



SREDNJA ŠOLA SLOVENSKA BISTRICA, SLOVENSKA BISTRICA

AVTONOMNA VREMENSKA POSTAJA ESP32

Aplikativni inovacijski predlogi in projekti

Raziskovalna naloga

Avtor: Gašper Koklič, 3. a

Mentor: Aljaž Galun, prof.

Somentor: mag. Marko Žigart, prof.

SLOVENSKA BISTRICA, FEBRUAR 2022

Koklič, Gašper. Avtonomna vremenska postaja ESP32. Srednja šola Slovenska Bistrica, 2022.

ZAHVALE

Velika zahvala gospodu profesorju Aljažu Galunu in gospodu profesorju mag. Marku Žigartu za pomoč pri ustvarjanju projektne naloge ter za podporo in pomoč pri razvoju mojega projekta.

Velika zahvala tudi Srednji šoli Slovenska Bistrica za denarno podporo pri nabavi potrebnih komponent.

Ogromna zahvala tudi gospe profesorici Lidiji Ličen za jezikovni pregled naloge.

KAZALO VSEBINE

AVTONOMNA VREMENSKA POSTAJA ESP32	1
ZAHVALE	3
KAZALO VSEBINE.....	4
KAZALO SLIK	5
KAZALO GRAFOV	5
POVZETEK	6
UVOD	7
VSEBINSKI DEL	8
ESP32 WROOM-32.....	8
BME680	12
Fotovoltaične sončne celice in koncentrirane sončne elektrarne	13
SESTAVLJANJE.....	14
PROGRAMIRANJE	16
Programiranje senzorja BME680.....	16
Prenos podatkov na splet	18
DELOVANJE	23
ZAKLJUČEK.....	25
VIRI BESEDILA.....	26
VIRI SLIK.....	26

KAZALO SLIK

Slika 1: Razporeditev zatičev ESP-WROOM-32 (ESP32-WROOM-32, 2021)	9
Slika 2: Shematika periferije ESP32-WROOM-32 (ESP32-WROOM-32, 2021)...	10
Slika 3: Dimenzije ESP-32-WROOM-32 (ESP32-WROOM-32, 2021).....	11
Slika 4: ESP32-WROOM-32 (Espressif, 2022)	11
Slika 5: BME680 (BME680, 2022).....	12
Slika 6: BME680 CJMCU-680	12
Slika 7: Fotovoltaična sončna celica (Gašper Koklič, 2022).....	13
Slika 8: Razstavljena vremenska postaja (Gašper Koklič, 2022).....	14
Slika 9: Shematika povezav med ESP32 in BME680 (Gašper Koklič, 2022).....	15
Slika 10: Zaprta vremenska postaja (Gašper Koklič, 2022)	15
Slika 10: Zaprta Vremenska postaja (Gašper Koklič, 2022).....	15
Slika 11: Izgled spletne strani (Gašper Koklič, 2022)	23

KAZALO GRAFOV

Graf 1: Časovni potek merjenja temperature 23. 2. 2022	24
Graf 2 Časovni potek merjenja tlaka 23. 2. 2022	24
Graf 3: Časovni potek merjenja vlage 23. 2. 2022	24

POVZETEK

Ustvaril sem vremensko postajo, sposobno avtonomnega delovanja. Vremenska postaja deluje na podlagi razvojnega ogrodja ESP32, ki ima vgrajen modul Wi-Fi. Za merilne funkcije uporabljam Boschov senzor BME680, ki omogoča natančne meritve v manjšem ohišju. S senzorjem se merijo podatki o temperaturi, vlagi in tlaku, ki jih z modulom Wi-Fi na ESP32 pošljemo na server, katerega gostitelj je isti ESP32. Na takšen način so meritve senzorja lahko berljive na spletni strani, ki se nahaja na serverju. Avtonomno delovanje vremenske postaje omogoča sončna celica z vgrajeno baterijo, ki lahko poganja vremensko postajo dan in noč. Vremenska postaja lahko deluje tudi v oblačnem vremenu, ko ni dovolj proizvedene sončne energije, saj lahko baterija v sončni celici poganja ESP32 več dni zapored.

Ključne besede: ESP32, vremenska postaja, BME680, avtonomno delovanje.

UVOD

Vremenske postaje so dandanes zelo kompleksne in zahtevne naprave. Če hočemo kupiti kvalitetno vremensko postajo, nas bo stala sto evrov in več. Za namene projektne naloge sem se odločil ustvariti kvalitetno avtonomno vremensko postajo, ki je zmožna izvajati meritve na terenu in jo je mogoče uporabljati v nadaljnje raziskovalne namene s področja meteorologije in geografije. Za izgradnjo vremenske postaje sem uporabil enostavno dobavljive poceni dele, ki jih je preprosto zamenjati, da bi tako omogočil trajnostno uporabo vremenske postaje. Če se katera izmed komponent v komercialnih vremenskih postajah pokvari, se je ne da enostavno zamenjati, temveč je vremensko postajo treba peljati na servis. Glavni cilj naloge je bil ustvariti avtonomno vremensko postajo, sposobno merjenja vlage, temperature in tlaka ter pošiljanja teh podatkov na splet.

Komponente za vremensko postajo sem nabavil na spletnih straneh 3Dsvet.eu, Arrisi.eu in Galagomarket.com ter v trgovinah Nano elektronika in ETS Pregl.

Za programiranje vremenske postaje sem uporabil razvojno okolje Arduino IDE, ki temelji na programskih jezikih C in C++. Za testiranje delovanja kode in za vizualizacijo povezav komponent vremenske postaje sem uporabil program Fritzing.

VSEBINSKI DEL

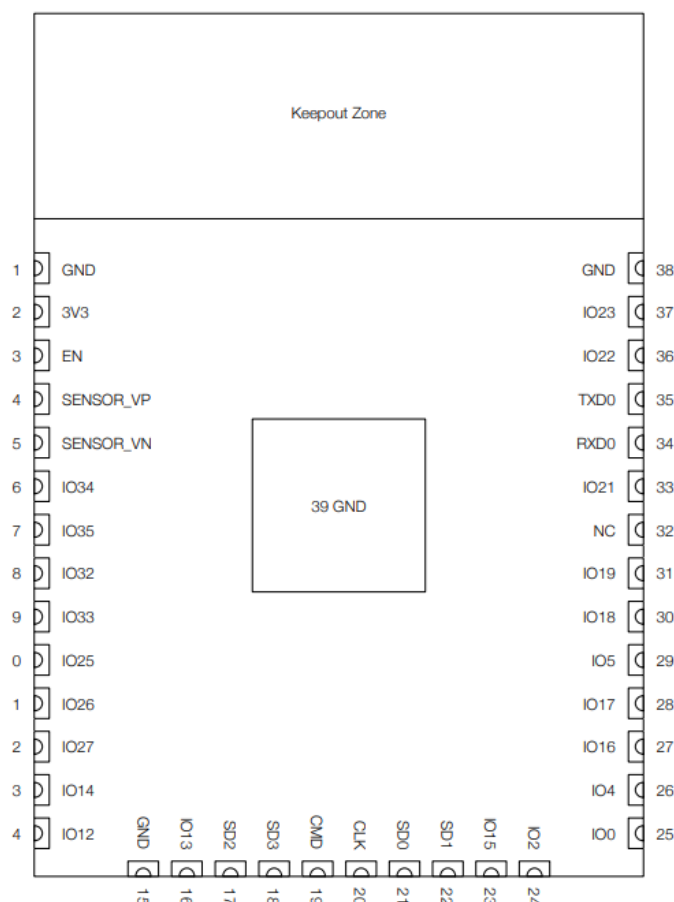
ESP32 WROOM-32

ESP32-WROOM-32 je MCU-modul z vgrajenimi sposobnostmi Wi-Fi, Bluetooth in Bluetooth low energy. Deluje na delovni napetosti med 3,0 V–3,6 V in toku ~80mA. MCU ali »Microcontroller unit« je kompaktno integrirano vezje, narejeno za določeno delovanje znotraj vgrajenega sistema. MCU je po navadi sestavljen iz procesorja, delovnega spomina in vhodnih/izhodnih (I/O) naprav. MCU v vgrajenem sistemu nadzoruje enojno nalogo v napravi, v katero je vgrajen. To nalogo opravi z interpretacijo podatkov, ki jih sprejme iz I/O-naprav, znotraj osrednjega procesorja. Začasne podatke shrani v podatkovni spomin, kjer jih procesor doseže in jih nato s pomočjo navodil, shranjenih v programskem spominu, dešifrira in aplicira prihajajoče podatke. Po tem MCU uporabi I/O-naprave za izvajanje primerne odgovora. (ESP32-WROOM-32, 2021) (Lutkevich, B., 2022)

Kakor sem omenil, je MCU sestavljen iz procesorja, spomina in I/O-naprav. ESP32-WROOM-32 ima vgrajen procesor ESP32-D0WDQ6. Ta vsebuje dve jedri, ki ju lahko kontroliramo posebej. Delovni takt procesorja je nastavljiv in obsega frekvence med 80 in 240 MHz. Poleg glavnega procesorja je vgrajen še nizko poraben koprocesor, namenjen opravljanju nalog, ki ne zahtevajo veliko moči. Spomin je razdeljen na pet delov. Prvi del je bralni pomnilnik (ROM), ki se uporablja pri zagonu in za osnovne funkcionalnosti procesorja. ROM obsega 448 kB. Drugi del je statični delovni spomin, ki je rezerviran za podatke in navodila. Obsega 520 kB. Tretji del je hiter delovni spomin (RTC FAST) v uri procesorja (RTC), ki se uporabi, ko sistem izstopi iz nizko porabnega stanja (deep-sleep mode) s procesom RTC boot. Ta spomin obsega 8 kB. Četrti del je počasen delovni spomin (RTC SLOW), ki se uporablja, ko naprava preide v nizko porabno stanje (temu spominu lahko dostopa le koprocesor). Zadnji del obsega 1 kB eFuse, pri čemer je 256 bitov namenjenih sistemu, 768 bitov pa je rezerviranih za razne aplikacije. Poleg spomina, vgrajenega v sam procesor, ima MCU vgrajene še 4MB spomina SPI FLASH, ki jih procesor uporablja za trajno shranjevanje programov, ki so nato prebrani in zapisani

v delovni pomnilnik. (ESP32-WROOM-32, 2021)

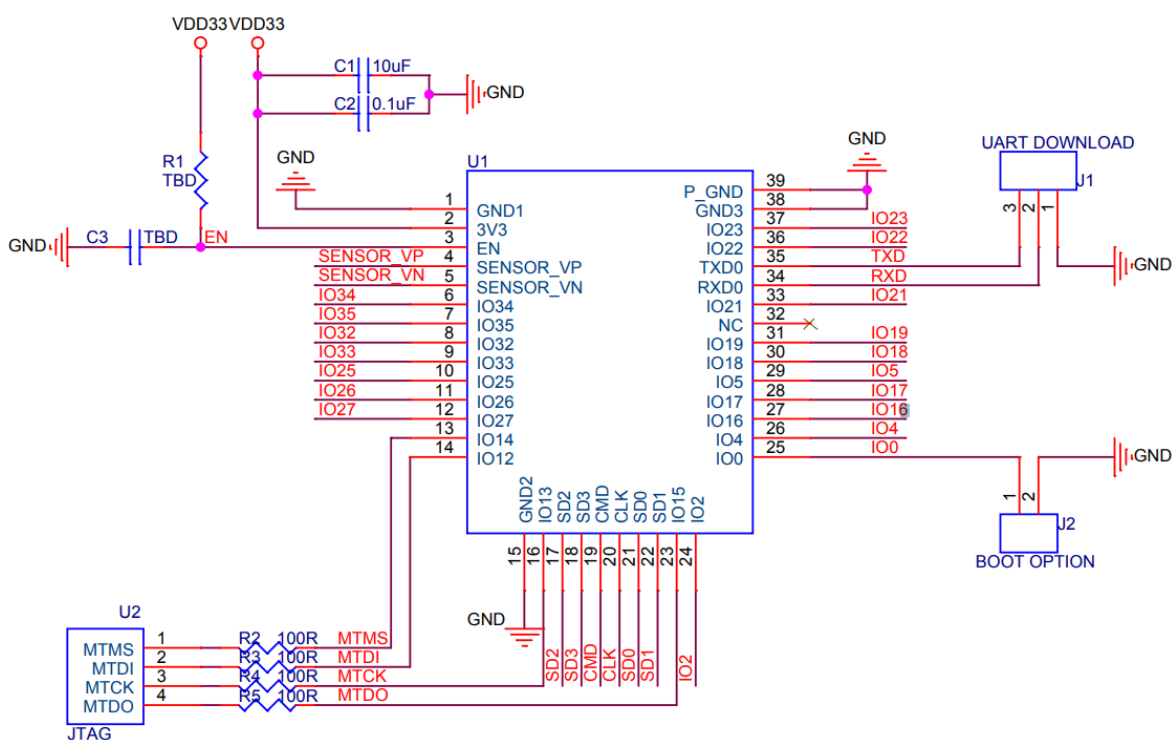
Za I/O-funkcije ima ESP32-WROOM-32 velik nabor I/O-vrat za podatkovno branje



Slika 1: Razporeditev zatičev ESP-WROOM-32

(ESP32-WROOM-32, 2021)

senzorjev in razne druge periferije. Za te I/O-funkcije uporablja veliko vmesnikov, in sicer ima vgrajeno funkcionalnost s SD-karticama, UART, SPI, SDIO, I²C, I²S, motorno modulacijo in LED-modulacijo širine impulza (PWM), IR, števec impulzov, splošna I/O-vrata (GPIO), kapacitivne senzorje na dotik, analogno-digitalne pretvornike (ADC), digitalne avdiopretvornike (DAC), dvožične avtomobilske vmesnike (TWAI®) in naprave, kompatibilne s specifikacijo ISO11898-1 (CAN-specifikacija 2.0). Poleg vseh teh funkcionalnosti ima MCU vgrajen še senzor hall za magnetna polja. (ESP32-WROOM-32, 2021)



Slika 2: Shematika periferije ESP32-WROOM-32 (ESP32-WROOM-32, 2021)

Za brezžično funkcionalnost ima ESP32-WROOM-32 vgrajena modula Wi-Fi (certifikat Wi-Fi Alliance) in Bluetooth (certifikat BQB). Modul Wi-Fi deluje pri frekvencah 2.4–2.5 GHz s protokoli 802.11b, 802.11g in 802.11n ter podpira agregaciji A-MPDU in A-MSDU in uporablja 0.4 μ s zaščitni interval med transmisijami. Protokol 802.11n doseže hitrosti do 150 Mbps.

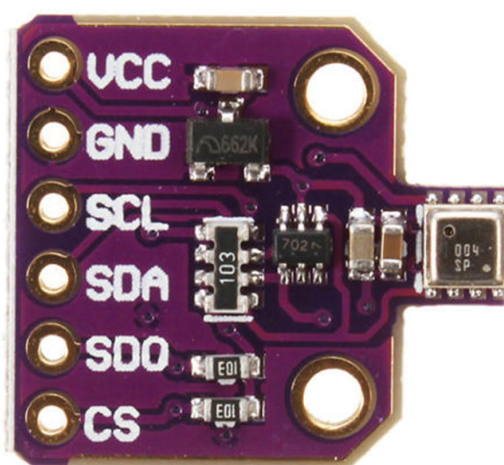
Modul Bluetooth deluje na protokolu Bluetooth v4.2 BR/EDR in specifikacijah BLE. Radio je sestavljen iz sprejemnika NZIF z občutljivostjo -97 dBm in treh razredov (1, 2, 3) oddajnikov. Modul Bluetooth uporablja prilagodljivi frekvenčno skakalni razpršenostni spekter (AFH), ki omogoča, da se radio izogne natrpanim frekvencam. Radio uporablja za prenašanje avdiosignalov neprekinjeno spremenljiv naklonski delta modulator (CVSD) in SBC-kodek, ki ga uporablja grupa Bluetooth za posebne interese (SIG) za kodiranje digitalnega avdia. (ESP32-WROOM-32, 2021)

BME680

BME680 je digitalni nizko porabni senzor za pline, tlak, temperaturo in vlago. Proizvaja ga Bosch. Uporablja se v merilnikih kvalitete zraka in tudi v meritvenih aplikacijah. Deluje na podlagi digitalnih vmesnikov I²C in SPI. Njegova delovna napetost je med 1.71 V in 3.6 V, skupna poraba vgrajenih senzorjev pa je 0.19–12 mA. Senzorji delujejo v razponu od –40 do +85 °C, 0–100 % vlage in 300–1100 hPa. Natančnost senzorjev je $\pm 3\%$ r. h., ± 0.6 hPa in ± 0.5 °C. (BME680, 2021)



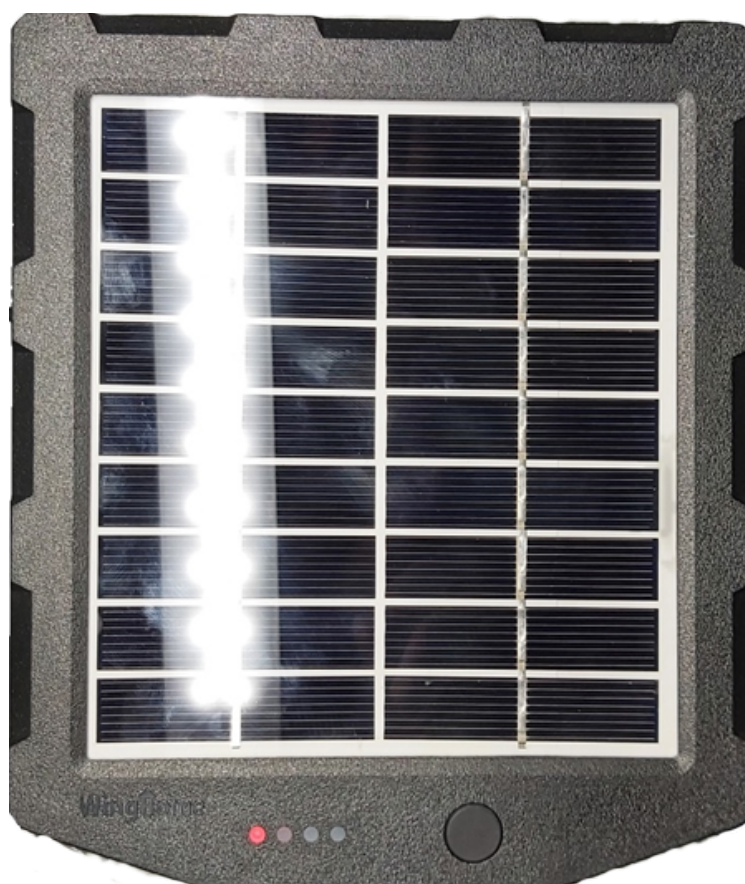
Slika 5: BME680 (BME680, 2022)



Slika 6: BME680 CJMCU-680
(BME680 CJMCU-680, 2022)

Fotovoltaične sončne celice in koncentrirane sončne elektrarne

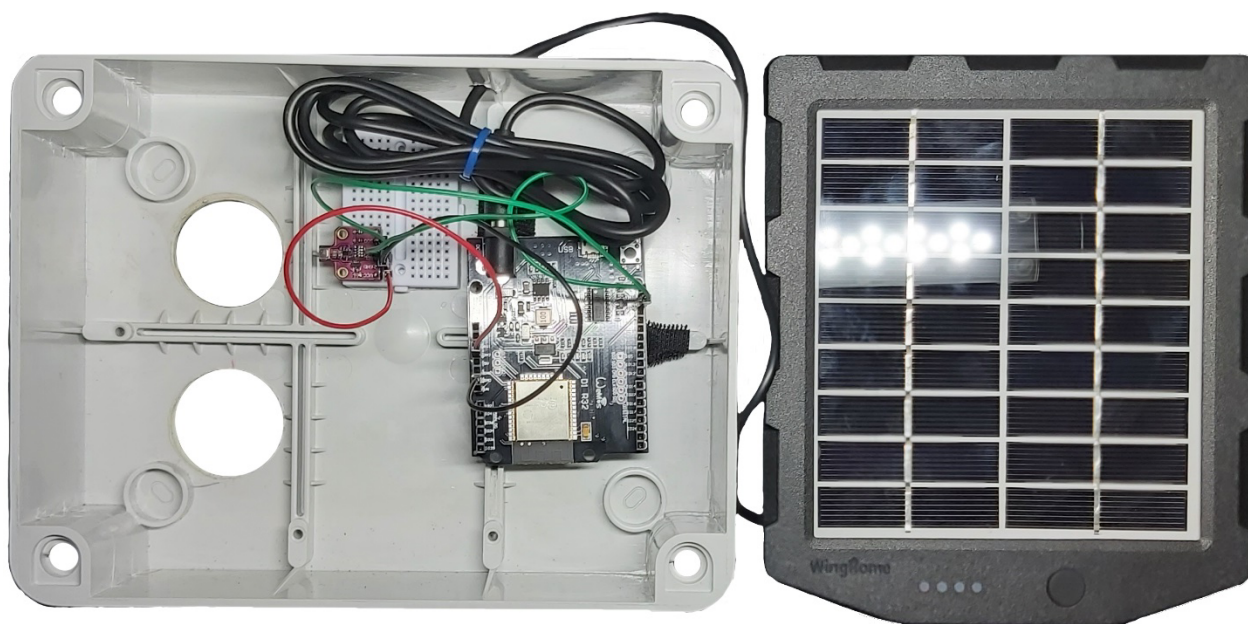
Fotovoltaične sončne celice so nizi celic iz materialov, ki pretvarjajo svetlobno valovanje v enosmerno napetost (DC). Materiali, ki se danes najpogosteje uporabljajo, so amorfni silikoni, polikristalni silikoni, mikro kristalni silikoni, kadmijev telurid in bakrov indijev selenid/sulfid. Materiali so dopirani zato, da se poveča število pozitivnih ali negativnih prenašalcev naboja. Ti pozitivni in negativni polprevodniki se povežejo v stičišča, ki omogočajo proizvodnjo električne energije ob prisotnosti svetlobe. Učinkovitost fotovoltaičnih celic se zmanjša, ko temperatura na celici prekorači 45 °C. Fotovoltaične celice lahko namestimo na strehe ali povežemo v sočne elektrarne. (Jacobson, J. M., 2008)



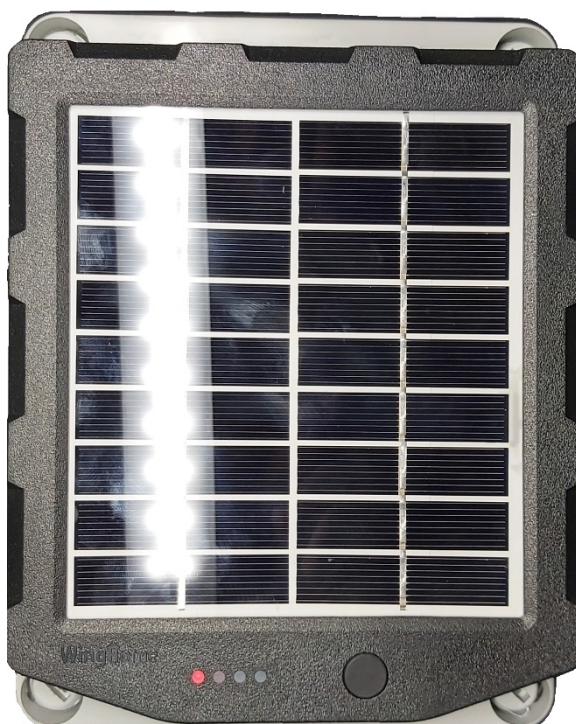
Slika 7: Fotovoltaična sončna celica (Gašper Koklič, 2022)

SESTAVLJANJE

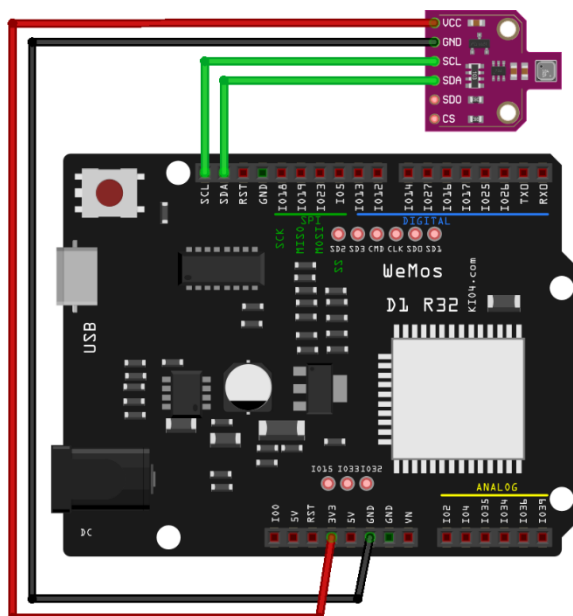
V osredju vremenske postaje se nahaja Wemos D1 R32 ESP-32. To razvojno ogrodje temelji na ESP32-WROOM-32, vgrajenem v ploščo iste velikosti kot Arduino Uno. Iz D1 R32 ESP-32 sta napeljana dva zelena kabla iz SDA- in SCL-zatičev v prototipno ploščo, kjer se tvori povezava z istoimenovanimi zatiči na senzor BME680 CJMCU-680. Ta dva kabla omogočata I²C-komunikacijo. Poleg obeh senzor potrebuje še napajanje in ozemljitev. Napajanje dovaja rdeči kabel, ki povezuje zatič 3V3 v ESP32 z zatičem VCC v BME680. Ozemljitev je povezana s črnim kablom iz ESP32 v BME680. Napajanje dovaja sončna celica z vgrajeno baterijo model WH100 proizvajalca WingHome.



Slika 8: Razstavljena vremenska postaja (Gašper Koklič, 2022)



Slika 10: Zaprta vremenska postaja
(Gašper Koklič, 2022)



Slika 9: Shematika povezav med ESP32 in BME680
(Gašper Koklič, 2022)

PROGRAMIRANJE

Programiranje senzorja BME680

Informacije s senzorja BME680 se prešajo na ESP32 s pomočjo digitalnih vmesnikov SPI in I²C. V mojem primeru sem uporabil vmesnik I²C, ki deluje na podlagi dveh zatičev na ESP32, in sicer SDA («serial data» oziroma serijski podatki) in SCL («serial clock» oziroma serijska ura). Zatiča SDA in SCL na BME680 in ESP32 sem povezal. V tem odnosu nastopa ESP32 v mojem primeru kot nadrejeni in BME680 kot podrejeni.

Za interpretacijo podatkov sem uporabil programske knjižnice Arduino IDE in knjižnici »Adafruit BME680 Library« in »Adafruit Unified Sensor«.

Za začetek je treba določiti, katere knjižnice bodo uporabljene.

```
#include <Wire.h> (predložena knjižnica za uporabo I2C)
#include <Adafruit_Sensor.h> (knjižnica »Adafruit Unified Sensor«)
#include "Adafruit_BME680.h" (knjižnica »Adafruit BME680 Library«)
```

Potem ko določimo knjižnice, moramo določiti predmet Adafruit_BME680, poimenovan bme, na I²C-zatičih SCL in SDA. To koristi v nadaljevanju, saj bomo imeli zaradi uporabe krajšave bme namesto Adafruit_BME680 bolj strnjen program. To dosežemo z:

```
Adafruit_BME680 bme;
```

Znotraj void.setup(), ki ponovi zapisana navodila enkrat ob zagonu, bomo sprva določili začetek serijske transmisije na serijski monitor s hitrostjo prenosa 115200 baud, kar je 115200 bitov na sekundo.

```
void setup() {
    Serial.begin(115200);
```


Potem ko pričnemo serijsko komunikacijo, moramo aktivirati senzor BME680.

```
if (!bme.begin()) { (če bme.begin() ne (!) poteče natisni spodnje)
Serial.println(F("Could not find a valid BME680 sensor, check
wiring!"));
```

```
    while (1);
}
```

bme.begin tvori true/false izraz oz 1 ali 0, dokler bme.begin() ne poteče se ta if zanka ne konča, saj če senzor ne deluje se tvori vrednost 1 zaradi !, ki določa nasprotje bme.begin(). Torej če bme.begin ne poteče se tvori vrednost 0, nasprotje je

Ko smo senzor že aktivirali, moramo določiti še stopnjo prekomernega vzorčenja. S tem določimo resolucijo naših neobdelanih meritev. V tem primeru smo uporabili privzete parametre.

```
bme.setTemperatureOversampling(BME680_OS_8X);
bme.setHumidityOversampling(BME680_OS_2X);
bme.setPressureOversampling(BME680_OS_4X);
```

S tem smo zaključili void.setup() in zdaj preidemo v void.loop(), ki neprestano ponavlja komande znotraj zanke. Znotraj zanke bomo brali podatke s senzorja, kar pričnemo tako, da aktiviramo senzor z bme.beginReading():

```
unsigned long endTime = bme.beginReading();
if (endTime == 0) {
    Serial.println(F("Failed to begin reading :("));
    return;
}
```

Meritev končamo z `bme.endReading()`:

```
if (!bme.endReading()) {  
    Serial.println(F("Failed to complete reading :("));  
    return;  
}
```

Ob koncu branja podatkov senzorja nam ostanejo spremenljivke: `bme.temperature`, `bme.pressure` in `bme.humidity`. `Bme.temperature` prikaže meritev toplote v °C brez dodatne obdelave. `Bme.pressure` je treba deliti s 100, da dobimo rezultat v hPa. `Bme.humidity` prikaže zračno vlago v odstotkih. Navedene spremenljivke bomo v nadaljevanju prenesli na spletno stran HTML.

(ESP32: BME680 Environmental Sensor using Arduino IDE, 2022)

Prenos podatkov na splet

Podatke senzorja BME680 se prenese na splet preko vgrajenega modula Wi-Fi na ESP32. ESP32 Wi-Fi-modul lahko funkcionira različno: kot odjemalec serverja, preko povezave LAN lahko prenaša na drugi ESP32 in v našem primeru funkcionira kot lasten spletni server, na katerega lahko pošilja podatke. S to funkcijo je ESP32 povezan preko Wi-Fi-ja na splet, kjer ustvari lasten server HTTP z naslovom, enakim njegovemu IP-naslovu.

Za to funkcionalnost se uporabita dve osnovni programski knjižnici za ESP32. To sta »Wifi.h« in »WebServer.h«. Sprva ju je treba določiti v kodi, določiti pa je treba tudi spremenljivke, ki jih bomo uporabljali:

```
#include <WiFi.h>  
#include <WebServer.h>  
Float temperature, humidity, pressure, p1;
```

Koklič, Gašper. Avtonomna vremenska postaja ESP32. Srednja šola Slovenska Bistrica, 2022.

Preden lahko delamo z Wi-Fi-funkcionalnostjo ESP32, moramo določiti spremenljivki za ime in geslo našega Wi-Fi-omrežja ter določiti vrata strežnika:

```
const char* ssid = »YourNetworkName«; (ime omrežja)
const char* password = »YourPassword«; (geslo omrežja)
WebServer server(80); (določanje vrat strežnika)
```

Potem ko smo določili spremenljivki za ime strežnika in njegovo geslo, lahko mnogo lažje izvedemo kodo za vpis v omrežje, in sicer znotraj void.setup(), ki ponovi kodo znotraj zanke enkrat ob zagonu, aktiviramo povezavo z Wi-Fi-omrežjem:

```
WiFi.begin(ssid, password); (zagon Wi-Fi funkcij)
  while (WiFi.status() != WL_CONNECTED){ (dokler se ne povežeš
ponavljaj zanko)
  delay(1000);
  Serial.print(».«);}
  (vsako sekundo po začetku Wi-Fi komunikacij brez povezave natisni
».«, to se ponavlja dokler se ESP32 ne poveže s Wi-Fi omrežjem)
  Serial.println(»WiFi connected..!«); (ko se povežeš natisni to)
  Serial.print(»Got IP: »); Serial.println(WiFi.localIP());
  (ko se povežeš natisni v serijski monitor svoj lokalni IP naslov)
```

Ko smo se povezali z omrežjem Wi-Fi, lahko začnemo s prenašanjem podatkov na splet. Sprva je treba določiti zanko, ki poteče, ko se stranka poveže na URL-serverja. Če ne gre za pravi naslov, nam druga zanka narekuje, da je prišlo do napake:

```
server.on(»/«, handle_OnConnect); (če se stranka serverja poveže na
veljaven URL, potem izvedi zanko v oklepaju za vejico)
server.onNotFound(handle_NotFound); (če se stranka serverja poveže
na neveljaven URL, potem izvedi zanko v oklepaju)
```

Za konec void.setup() zanke moramo zagnati server:

```
server.begin(); (zagon serverja)
Serial.println(»HTTP server started«);
```

Znotraj zanke void.loop() moramo določiti komando, ki bo omogočala serverju, da obdela zahteve HTTP:

```
server.handleClient();
```

Prej smo določili, katere zanke se izvedejo, ko server dobi veljavno in neveljavno zahtevo HTTP. Zdaj jih je treba še določiti:

```
void handle_OnConnect() { (zanka v primeru veljavne URL zahteve)
temperature = bme.temperature - 4.5;
p1 = bme.pressure / 100;
pressure = p1 + 36;
humidity = bme.humidity + 11.5;
(številke ki so prištete vrednosti so popravki za napake v
meritvah mojega specifičnega senzorja)
server.send(200,"text/html",SendHTML(temperature,humidity,pressure
));
(200 je HTML koda za OK, v narekovajih določimo tip podatkov, ki jih
pošiljamo (v našem primeru besedilo) in končno za vejico določimo
katere spremenljivke bomo pošiljali)
}
void handle_NotFound(){ (zanka v primeru neveljavne URL zahteve)
server.send(404, "text/plain", "Not found"); (404 je HTML koda, ki
nam pove, da URL, ki ga iščemo, ne obstaja na serverju)
}
```

Koklič, Gašper. Avtonomna vremenska postaja ESP32. Srednja šola Slovenska Bistrica, 2022.

S temi zankami smo končali programiranje delovanja funkcij serverja in zdaj lahko pričnemo oblikovati spletno stran. Sprva je treba ustvariti spletno stran in ustvariti dinamične spremenljivke iz prej poslanih spremenljivk s komando `Send.HTML()`. Poleg tega moramo določiti, katero vrsto kode pošiljamo:

```
String SendHTML(float temperature,float humidity,float  
pressure,float altitude){  
(določanje dinamičnih spremenljivk)  
    String ptr = "<!DOCTYPE html> <html>\n"; (določanje tipa kode)
```

Potem ko določimo dinamične spremenljivke, je treba izdelati spletno stran, dostopno iz vseh brskalnikov, in ime spletne strani:

```
ptr += "<head><meta name=\"viewport\" content=\"width=device-width,  
initial-scale=1.0, user-scalable=no\">\n"; (omogočanje uporabe  
strani v vseh brskalnikih)  
ptr += "<title>ESP32 Weather Station</title>\n"; (nastavljanje  
naslova spletne strani)
```

Do zdaj smo programirali funkcionalnost spletne strani, v nadaljevanju pa je spletno stran treba še stilizirati. Da pričnemo, moramo v uvodu določiti pisavo, odmike, poravnavo, barvo pisave in velikost pisave:

```
ptr += "<style>html { font-family: Helvetica; display: inline-block;  
margin: 0px auto; text-align: center;}\n";  
ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto  
30px;}\n";  
ptr += "p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";  
ptr += "</style>\n";  
ptr += "</head>\n";  
ptr += "<body>\n";
```

Koklič, Gašper. Avtonomna vremenska postaja ESP32. Srednja šola Slovenska Bistrica, 2022.

Z določenim zapisom lahko pričnemo oblikovati razporeditev podatkov na spletni strani. Sprva je treba urediti naslovno besedilo:

```
ptr += "<div id=\"webpage\">\n";  
ptr += "<h1>ESP32 Weather Station</h1>\n"; (naslov spletne strani)
```

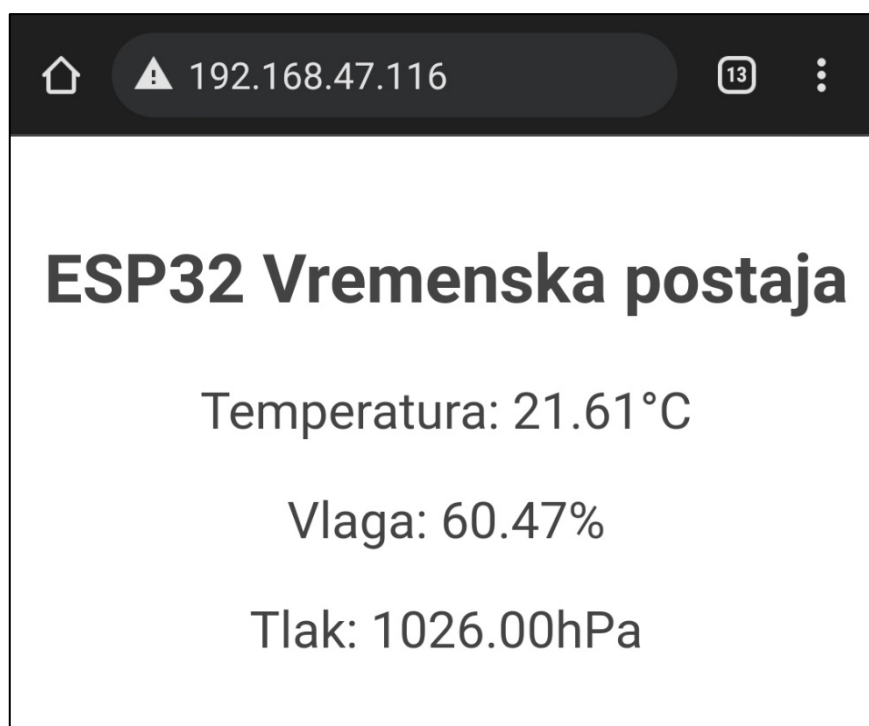
Na koncu je treba spremenljivke sprogramirati še v spletno stran:

```
ptr += "<p>Temperature: ";  
ptr += temperature; (temperatura)  
ptr += "&deg;C</p>";  
ptr += "<p>Humidity: ";  
ptr += humidity; (vlaga)  
ptr += "%</p>";  
ptr += "<p>Pressure: ";  
ptr += pressure; (Tlak)  
ptr += "hPa</p>";  
ptr += "</div>\n";  
ptr += "</body>\n";  
ptr += "</html>\n";  
return ptr;  
}  
(Create A Simple ESP32 Weather Station With BME280, 2022)
```

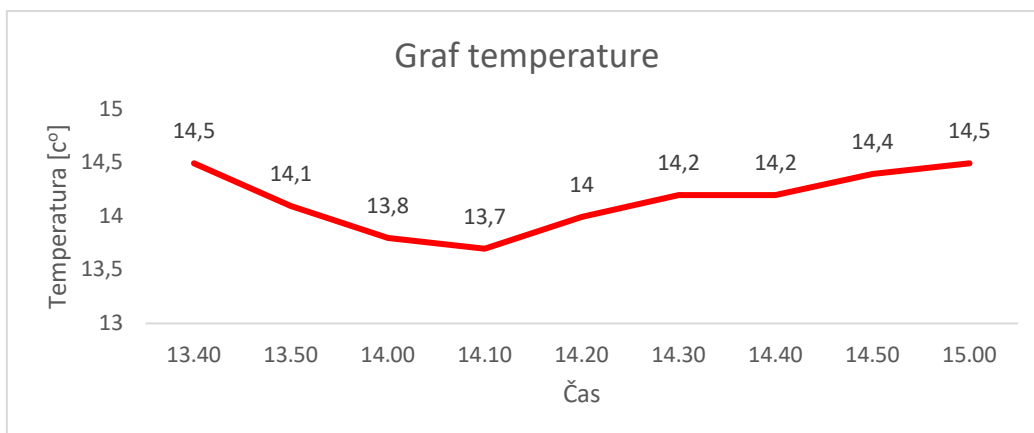
DELOVANJE

Vremenska postaja temelji na napajanju, ki ga dovede 12V Li-on baterija, vgrajena v sončno celico. To omogoči postaji funkcionalnost tudi ponoči, ko sončne celice ne proizvajajo energije. Ko je ESP32 dovedena napetost, ta začne iskati omrežje Wi-Fi, določeno v kodi. Ko se ESP32 poveže v serijski monitor, napiše lokalni IP in prižge senzor BME680. S senzorja prebere temperaturo, vlago in tlak ter podatke v oblike spremenljivk pošlje na server. Server ESP32 nato obdela podatke in jih prikaže na spletni strani z naslovom, ki je enak kot lokalni IP na omrežju Wi-Fi.

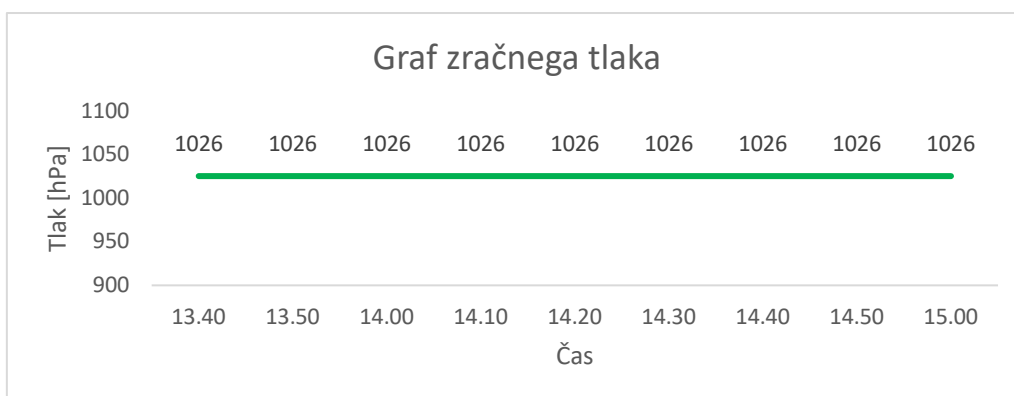
Za predstavitev delovanja vremenske postaje smo izmerili podatke postaje v uri delovanja in podatke razporedili v grafe (glej sliko 11 in grafe 1, 2 in 3).



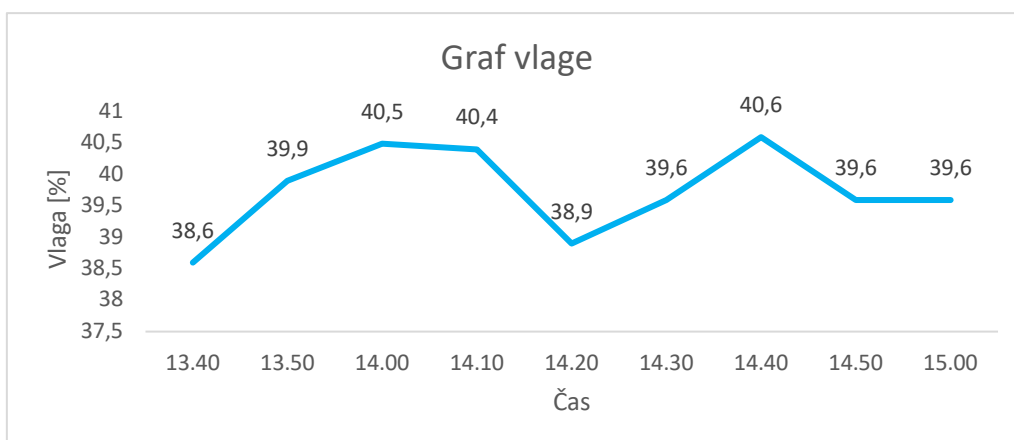
Slika 12: Izgled spletne strani (Gašper Koklič, 2022)



*Graf 3: Časovni potek merjenja temperature 23. 2. 2022
(Gašper Koklič, 2022)*



*Graf 2 Časovni potek merjenja tlaka 23. 2. 2022
(Gašper Koklič, 2022)*



*Graf 1: Časovni potek merjenja vlage 23. 2. 2022
(Gašper Koklič, 2022)*

ZAKLJUČEK

Cilj naloge je bil ustvariti avtonomno vremensko postajo, sposobno merjenja vlage, temperature in tlaka ter pošiljanja teh podatkov na splet, kar smo tudi dosegli.

Z izdelano vremensko postajo bodo lahko bodoči raziskovalci na naši šoli merili meteorološke podatke in meritve uporabili v svojih raziskovalnih nalogah.

Pri ustvarjanju vremenske postaje sem se naučil ogromno o razvojnem ogrodju ESP32 in o ustvarjanju lastnega serverja z ESP32 ter o pošiljanju meritev senzorjev na splet.

Pri nalogi sem naletel na ogromno težav. Prva težava je bila razvojno ogrodje. Sprva sem načrtoval, da bi za razvojna ogrodja uporabil dva Arduina Nana z oddajnikom RF in med njima vzpostavil RF-komunikacijo, s katero bi prenašal meritve senzorja z oddajnega na sprejemni Arduino Nano. Pri tem sem naletel na dva kritična problema. Komunikacija, ki poteka med RF-oddajniki, je omejena na 8 bitov (jaz sem potreboval 32-bitno komunikacijo); spravljanje podatkov na splet preko serijske povezave je slabo dokumentirano in nepotrebno zahtevno. Rešitev za oba problema je bil ESP32, ki lahko brez sprejemnika podatke senzorja objavi na lastni server preko Wi-Fi-povezave.

Poleg težav z Arduino Nanoti sem imel velike težave tudi s prvim ESP32, ki sem ga dobavil, saj je ta za izvajanje Wi-Fi-komunikacije potreboval Wi-Fi-anteno. Ker antene nisem imel, sem moral naročiti ESP32 z vgrajeno Wi-Fi-anteno.

Načrtujem, da bom v nadaljevanju na vremensko postajo vgradil anemometer Rev.P znamke Modern device in senzor UV sevanja ML8511. To bo omogočalo še več uporabnih meritev meteoroloških dejavnikov. Ob tem bom zamenjal trenutni ESP32 s takšnim, ki ima zatič za priklop močnejše antene in vgradil močnejšo Wi-Fi-anteno, ki bo izboljšala doseg vremenske postaje.

VIRI BESEDILA

1. Lutkevich, B. (2022). microcontroller (MCU). Pridobljeno 18. 2. 2022 s: <https://internetofthingsagenda.techtarget.com/definition/microcontrollerESP32->
2. ESP32-WROOM-32 Datasheet (2021) Na espressif.com. Pridobljeno 18. 2. 2022 s: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
3. BME680 – Datasheet (2021) Na bosch-sensortec.com. Pridobljeno 18. 2. 2022 s: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme680-ds001.pdf>.
4. Jacobson, M. Z. (2008). Review of solutions to global warming, air pollution, and energy security. Pridobljeno 18. 2. 2022 s: <https://web.stanford.edu/group/efmh/jacobson/Articles/I/ReviewSolGW09.pdf>.
5. ESP32: BME680 Environmental Sensor using Arduino IDE (Gas, Pressure, Humidity, Temperature) (2022) Na randomnertutorials.com. Pridobljeno 18. 2. 2022 s: <https://randomnertutorials.com/esp32-bme680-sensor-arduino/>.
6. Create A Simple ESP32 Weather Station With BME280 (2022) Na lastminuteengineers.com. Pridobljeno 18. 2. 2022 s: <https://lastminuteengineers.com/bme280-esp32-weather-station/>.

VIRI SLIK

1. ESP32-WROOM-32 Datasheet (2021) [Fotografija na spletu]. Pridobljeno 18. 2. 2022 s: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
1. Espressif ESP32-WROOM-32 8M 64Mbit Flash Wi-Fi Bluetooth Module [Fotografija na spletu]. (2022). Pridobljeno 18. 2. 2022 s: <https://www.robotistan.com/espressif-esp32-wroom-32-8m-64mbit-flash-wi-fi-bluetooth-module>.
2. BME680 (2022) [Fotografija na spletu]. Pridobljeno 18. 2. 2022 s: <https://eu.mouser.com/ProductDetail/Bosch-Sensortec/BME680?qs=v271MhAjFHjo0yA%2FC4OnDQ%3D%3D>.
3. BME680 CJMCU-680 Temperature Humidity Pressure Sensor BOSCH Ultra-small Module (2022) [Fotografija na spletu]. Pridobljeno 18. 2. 2022 s: <https://www.joom.com/en/products/5f378f4fd784b201066ab41e>.