

---

## **ITTD - zbirka vaj**

Ta zbirka vaj je namenjena študentom Pedagoške fakultete,  
Univerze v Ljubljani.

dr. David Rihtaršič

2020-11-18



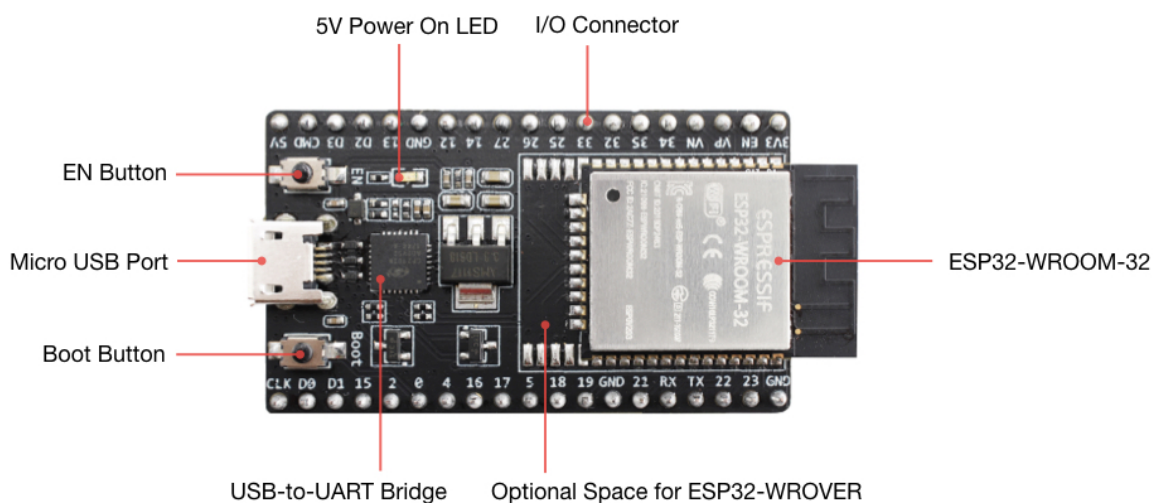
# Kazalo

<b>1</b>	<b>KRMILNIK ESP32-WROOM-32D</b>	<b>3</b>
1.1	OSNOVNE LASTNOSTI . . . . .	3
1.1.1	NALOGA: OSNOVE LASTNOSTI KRMILNIKA . . . . .	4
1.2	UPORABA KRMILNIKA ESP32 . . . . .	5
1.2.1	NALOGA: PRIPRAVA IDE ZA DELO Z ESP32 . . . . .	6
1.2.2	NALOGA: UPORABA KRMILNIKA ESP32 . . . . .	6
<b>2</b>	<b>MERJENJE TEMPERATURE</b>	<b>7</b>
2.1	Temperaturni senzor LM35 . . . . .	7
2.1.1	NALOGA: OSNOVNI PODATKI SENZORJA LM35 . . . . .	7
2.1.2	NALOGA: PREIZKUS SENZORJA . . . . .	7
2.1.3	NALOGA: OJAČANJE IZHODNE NAPETOSTI SENZORJA . . . . .	8
2.1.4	NALOGA: UMERITEV SENZORJA . . . . .	8
2.1.5	NALOGA: IZPIS TEMPERATURE . . . . .	9
2.1.6	NALOGA: NAPOVED NA PODLAGI 3-H IZMERJENIH TOČK . . . . .	10



# 1 KRMILNIK ESP32-WROOM-32D

ESP32\_DevKitC\_V4 krmilnik je majhna razvojna plošča na osnovi družine krmilnikov ESP32, ki jih proizvaja Espressif. Večina vhodno/izhodnih zatičev je razdeljena dve vrsti obeh straneh za lažje povezovanje modula z periferno elektroniko. Razvijalci lahko periferne naprave povežejo z žicami ali pa ESP32\_DevKitC\_V4 namestijo na razvojno ploščo. Ena od različic ESP32-DevKitC krmilnikov je na sl. 1.1.



**Slika 1.1:** Krmilnik ESP32-DevKitC\_v4 z modulom ESP32-WROOM-32

Več o tem krmilniku si lahko preberete na njihovi [spletni strani](#)

## 1.1 OSNOVNE LASTNOSTI

Pred samo uporabo se najprej seznanimo z glavnimi lastnostmi modula ESP32\_DevKitC\_V4. Tako bomo spoznali njegove možnosti uporabe in tudi omejitve. Glavni sestavni krmilnika je modul

ESP32\_WROOM\_32D, ki ga prikazuje sl. 1.2.



Slika 1.2: ESP32\_WROOM\_32D modul.

Na krmilniku ESP32\_DevKitC\_V4 so priključki razporejeni podobno, kot jih vidimo na sl. 1.2. Njihovo zaporedje je ohranjeno, le razporeditev je razvrščena v dve vrsti. Bolj natančna vezava pa je predstavljena v sami [el.-teh. shemi](#) krmilnika.

### 1.1.1 NALOGA: OSNOVE LASTNOSTI KRMILNIKA

Preglejte [dokumentacijo](#) modula ESP32\_WROOM\_32D in izpišite njegove osnovne lastnosti. Če katerih lastnosti ne najdete si lahko pomagate z dodatno literaturo mikrokrmilnika [ESP32\\_D0WD](#), ki je osrednja komponenta modula. Na spletu lahko najdete tudi druge [spletne strani](#) drugih uporabnikov, saj je krmilnik zaradi svojih lastnosti izredno popularen.

#### OSNOVNE SPLOŠNE LASTNOSTI MODULA ESP32\_WROOM\_32D:

Delovni takt:

Tipi brezžične komunikacije:

Protokoli ožičene komunikacije:

Vključujoči senzorji:

Vključujoče krmilne enote:

Št. digitalnih GPIO priključkov:

Št. ADC enot:

Št. ADC priključkov:

ADC resolucija:

Št. DAC priključkov:

**OSNOVNE ELEKTRIČNE LASTNOSTI MODULA ESP32\_WROOM\_32D:**

Napajalna napetost (max):

Delovni tok:

Vhodni nap. pot. na GPIO prik. (max):

Izhodni nap. pot. na GPIO prik. (min):

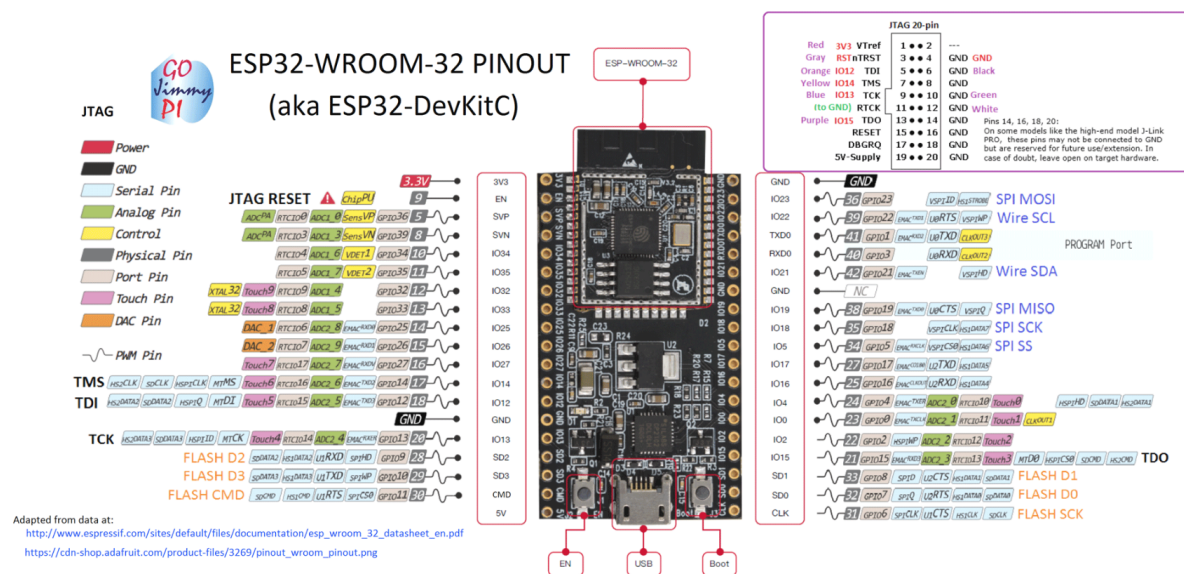
Izhodni tok GPIO prik. (izvor/ponor):

Upornost upora proti nap.:

Upornost upora proti GND:

## 1.2 UPORABA KRMILNIKA ESP32

Enostaven in nazoren pregled priključkov nudi sl. 1.3, katero boste morali dosledno upoštevati pri vaši vezavi. Ker so največji dopustni napetostni potenciali omejeni na 3.3V, je verjetnost uničenja modula pri vezavi na 5V zelo verjetna. D Upoštevajte, da je uporaba priključkov GPIO6 .. GPIO11, saj so ti priključki povezani



Slika 1.3: Pregled priključkov.

### 1.2.1 NALOGA: PRIPRAVA IDE ZA DELO Z ESP32

Krmilnik povežite z računalnikom in programsko okolje Arduino IDE (ali podobno) nastavite za delo s tem krmilnikom. Pri nastavitvah si pomagajte s spletnimi napotki, kot je naprimer [tale](#). Pomembnejše nastavitve strnite v nekaj točkah.

### 1.2.2 NALOGA: UPORABA KRMILNIKA ESP32

Za krmilnik ESP32-WROOM napišite enostaven program s katerim boste predstavili uporabo osnovnih I/O funkcij kot so:

1. Krmiljenje LED z digitalnim izhodom.
2. Odčitavanje digitalnega vhoda na katerega smo vezali tipko.
3. Odčitavanje analognega vhoda na katerega smo vezali potenciometer.
4. Pošiljanje podatkov z uporabo serijske komunikacije UART.

Vse 4 prej naštetih funkcionalnosti lahko strnite v eno nalogo in jo ustrezno dokumentirate (el.-teh. shema vezja in program.)



## 2 MERJENJE TEMPERATURE

### 2.1 Temperaturni senzor LM35

Temperaturo bomo merili s senzorjem temperature LM35. Poglejmo si [dokumentacijo](#) tega integriranega vezja in izpišite nekaj osnovnih podatkov.

#### 2.1.1 NALOGA: OSNOVNI PODATKI SENZORJA LM35

Odziv senzorja  $K_{LM35} =$

Točnost senzorja  $\Delta T =$

Temp. del. območje =

#### ELEKTRIČNE SPECIFIKACIJE:

Napajalna napetost  $U_s =$

Izh. not. upornost.  $R_{OUT} =$

Najv. izh. tok  $I_{OUT-MAX} =$

#### 2.1.2 NALOGA: PREIZKUS SENZORJA

1. Temperaturni senzor LM35 priključite na napajanje in ga postavite v okolje s konstantno temperaturo (posoda z vodo cca 35°C in preverite temperaturo z dodatnim termometrom.
2. Izhodni priključek povežite na vhod krmilnika ESP32\_DevKitC in preverite napetostni odziv.
3. Krmilnik ESP32\_DevKitC naj izmerjene podatke prikazuje v napetosti ter jih po serijski komunikaciji UART pošilja računalniku.

Z ustrezno dokumentacijo, ki naj vključuje:

1. fotografije poskusa,
2. elek.-teh. shema,
3. program krmilnika,

4. izpis vrednosti ter vaše ugotovitve

opišite preskus temperaturnega senzorja.

Programske vrstice za povprečno vrednost meritve:

```
1  int ADC = 0;
2  for (int i = 0; i < 128; i++) {
3      ADC += analogRead(A7);
4      delay(1);
5  }
6  float avg_ADC = float(ADC/128.0);
```

### 2.1.3 NALOGA: OJAČANJE IZHODNE NAPETOSTI SENZORJA

Z ustreznimi elektronskimi sistemi (npr.: ojačevalnik napetosti) prilagodite izhodni napetostni potencial temperaturnega senzorja tako, da bo območje izhodne napetosti  $U_{OUT} = [0..3, 0]V$  ustrezalo temperaturnemu območje  $T_{MIN-MAX} = [0..40]^{\circ}C$ .

Vhod krmilika tudi primerno zavarujte proti napetostim večjim od 3.3V.

Dokumentacija naloge naj vsebuje:

1. el.-teh. shemo in
2. rezultate vsaj ene meritve z ugotovitvami (kako vam je transformacija uspela).

### 2.1.4 NALOGA: UMERITEV SENZORJA

Napravite umeritev senzorja po celem temperaturnem območju  $T_{MIN-MAX} = [0..40]^{\circ}C$ . Dokumentacija vaje naj vsebuje:

1. tabelo meritev (temperatura | ADC\_vrednost),
2. graf umeritve  $T(ADC\_vrednost)$ ,
3. dodan ustrezen trend funkcije (verjetno nek polinom n-te stpnje) in
4. njene enačbe.

V kolikor želite primeren izpis podatkov tabele sestaviti že na krmilniški strani, lahko podatek o izmerjeni temperaturi z referenčnim termometrom posredujete krmilniku po serijski komunikaciji. Za sprejemanje pa uporabite dodatno funkcijo `serialEvent()` z nasladnim programom:

```

1  String input_temp = "";
2  void serialEvent() {
3      while (Serial.available()) {
4          char inChar = (char)Serial.read(); // get the new byte:
5          if (inChar == '\n') {           // if ENTER was pressed
6              int ADC = analogRead(A0);
7              Serial.print(input_temp);Serial.print("\t");Serial.println(ADC);
8              //Serial.printf("%d\t%d",input_temperature,ADC);
9              input_temp="";
10         } else {
11             input_temp+= inChar;        // add it to the input_temperature:
12         }
13     }
14 }

```

### 2.1.5 NALOGA: IZPIS TEMPERATURE

S pomočjo enačbe umeritvene krivulje napišite program za krmilnik tako, da bo ustrezno podajal že izračunan podatek o temperaturi. Izračunan podatek primerjajte s temperaturo, ki jo izmerite z referenčnim termometrom. V poročilo vključite:

1. fotografijo preskusa meritve,
2. program krmilnika,
3. izpis vrednosti in
4. vaš komentar meritve.

Pri preračunu vrednosti polinoma 6. stopnje po 2.1

$$y = k_5x^5 + k_4x^4 + k_3x^3 + k_2x^2 + k_1x^1 + k_0x^0 \quad (2.1)$$

si lahko pomagata z naslednjimi programskimi vrsticami.

```

1  float k[6] = { -74.9, 530E-3, -1.68E-3, 3.25E-6, -3.12E-9, 1.22E-12};
2  float y = 0;
3  float ADC = analogRead(A0);
4  for (int i = 0; i <= 5; i++) {
5      y += k[i] * pow(ADC, i);
6  }

```

**2.1.6 NALOGA: NAPOVED NA PODLAGI 3-H IZMERJENIH TOČK**

Na podlagi eksponentnega časovnega poteka segrevanja temperaturnega senzorja, ki ga podaja 2.2 lahko z meritvijo le treh meritev določimo končno temperaturo. Za ta preračun uporabite 2.3, ki predvideva, da sta časa med posameznimi meritvami enaki.

Napišite tak program, ki bo končno temperaturo ocenil na podlagi časovnega odziva. Meritev naj se začne s pritiskom na tipko in uporabniku nudi informacijo o časovnem poteku meritve (pričetek merjenja, vmesni čas, konec meritve). V poročilo vključite:

1. fotografijo preskusa meritve,
2. program krmilnika,
3. izpis meritve in uporabniških navodil ter
4. vaš komentar.

$$T(t) = T_k + (T_z - T_k)e^{\frac{t}{\tau}} \quad (2.2)$$

$$T_k = T_1 - \frac{(T_1 - T_2)^2}{T_1 - 2T_2 + T_3} \quad (2.3)$$