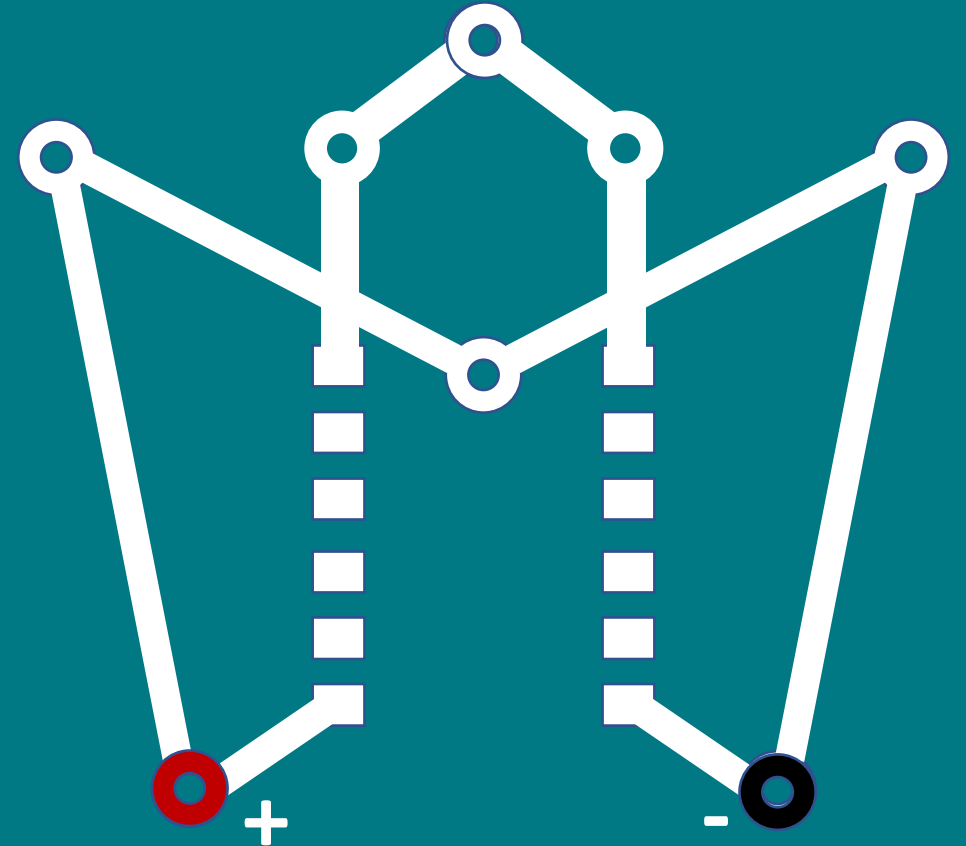


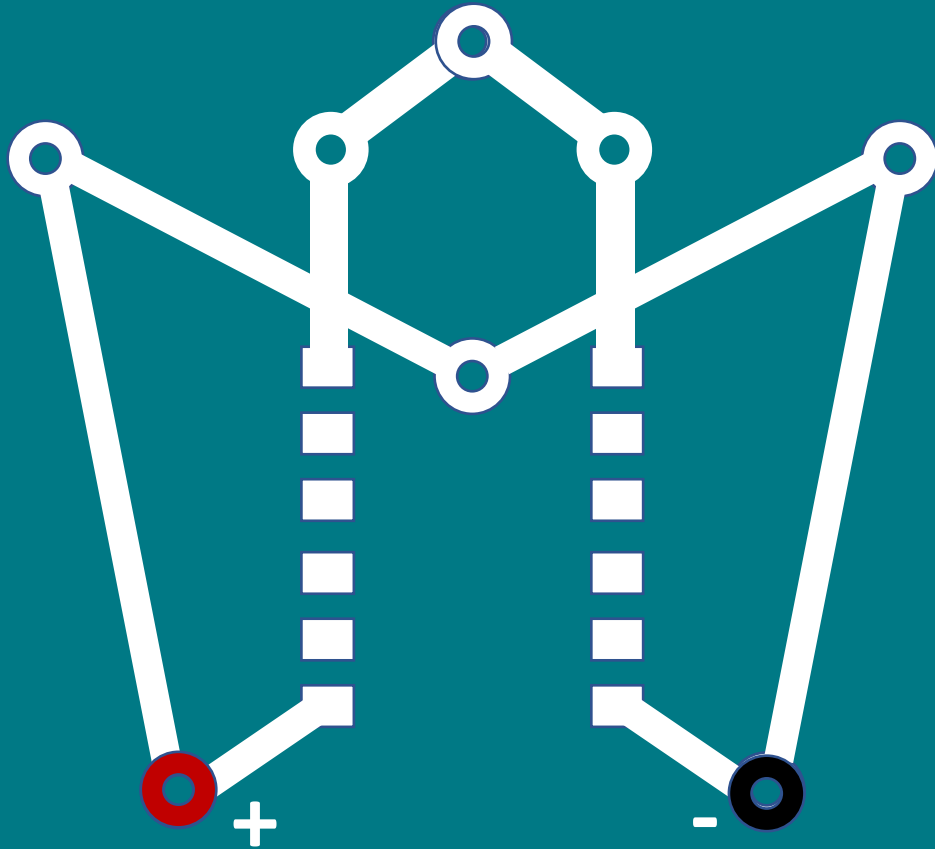
Dronovi i Internet stvari

Prof. dr Marko Simeunović, *MoDrone CTO*



MODRONE

Shapes the future of
innovation




M O D R O N E

Shapes the future of
innovation

Sadržaj

1. Arduino IDE
2. ESP32 mikrokontroler
3. Kontrola BNO055 multi senzora
4. Upravljanje Tello dronom
5. Zaključci

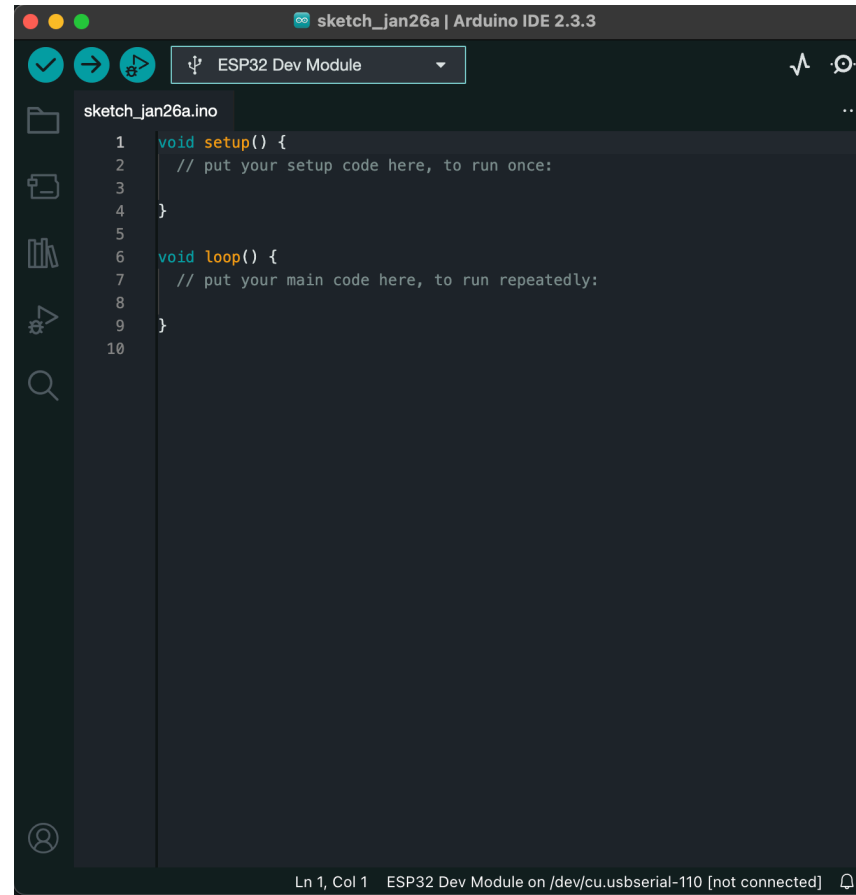
Arduino

- **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
```

Ln 1, Col 1 ESP32 Dev Module on /dev/cu.usbserial-110 [not connected]



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.



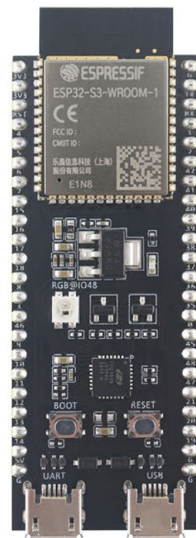
ESP32 razvojne ploče - varijante



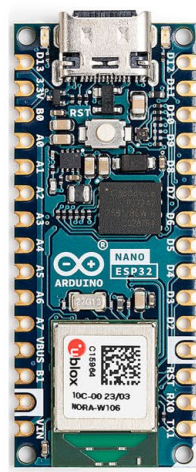
Seed Studio
ESP32-C3
\$4.99



Adafruit QT Py
ESP32-C3
\$9.95



Espressif ESP32-S3 Devkit
\$15



Arduino Nano ESP32-S2
\$20



Adafruit ESP32-S2 TFT
\$24.95



Sparkfun Thing+ ESP32
\$22.50



Sparkfun Thing+ ESP32-S2
\$21.50



Sparkfun Thing+ ESP32-S3
\$19.95



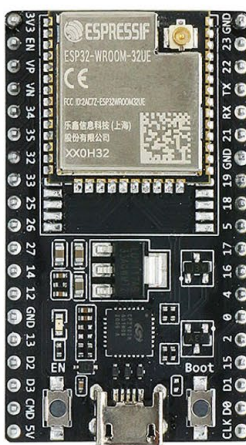
Wemos Lolin
ESP32-C3 Pico
\$5.46



Adafruit QT Py
ESP32-S3
\$12.50



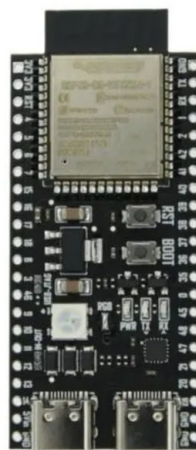
Sparkfun Pro Micro
ESP32-C3
\$9.95



Espressif ESP32 Devkit
\$10



Wemos Lolin ESP32-S3
\$7.09



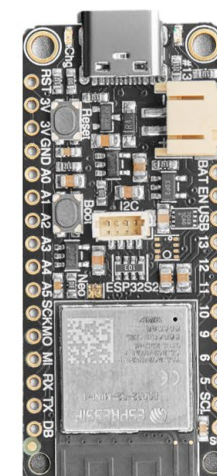
Generic ESP32-S3
\$0.99



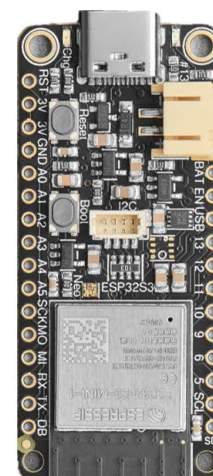
Lilygo T-Disp S3 AMOLED
\$33.98



Adafruit ESP32
\$19.95



Adafruit ESP32-S2
\$17.50

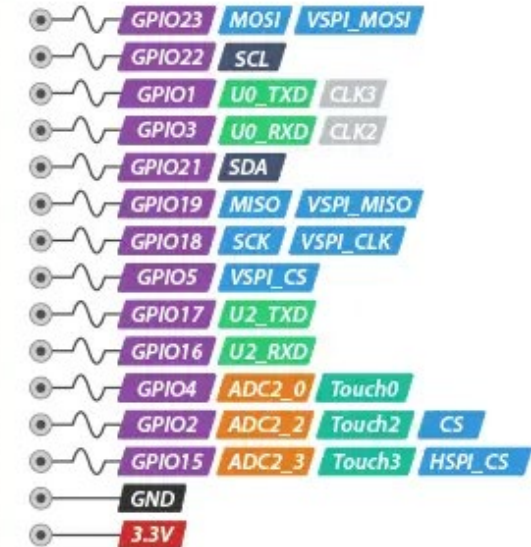
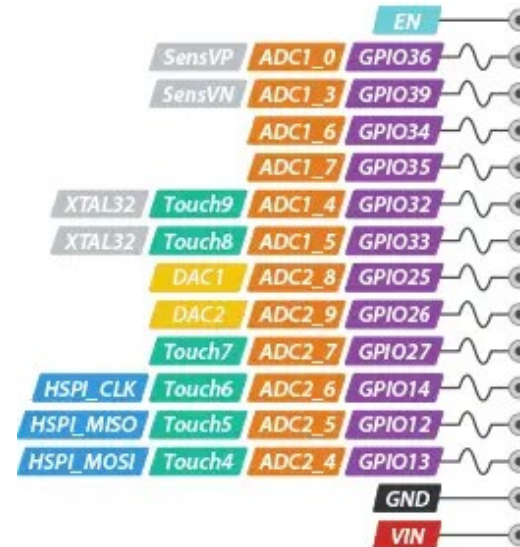


Adafruit ESP32-S3
\$17.50

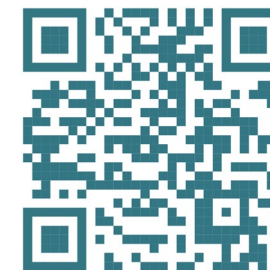


ESP32 karakteristike

- 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



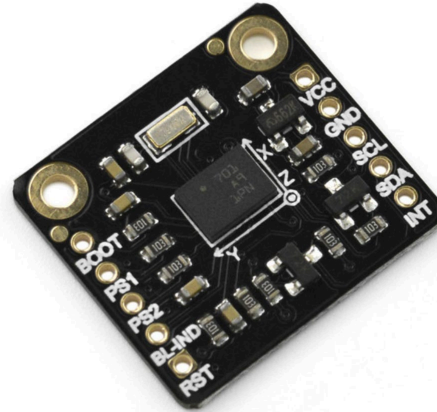
BNO055 senzor

- **BNO055** - senzor za apsolutnu orijentaciju koji kombinuje akcelerometar, žiroskop i magnetometar u jednom uređaju.
- Proizvođač: Bosh.
- Glavne karakteristike:
 - integrisani mikroprocesor za obradu podataka o orijentaciji,
 - očitava ubrzanje, ugaonu brzinu i magnetno polje,
 - I2C i UART interfejs za komunikaciju,
 - mala potrošnja.
- Primjena:
 - stabilizacija dronova,
 - kontrola pokreta u robotici,
 - sistem za orijentaciju u prenosivim uređajima, itd.



BNO055 senzor

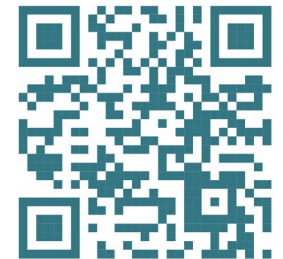
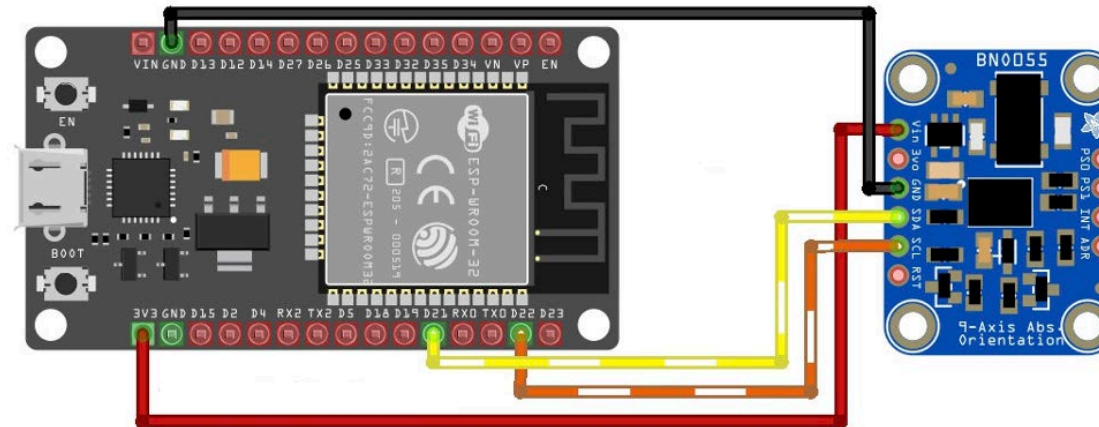
- Tehničke specifikacije:
 - napajanje: 3.3V - 5.5V;
 - merni opseg:
 - akcelerometar: $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$;
 - žiroskop: $\pm 125^\circ/s$, $\pm 250^\circ/s$, $\pm 500^\circ/s$, $\pm 1000^\circ/s$, $\pm 2000^\circ/s$;
 - magnetometar: $\pm 1300 \mu T$.
 - dimenzije: 2.cm x 2.1cm;
 - radna temperatura: $-40^\circ C$ do $+85^\circ C$;
 - I2C adresa: podrazumijevana 0X28.



PS1	PS2	Function
0	0	Standard/Fast 12C Interface
0	1	HID OVER I2C
1	0	UART Interface
1	1	Reserved



[BNO055 tutorial i datasheet](#)



[Arduino biblioteka](#)



BNO055 primjer

```
read_all_data.ino
1  #include "BNO055_support.h" //Contains the bridge code between the API and Arduino
2  #include <Wire.h>
3
4  //The device address is set to BNO055_I2C_ADDR2 in this example. You can change the
5  // /* bno055 I2C Address */
6  // #define BNO055_I2C_ADDR1          0x28
7  // #define BNO055_I2C_ADDR2          0x29
8  // #define BNO055_I2C_ADDR          BNO055_I2C_ADDR2
9
10 //Pin assignments as tested on the Arduino Due.
11 //Vdd,Vddio : 3.3V
12 //GND : GND
13 //SDA/SCL : SDA/SCL
14 //PS0/PS1 : GND/GND (I2C mode)
15
16 //This structure contains the details of the BNO055 device that is connected. (Updated)
17 struct bno055_t myBNO;
18 struct bno055_euler myEulerData; //Structure to hold the Euler data
19
20 unsigned long lastTime = 0;
21
22 void setup() //This code is executed once
23 {
24     //Initialize I2C communication
25     Wire.begin();
26
27     //Initialization of the BNO055
28     BNO_Init(&myBNO); //Assigning the structure to hold information about the device
29
30     //Configuration to NDoF mode
31     bno055_set_operation_mode(OPERATION_MODE_NDOF);
32
```

```
read_all_data.ino
30 //Configuration to NDoF mode
31 bno055_set_operation_mode(OPERATION_MODE_NDOF);
32
33 delay(1);
34
35 //Initialize the Serial Port to view information on the Serial Monitor
36 Serial.begin(115200);
37 }
38
39 void loop() //This code is looped forever
40 {
41     if ((millis() - lastTime) >= 100) //To stream at 10Hz without using additional timers
42     {
43         lastTime = millis();
44
45         bno055_read_euler_hrp(&myEulerData); //Update Euler data into the structure
46
47         Serial.print("Time Stamp: "); //To read out the Time Stamp
48         Serial.println(lastTime);
49
50         Serial.print("Heading(Yaw): "); //To read out the Heading (Yaw)
51         Serial.println(float(myEulerData.h) / 16.00); //Convert to degrees
52
53         Serial.print("Roll: "); //To read out the Roll
54         Serial.println(float(myEulerData.r) / 16.00); //Convert to degrees
55
56         Serial.print("Pitch: "); //To read out the Pitch
57         Serial.println(float(myEulerData.p) / 16.00); //Convert to degrees
58
59         Serial.println(); //Extra line to differentiate between packets
60     }
61 }
```



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.

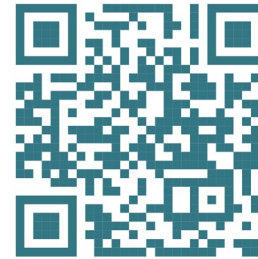


Karakteristike Tello drona:

- **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

```
read_all_data.ino
1  #include <Tello.h>
2
3  // WiFi network name and password:
4  const char * networkName = "TELLO-XXXXXX";//Replace with your Tello SSID
5  const char * networkPswd = "";
6
7  //Are we currently connected?
8  boolean connected = false;
9
10 Tello tello;
11
12 void setup()
13 {
14     Serial.begin(9600);
15     //Connect to the WiFi network
16     connectToWiFi(networkName, networkPswd);
17 }
18
19 void loop()
20 {
21     // put your main code here, to run repeatedly:
22     if(connected )
23     {
24         tello.takeoff();
25         delay(5000);
26         tello.up(30);
27         delay(2000);
28         tello.down(30);
29         delay(2000);
30         tello.right(30);
31         delay(2000);
32         tello.left(30);
33         delay(2000);
34         tello.land();
35         //you have 5 seconds to save your tello before it takes off again
36         delay(5000);
37
38         //do once and go into a while loop
39         while(1)
40         {
```

```
read_all_data.ino
41     delay(5000);
42     }
43 }
44 }
45
46 void connectToWiFi(const char * ssid, const char * pwd)
47 {
48     Serial.println("Connecting to WiFi network: " + String(ssid));
49
50     // delete old config
51     WiFi.disconnect(true);
52     //register event handler
53     WiFi.onEvent(WiFiEvent);
54
55     //Initiate connection
56     WiFi.begin(ssid, pwd);
57
58     Serial.println("Waiting for WIFI connection...");
59 }
60
61 //wifi event handler
62 void WiFiEvent(WiFiEvent_t event)
63 {
64     switch (event)
65     {
66         case SYSTEM_EVENT_STA_GOT_IP:
67             //When connected set
68             Serial.print("WiFi connected! IP address: ");
69             Serial.println(WiFi.localIP());
70             //initialise Tello after we are connected
71             tello.init();
72             connected = true;
73             break;
74
75         case SYSTEM_EVENT_STA_DISCONNECTED:
76             Serial.println("WiFi lost connection");
77             connected = false;
78             break;
79     }
80 }
```



Zaključak

- Arduino, ESP32, BNO055 senzor 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!



M O D R O N E

Shapes the future of
innovation