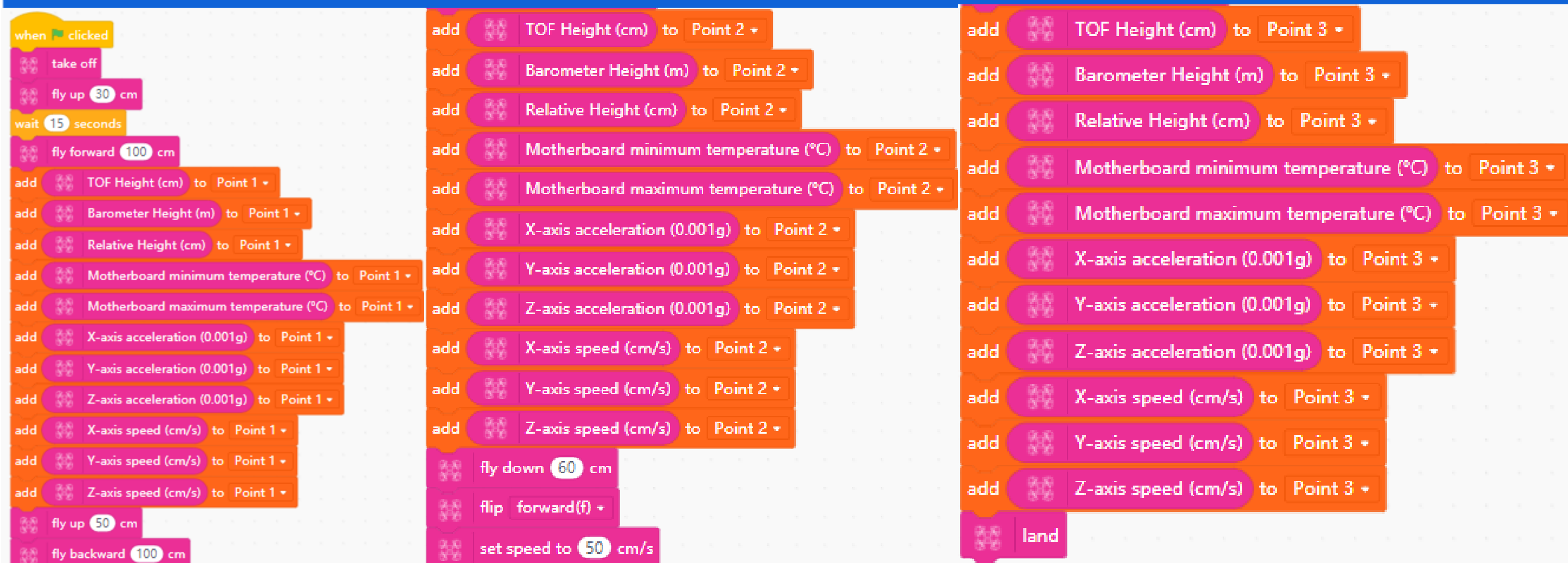
najdemo

spremenljivke, kot je



ki jo uporabimo v (5), da shranimo podatke, zbrane iz notranjih senzorjev drona, v seznam *Point 1*

Uporaba primera 2: Končni program



Postopek ponovimo za Točko 2 in Točko 3. Ukaze združimo s preprostim programom iz začetka.

Med meritvami dodamo različna dejanja drona, da zagotovimo raznolikost podatkov.

Uporaba primera 2: Končni rezultat

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 =	

Po zagonu programa imamo tri različne sezname z meritvami iz senzorjev drona.

Te podatke bomo uporabili v naslednjih fazah misije!

Postopek lahko ponovimo tudi z zunanjimi senzorji.

Uporaba primera 2: Bonus



Zunanji senzorji

Za uporabo zunanjih senzorjev potrebujemo:

- DHT11 senzor temperature in vlage
- ESP32 razvojno ploščo
- Breakout ploščico za Micro SD kartico
- Baterijo (Li-Po – 3,7V 800mAh)
- Polnilec baterij (1A USB-C – TP4056)
- Stikalo (Mini SPDT – 1A/250VAC)

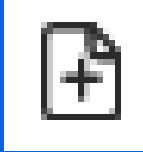
Uporaba primera 2: Bonus



Za uporabo zunanjega senzorja bomo presli v razdelek Code v Mind+, da bomo lahko pisali Python kodo.

Datoteka main.py iz primera uporabe je tista, ki jo bomo uporabljali.

Za ustvarjanje nove Python datoteke klikni na

ikono  poleg »File in the project«, nato pa napiši ali prilepi svojo kodo. Datoteka main.py bo izvajala program za zbiranje podatkov iz senzorja.

Uporaba primera 2: 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)
```

Koda

Koda v datoteki main.py

skrbi za:

- Nastavitev senzorja

DHT11

- Konfiguracijo SD

kartice

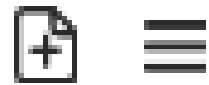
- Ustvarjanje .csv

datoteke za podatke

- Beleženje senzorjev

Uporaba primera 2: Bonus

File in the project



 main.py

 sdcard.py

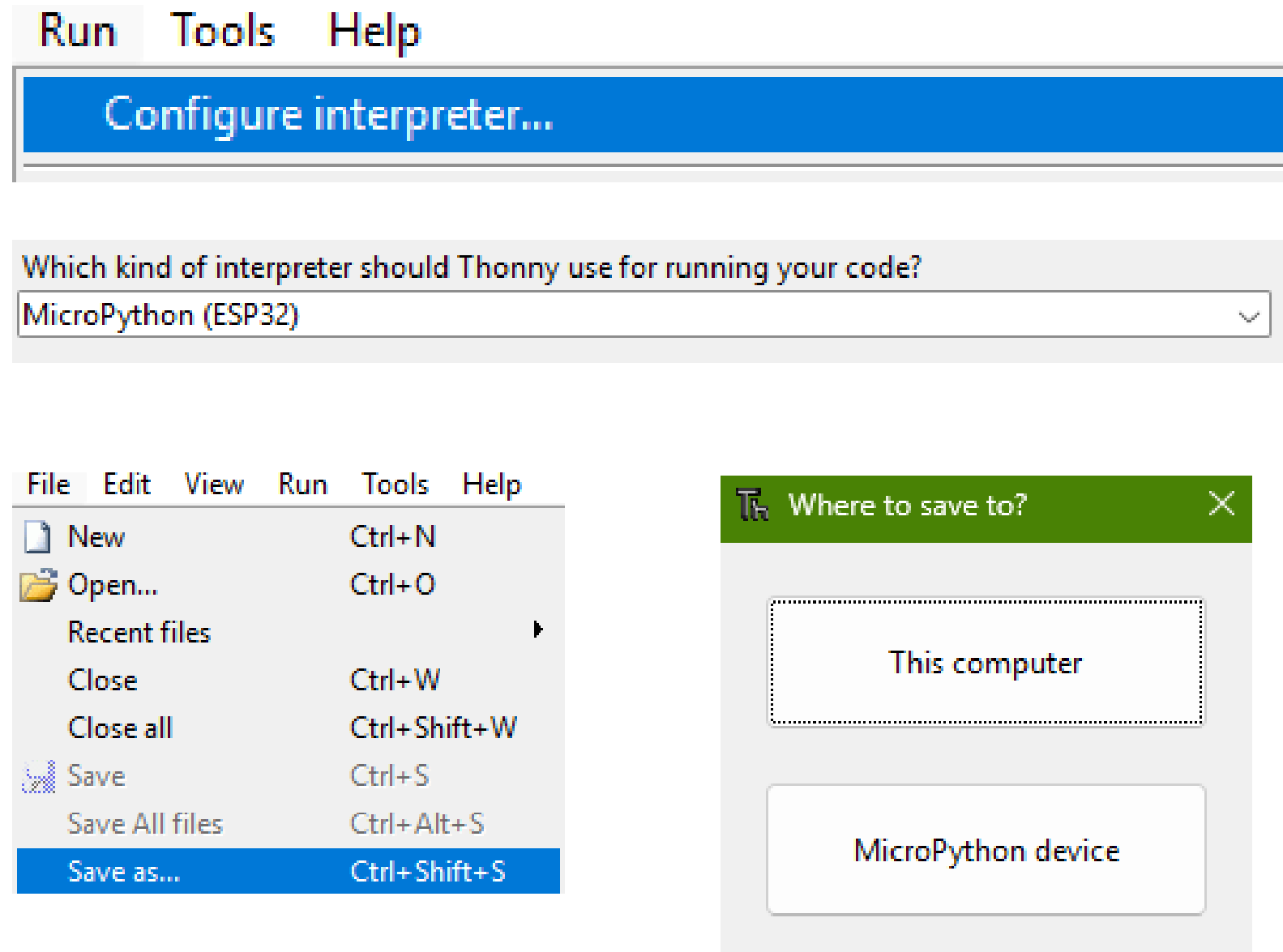
Dodatna knjižnica

Za delovanje kode v main.py potrebujemo še dodatno knjižnico. Knjižnica sdcard.py je na voljo na <https://github.com/micropython/micropython-lib/blob/master/micropython/drivers/storage/sdcard/sdcard.py> povezavi::

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

Kopiramo kodo in jo shranimo kot novo datoteko sdcard.py.

Uporaba primera 2: Bonus



Nalaganje kode na razvojno ploščo

Za shranjevanje datotek `main.py` in `sdcard.py` na razvojno ploščo uporabimo IDE Thonny:

1. Odpri datoteko `main.py` v Thonny.
2. Klikni `Run` → `Configure Interpreter`, izberi `MicroPython ESP32` in potrdi.
3. Klikni `File` → `Save As` ter izberi možnost `MicroPython`.
4. Postopek ponovi za datoteko `sdcard.py`.

Vprašanja

- Kako programiranje z bloki naredi programiranje bolj dostopno?
- Zakaj ima dron notranje senzorje?
- Kako se lahko programiranje dronov z bloki uporablja za spremljanje okolja?
- Kakšna so tvoja vprašanja?



Dodatni viri

Za nadaljnje branje:

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

Video

[Introducing Mind+ : A Sctrach 3.0 Programming Software](#)



Hvala za pozornost!



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.



