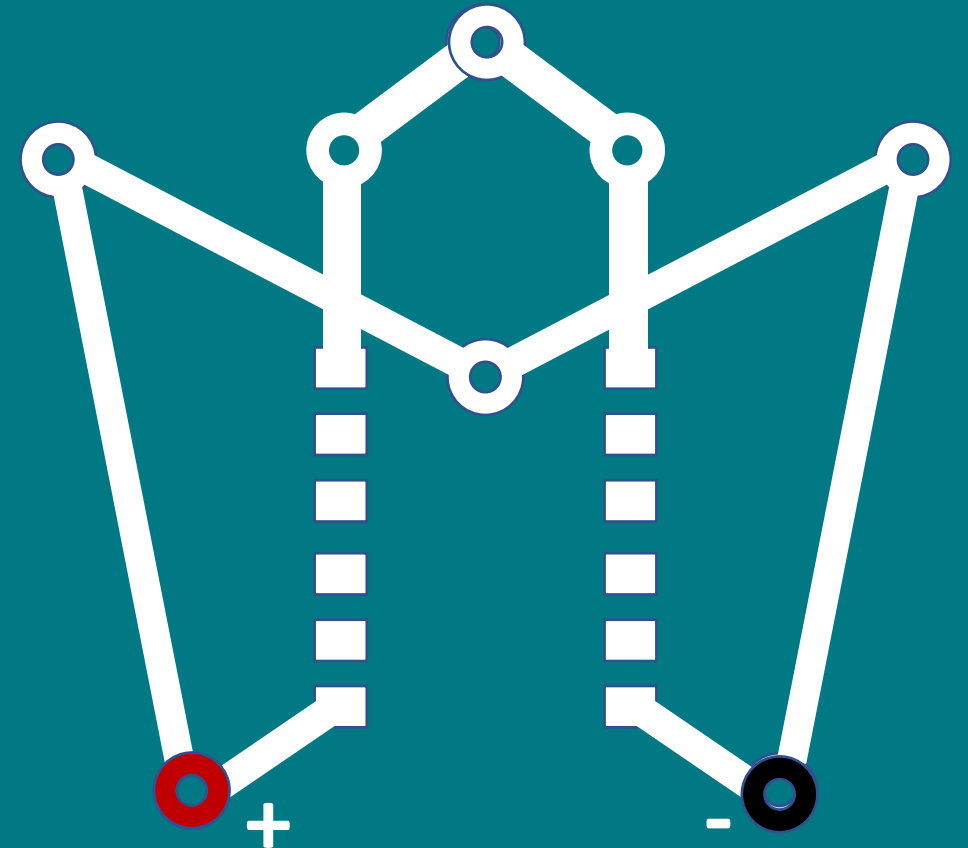


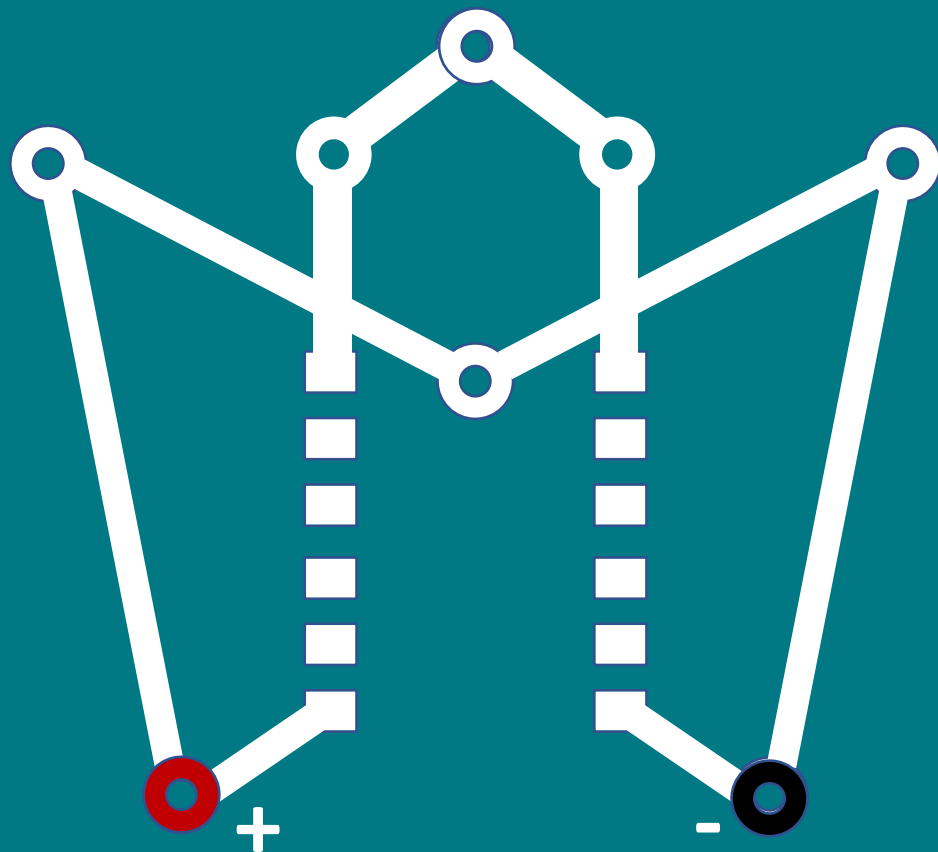
Dronovi i Internet stvari

Prof. dr Marko Simeunović, *MoDrone CTO*




MODRONE

Shapes the future of
innovation




MODRONE

Shapes the future of
innovation

Sadržaj

1. Arduino IDE
2. MoDrone Edukativna platforma
3. Tello dronovi
4. Integracija sa DJI API
5. Zaključci

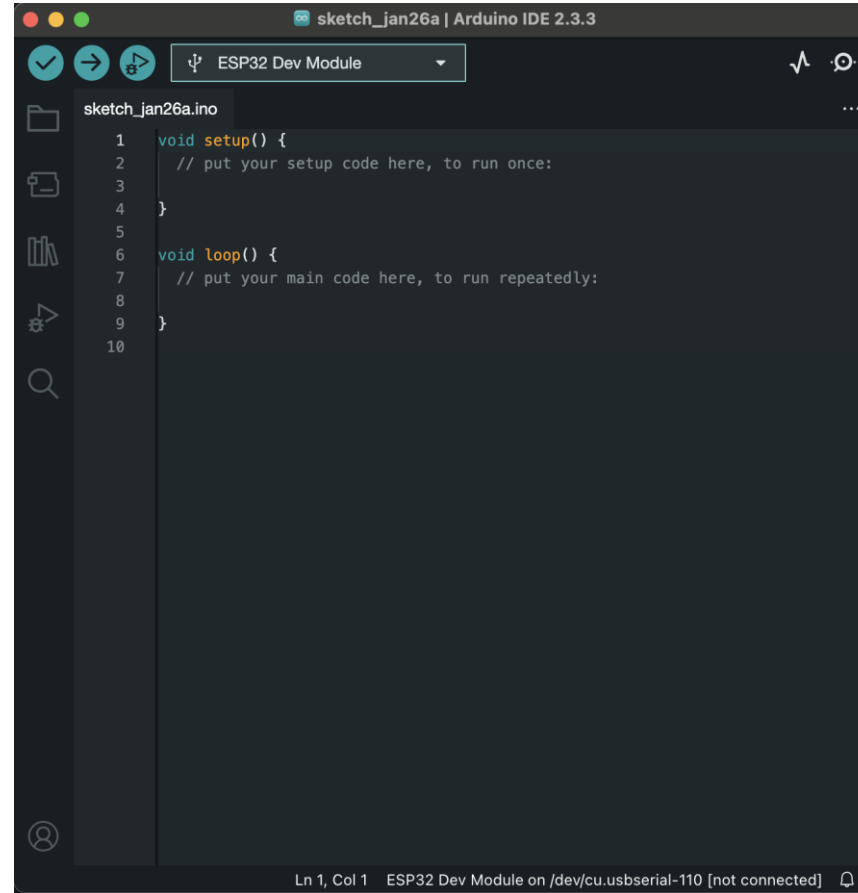
Arduino

- **Arduino** - open source platforma za razvoj aplikacija za mikrokontrolere.
- Osnovni elementi:
 - hardverska ploča sa mikrokontrolerom,
 - Arduino IDE - softversko okruženje.
- Glavne karakteristike:
 - jednostavna za korišćenje,
 - visok stepen kompatibilnosti,
 - ogromna zajednica koja razvija i održava ovu platformu.



Arduino IDE – pisanje, kompajliranje i učitavanje koda

- Ključne funkcionalnosti
 - podržava C i C++ programske jezike;
 - posjeduje serijski monitor;
 - jednostavan interfejs.
- Prednosti
 - open source - besplatan;
 - podrška za Win, Linux i Mac;
 - veliki broj biblioteka i primjera.



```
sketch_jan26a.ino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```



Arduino IDE
download



Arduino tutorial

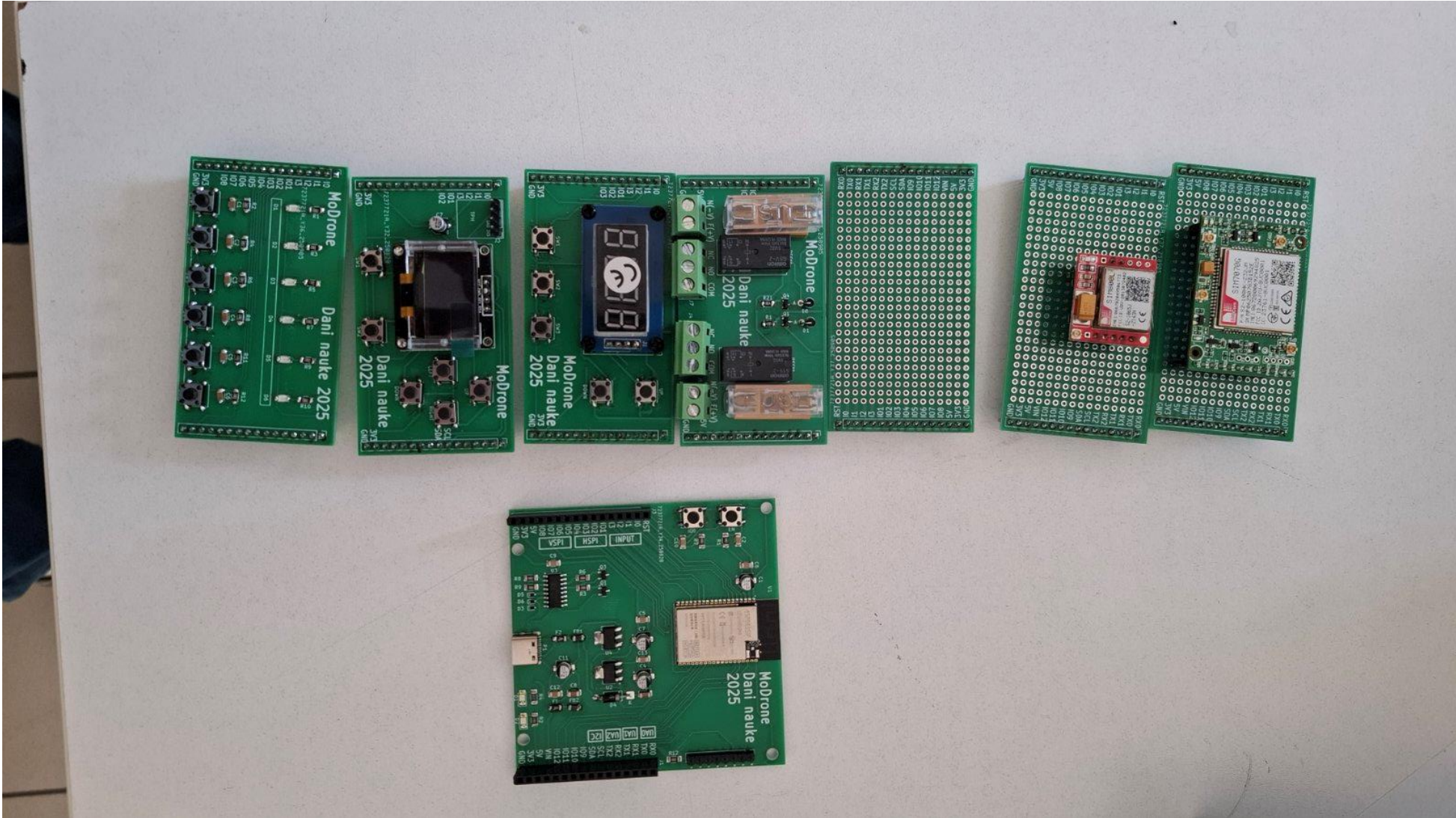
postupak



- **ESP32** - mikrokontroler sa integrisanim Wi-Fi i Bluetooth modulima.
- Proizvođač: Espressif Systems.
- Glavne karakteristike:
 - procesor sa dva jezgra i taktom do 240 MHz,
 - veliki broj ulazno-izlaznih portova,
 - podrška za sledeće protokole: ADC, DAC, I2C, SPI, UART, PWM.
- Primjena:
 - pametne kuće,
 - prenosivi uređaji,
 - automatizacija, itd.

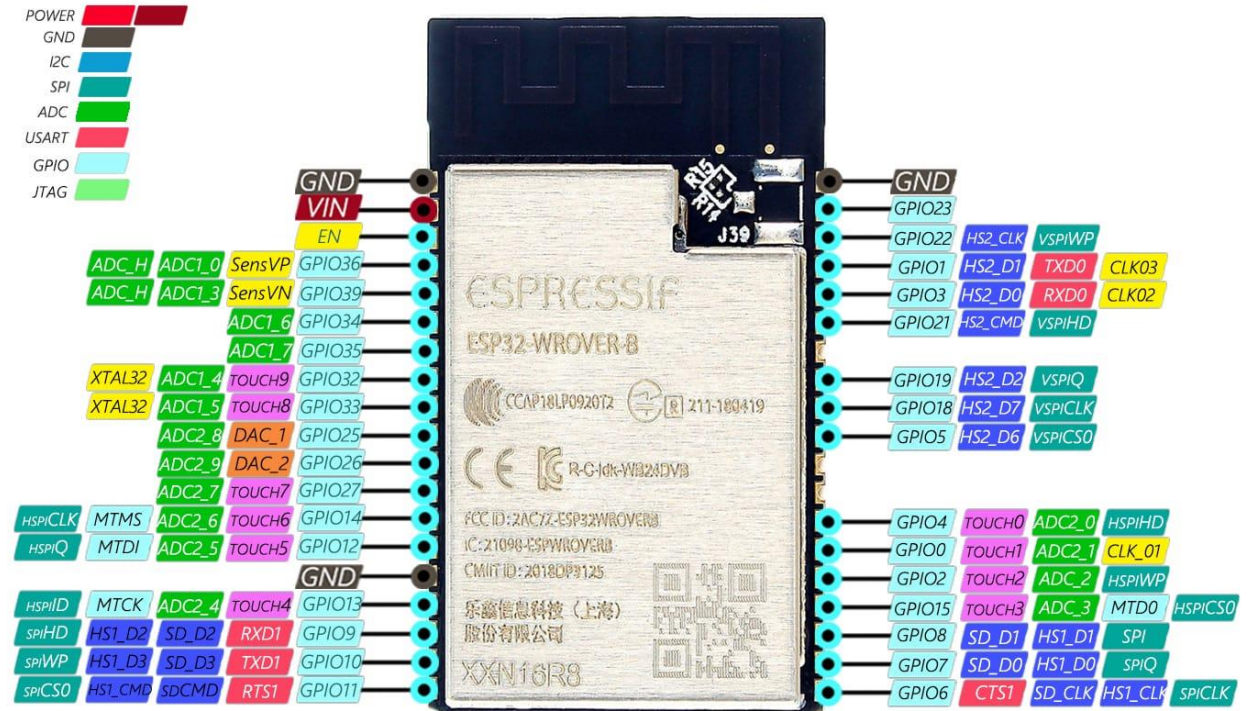


MoDrone edukativna platforma



ESP32 karakteristike

- Osnovne funkcionalnosti ESP32:
 - **Wi-Fi:** Podrška za 802.11 b/g/n protokole za bežičnu povezivost;
 - **Bluetooth:** Dual Mode (Bluetooth klasični i BLE - Bluetooth Low Energy);
 - **ulazi i izlazi:** digitalni i analogni pinovi za rad sa senzorima i aktuatorima;
 - **energetska efikasnost:** režimi spavanja (Deep Sleep) za štednju energije.;
 - **ugrađene periferije:** tajmeri, PWM, senzori dodira, RTC (Real Time Clock).



Povezivanje ESP32
sa Arduino IDE



ESP32 datasheet



Tello dron

- **Tello dron** - Mali, lagani i pametni dron namenjen za edukaciju i zabavu.
- Proizvođač: Ryze Technology, u saradnji sa DJI i Intelom.
- Primjena:
 - **edukacija:** učenje osnovnih koncepata programiranja i upravljanja dronovima,
 - **zabava:** snimanje video zapisa i fotografija u pokretu,
 - **eksperimenti:** idealan za testiranje i programiranje u STEM okruženju.



Karakteristike Tello drona:

- **Dimenzije:** 98mm x 92.5mm x 41mm,
- **Težina:** 80g (sa propelerima i baterijom),
- **Kamera:**
 - **rezolucija:** 5 MP (2592 x 1936),
 - **video:** HD 720p @30fps,
- **Karakteristike leta:**
 - **vrijeme leta:** do 13 minuta;
 - **max. brzina:** do 8m/s;
 - **visina leta:** do 10m;
- **Upravljanje:**
 - Tello app;
 - UDP protokolom.



Tello SDK



Tello tehničke
karakteristike



ESP32 Arduino
biblioteka



Python paket



Kontrola Tello drona ESP32

StoxAkademijaEDUDronPrimjer.ino

```
1  #include <TelloESP32.h>
2  using namespace TelloControl;
3  TelloESP32 tello;
4
5  #define IO 36
6  #define I1 39
7  #define I2 34
8  #define I3 35
9  #define IO1 14
10 #define IO3 13
11 #define IO4 15
12
13 void setup() {
14
15     Serial.begin(115200);
16     Serial.println("Krenuo");
17
18     pinMode(IO, INPUT_PULLUP);
19     pinMode(I1, INPUT_PULLUP);
20     pinMode(I2, INPUT_PULLUP);
21     pinMode(I3, INPUT_PULLUP);
22     pinMode(IO1, INPUT_PULLUP);
23
24     TelloStatus status = tello.connect("TELLO-A01133", "");
25     if (status == TelloStatus::OK) {
26         Serial.println("Povezan sa Tello dronom!");
27     } else {
28         Serial.print("Greska pri povezivanju sa Tello dronom: ");
29         Serial.println((int)status);
30     }
31 }
32
33 void loop() {
34     // ---- Buttons ----
35     if (digitalRead(IO) == LOW){
36         Serial.println("IO pritisnuto");
37         tello.takeoff();
38         delay(300);
39         Serial1.flush();
```

```
40     }
41     if (digitalRead(I1) == LOW){
42         Serial.println("I1 pritisnuto");
43         tello.land();
44         delay(300);
45         Serial1.flush();
46     }
47     if (digitalRead(I2) == LOW){
48         Serial.println("I2 pritisnuto");
49         tello.move_left(20);
50         delay(300);
51         Serial1.flush();
52     }
53     if (digitalRead(I3) == LOW){
54         Serial.println("I3 pritisnuto");
55         tello.move_right(20);
56         delay(300);
57     }
58     if (digitalRead(IO1) == LOW){
59         Serial.println("IO1 pritisnuto");
60         tello.rotate_clockwise(180);
61         delay(300);
62     }
63 }
64 }
```



AI sa Tello dronom

```
detekcijaLopte.py x
1 from djitellopy import Tello
2 import cv2
3 from ultralytics import YOLO
4 import time
5
6 SPORTS_BALL_CLASS_ID = 32 # COCO "sports ball"
7
8 usage
9 def pokreni_upravljanje_tello_dronom():
10     """
11     - Uvijek radi detekciju i prikazuje loptu (YOLOv8)
12     - Rotira dron lijevo/desno (yaw) samo kada je opcija rotacije aktivna
13     - Toggle rotacije tasterom 'p'
14     - Ručno poletanje/slijetanje: t/l/q
15     """
16
17     model = YOLO("yolov8n.pt")
18
19     tello = Tello()
20     tello.connect()
21     print("Battery:", tello.get_battery(), "%")
22
23     tello.streamon()
24     frame_reader = tello.get_frame_read()
25
26     flying = False
27     rotate_enabled = False # <<< samo ovo uključuje/isključuje rotiranje
28
29     # ON/OFF rotacija parametri
30     DEADBAND_PX = 35 # tolerancija oko centra (px)
31     YAW_SPEED = 35 # fiksna yaw brzina kada je uključen (-100..100)
32     RC_PERIOD = 0.10 # slanje RC komandi ~10 Hz
```

```
detekcijaLopte.py x
31
32 last_rc_time = 0.0
33 last_sent_yaw = 0
34
35 print("Controls: [t]=takeoff, [l]=land, [p]=rotate ON/OFF, [q]=quit")
36
37 while True:
38     frame = frame_reader.frame
39     if frame is None:
40         continue
41
42     frame = cv2.resize(frame, dsize=(640, 480))
43     h, w = frame.shape[:2]
44     frame_center_x = w // 2
45
46     # Uvijek radi detekciju
47     results = model.predict(
48         source=frame,
49         classes=[SPORTS_BALL_CLASS_ID],
50         conf=0.25,
51         verbose=False
52     )
53
54     annotated = frame.copy()
55     r0 = results[0]
56
57     # Podrazumijevano yaw=0; promijeniće se samo ako je rotate_enabled=True i lopta nađena
58     yaw = 0
59     ball_found = False
60     best_conf = 0.0
61     best_box = None
62     error_x = None
```

AI sa Tello dronom

```
opte.py x
# Izaberi najbolju detekciju (najveći confidence) i prikaži je UVIJEK
if r0.bboxes is not None and len(r0.bboxes) > 0:
    for b in r0.bboxes:
        conf = float(b.conf[0])
        if conf > best_conf:
            best_conf = conf
            best_box = b

if best_box is not None:
    x1, y1, x2, y2 = map(int, best_box.xyxy[0].tolist())
    ball_center_x = (x1 + x2) // 2
    ball_center_y = (y1 + y2) // 2
    ball_found = True

# Crtaj box i tekst (uvijek)
cv2.rectangle(annotated, (x1, y1), (x2, y2), (0, 255, 0), 2)
cv2.putText(
    annotated,
    text: f"ball {best_conf:.2f}",
    org: (x1, max(0, y1 - 8)),
    cv2.FONT_HERSHEY_SIMPLEX,
    fontScale: 0.6,
    color: (0, 255, 0),
    thickness: 2
)

# Linija centra ekrana
cv2.line(annotated, pt1: (frame_center_x, 0), pt2: (frame_center_x, h), color: (255, 255, 255), thickness: 1)
cv2.circle(annotated, center: (ball_center_x, ball_center_y), radius: 5, color: (255, 255, 255), -1)

# Greška po horizontali
error_x = ball_center_x - frame_center_x
```

```
te.py x
# Rotacija samo ako je opcija aktivna
if rotate_enabled:
    if error_x > DEADBAND_PX:
        yaw = +YAW_SPEED
    elif error_x < -DEADBAND_PX:
        yaw = -YAW_SPEED
    else:
        yaw = 0
else:
    yaw = 0

# Status na ekranu
status = "FLYING" if flying else "LANDED"
rot_status = "ON" if rotate_enabled else "OFF"
det_status = "YES" if ball_found else "NO"
err_txt = f"{error_x}" if error_x is not None else "N/A"

cv2.putText(
    annotated,
    text: f"Status: {status} | Ball: {det_status} | Rotate: {rot_status} | err_x: {err_txt} | yaw: {yaw}",
    org: (10, 25),
    cv2.FONT_HERSHEY_SIMPLEX,
    fontScale: 0.6,
    color: (255, 255, 255),
    thickness: 2
)

cv2.putText(
    annotated,
    text: "t=takeoff l=land p=rotate ON/OFF q=quit",
    org: (10, 50),
    cv2.FONT_HERSHEY_SIMPLEX,
    fontScale: 0.6,
    color: (255, 255, 255),
    thickness: 2
)

nje_tello_drono... while True
```



AI sa Tello dronom

```
        thickness: 2
    )

cv2.imshow( winname: "Tello + YOLOv8 Ball Detection (Rotate Toggle)", annotated)

# RC komande šalji periodično samo dok je dron u letu
now = time.time()
if flying and (now - last_rc_time) >= RC_PERIOD:
    # Ako rotacija nije omogućena ili lopta nije nađena -> yaw će biti 0
    tello.send_rc_control( left_right_velocity: 0, forward_backward_velocity: 0, up_down_velocity: 0, yaw)
    last_sent_yaw = yaw
    last_rc_time = now

key = cv2.waitKey(1) & 0xFF

if key == ord("t"):
    if not flying:
        try:
            tello.takeoff()
            flying = True
            tello.send_rc_control( left_right_velocity: 0, forward_backward_velocity: 0, up_down_velocity: 0, yaw_velocity: 0)
            last_rc_time = time.time()
            last_sent_yaw = 0
            print("Takeoff.")
        except Exception as e:
            print("Takeoff failed:", e)
    else:
        print("Already flying.")

elif key == ord("l"):
    if flying:
        try:
            tello.send_rc_control( left_right_velocity: 0, forward_backward_velocity: 0, up_down_velocity: 0, yaw_velocity: 0)
```

```
        tello.land()
        flying = False
        rotate_enabled = False # opciono: ugasi rotaciju pri slijetanju
        print("Landing.")
    except Exception as e:
        print("Landing failed:", e)
    else:
        print("Already landed.")

elif key == ord("p"):
    rotate_enabled = not rotate_enabled
    # Sigurnosno: čim ugasiš rotaciju, pošalji yaw=0
    if flying and not rotate_enabled and last_sent_yaw != 0:
        tello.send_rc_control( left_right_velocity: 0, forward_backward_velocity: 0, up_down_velocity: 0, yaw_velocity: 0)
        last_sent_yaw = 0
        last_rc_time = time.time()
        print("Rotate:", "ON" if rotate_enabled else "OFF")

elif key == ord("q"):
    if flying:
        try:
            tello.send_rc_control( left_right_velocity: 0, forward_backward_velocity: 0, up_down_velocity: 0, yaw_velocity: 0)
            tello.land()
            flying = False
            print("Landing before exit.")
        except Exception as e:
            print("Landing failed:", e)
    break

tello.streamoff()
tello.end()
cv2.destroyAllWindows()
```



AI sa Tello dronom

```
196  
197 ▶ if __name__ == "__main__":  
198     pokreni_upravljanje_tello_dronom()  
199
```



Zaključak

- Arduino, ESP32 i Tello dronovi pružaju izuzetne mogućnosti za učenje i praktičnu primjenu u oblasti IoT i dronova.
- Korisnici mogu razvijati projekte koji povezuju hardver, softver i realne aplikacije.
- Koristeći ove uređaje, moguće je realizovati raznovrsne projekte od edukativnih zadataka do profesionalnih IoT rješenja.
- Ove tehnologije pružaju jednostavnost korišćenja i dostupnost resursa, što ih čini idealnim za početnike i profesionalce.



Hvala na pažnji!



Shapes the future of
innovation