

Razvoj strojno-programске opreme za modularen merilnik fizikalnih veličin

Marko Pavetić

Mentorja: Darko Hercog¹, Tone Lerher²

Univerza v Mariboru,

¹Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor

²Fakulteta za strojništvo, Smetanova ulica 17, 2000 Maribor

marko.pavetic@student.um.si, darko.hercog@um.si, tone.lerher@um.si

Kratek pregled prispevka

V članku je predstavljen razvoj strojno-programске opreme (firmware) za modularno in splošno namensko napravo, ki omogoča merjenje različnih fizikalnih veličin. Razvita naprava bazira na zmogljivem mikrokrmilniku ESP32, ki vključuje mikroprocesor Tensilica Xtensa LX6, Wi-Fi in Bluetooth vmesnik, ter obsežno vhodno izhodno periferijo. Razvita naprava omogoča zajemanje različnih veličin, shranjevanje zajetih podatkov na SD-kartico (Secure Digital) in prenos podatkov z uporabo brezžičnega Wi-Fi omrežja in/ali mobilnega omrežja (GPRS) v oddaljeni strežnik. Z uporabo spletnega brskalnika je mogoče zajete podatke tudi sproti opazovati. Pri zasnovi naprave je bil poudarek na ceni komponent, na multifunkcionalnosti in konfigurabilnosti. Na razvito napravo je tako mogoče priključiti različne senzorje, preko implementirane dostopne točke, pa jo je mogoče z uporabo mobilnega telefona dodatno konfigurirati.

1 Uvod

Ideja za modularno in splošno namensko napravo se je pojavila v sklopu projekta, pri katerem smo izdelali prototip naprave za sledenje transportnih kontejnerjev. Razlogov, da bi sledili kontejnerom je mnogo, eden najpomembnejših je ta, da preprečimo poškodbe in/ali propad blaga med transportom. Prvotna naprava je bila omejena glede na namen in funkcionalnosti, ki jih je ponujala. Zato smo jo nadgradili v modularno in splošno namensko, ki omogoča merjenje različnih fizikalnih veličin, shranjevanje zajetih veličin na SD-kartico ter prenos podatkov z uporabo brezžičnega Wi-Fi-omrežja in/ali mobilnega omrežja (GPRS).

Merilna naprava je sestavljena iz dveh delov, in sicer iz: (1) osnovnega modula, na katerem se nahajajo glavne komponente (torej procesor z vgrajenim Wi-Fi-čipom, SD-kartica, USB, itn.) in (2) iz senzorskega dela, na katerem se nahajajo komunikacijski moduli (npr. GPRS-modul), senzorji in priključki za priključitev zunanjih senzorjev. Z razvojem različnih senzorskih modulov bo takšen merilni sistem možno prilagoditi različnim namenom uporabe, kot je npr. merjenje okoljskih parametrov (torej merjenje temperature, vlage, hrupa, hitrosti vetra, itn.) ali npr. sledenje premičnim objektom, kot so sledenje oseb, avtomobilov, koles, motorjev, ...

Raziskava trga je pokazala, da se ponudniki trudijo narediti in na trg plasirati rešitev, ki zajema tako strojno, kot tudi programsko rešitev. Ponudniki torej nudijo glede na potrebe strank celovite rešitve. Na podlagi teh in ostalih informacij smo ugotovili, kaj mora naš merilnik vsebovati, da bi bil konkurenčen, ter kaj je potrebno narediti bolj(š)e ali drugače, da bi se postavili pred konkurenco oz. našli svojo pozicijo na trgu.

Pred konkurenco bi se postavili z multifunkcionalnostjo in ceno. Prvo smo zagotovili z modularnostjo oziroma možnostjo hitrega dodajanja dodatnih komponent. Drugo pa je rezultat skrbne izbire komponent.

Končna aplikacija oz. algoritem omogoča zajem podatkov iz senzorjev in shranjevanje le-

teh na SD-kartico ter hkrati (po želji uporabnika) tudi pošiljanje podatkov na spletno platformo. S klikom na konfiguracijsko tipko, ki postavi merilnik v način dostopne točke se je mogoče s katerokoli brezžično napravo povezati na merilnik in v implementiranem razširjenem upravitelju povezav (*Connection Manager*) konfigurirati parametre merilnika.

2 Sistem in komponente

Pri izbiri komponent je bilo potrebno upoštevati to, da bo naša naprava kompleksna, večnamenska in predvsem prilagodljiva. Posledično je pri izbiri glavnega modula bilo potrebno poskrbeti, da le-ta vsebuje zadostno število vhodov in izhodov ter ima po možnosti vgrajene različne komunikacijske vmesnike. Pomemben je predvsem serijski vmesnik SPI (*Serial Peripheral Interface*), saj veliko število okoljskih (temperatura, vlaga, pritisk, itn.) in drugih senzorjev deluje prav s pomočjo tega vmesnika. Čeprav obstajajo na trgu zelo poceni komponente, ki bi jih lahko uporabili za ta projekt, se izkaže, da je nakup modulov, ki so že pripravljene za uporabo, cenovno najbolj ugoden. Če upoštevamo, koliko je potrebnega dela, da sami izdelamo modul s samo najbolj osnovnimi komponentami – in koliko vse te komponente stanejo –, ugotovimo, da časovno in cenovno precej prihranimo, če izberemo module in senzorje istega proizvajalca.

Glede na našeto smo izbrali senzorje in ostale module istega proizvajalca, kot je glavni modul. Adafruit ima v ponudbi Feather Wings komponente, ki so osnovane za lahek fizični priklop in programsko implementacijo na naš glavni modul Feather Board. Po precejšnjem razmisleku, raziskovanju in upoštevanju prej navedenih parametrov smo tako izbrali naslednje komponente:

- GPS-modul: *Adafruit Ultimate GPS FeatherWing* [4],
- antena za GPS-modul: *GPS Antenna – External Active Antenna – 3-5 V 28 dB 5 Meter SMA*,

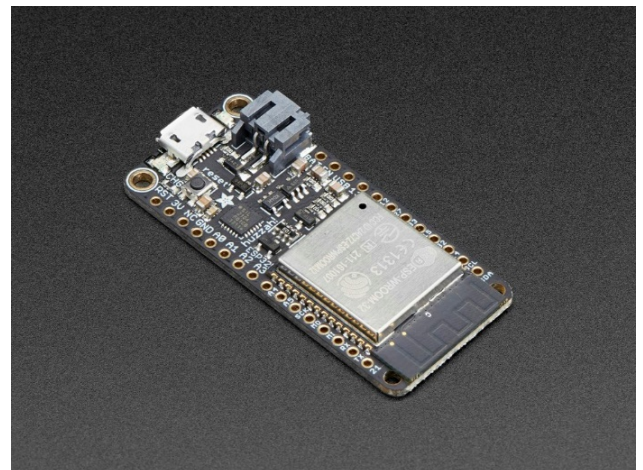
- modul za shranjevanje podatkov na micro SD-kartico: *Adafruit Adalogger FeatherWing* [5],
- senzor temperature, vlage in pritiska: *Bosch BME280* [6],
- modul za mobilni telefon (SIM-modul): *GPRS SIM800C GSM Module 5V/3.3V TTL STM32 C51 with Bluetooth and TTS (Text-to-Speech) for Arduino* [7] in
- litij-ionska polimerna baterija z napetostjo 3,7 V z naknadno določeno kapaciteto.

2.1 Mikrokrmilnik *Adafruit Feather ESP32 Huzzah*

Pri izbiri glavne enote oz. mikrokrmilnika smo dodatno k številu vhodov in izhodov na napravi upoštevali možnost enostavnega programiranja, število obstoječih knjižnic, uporabniške baze znanja in ostale podpore razvijalcem programske opreme. Potrebno je bilo upoštevati majhno porabo energije, saj bo sistem baterijsko napajanje, in možnost programiranja v C++ jeziku (po možnosti v Arduino okolju). Na koncu smo zaradi velike izbire vhodno-izhodnih enot (tako imenovanih Feather) proizvajalca Adafruit in njihove kompatibilnosti ter uporabniške podpore izbrali HUZAH32 ESP32 Feather istega proizvajalca. Razširitvena ploščica Adafruit Feather ESP32 Huzzah je prikazana na sliki 1, nekatere pomembnejše specifikacije pa so:

- 240 MHz dvojedrni mikroprocesor Tensilica LX6
- vgrajeni 520 kB SRAM (*Static Random-Access Memory*),
- vgrajen 802.11b/g/n oddajnik HT40 Wi-Fi,
- vgrajeni dvojni način Bluetooth (klasičen in BLE – *Bluetooth Low Energy* oz. energijsko varčevalni Bluetooth),
- 4 MB bliskovnega (*flash*) pomnilnika,
- vgrajena PCB-antena (*Printed Circuit Board* – ploščica tiskanega vezja),
- 32 kHz kristalni oscilator,

- 3x UART (*Universal Asynchronous Receiver-Transmitter* – univerzalni asinhroni sprejemnik-oddajnik),
- 3 x SPI (*Serial Peripheral Interface* – Serijski periferni vmesnik),
- 2 x I²C (*Inter-Integrated Circuit* – protokol komunikacije med integriranimi vezji),
- 12 x ADC (*Analog-to-Digital Converter* – analogno-digitalni pretvornik) vhodnih kanalov,
- 2 x DAC (*Digital-to-Analog Converter* – digitalno-analogni pretvornik) vhodnih kanalov,
- PWM (*Pulse Width Modulation* – Modulacija širine impulza) / časovni vhod/izhod na voljo na vsakem GPIO-pinu (*General-Purpose Input/Output* - vhodno/izhodni pin za splošno uporabo), in
- podpora za vmesnik SD-kartic.



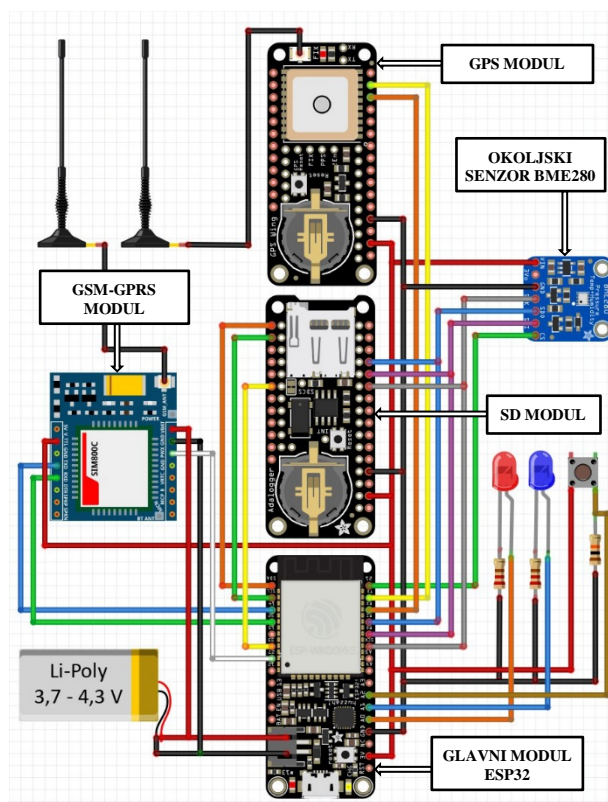
Slika 1: *Adafruit Feather ESP 32 Huzzah* [2]

2.2 Celoten sistem

V celoto smo posamezne module povezali tako, kot je prikazano na sliki 1, kjer se nahaja grafična shema, narejena v programu Fritzing (orodje za dizajn diagramov ožičenja) [8].

Modul za SD kartice in senzor Bosch BME280 sta z mikrokrmilnikom povezana preko SPI povezave, medtem ko sta GPS in SIM modul povezana preko UART serijske povezave. Pri povezovanju SIM-modula je bilo potrebno

nastaviti pine za UART-povezavo. Privzeti ESP32 nima nastavljenih vseh treh UART-serijskih povezav, temveč samo dve. UART0 oziroma serijska povezava 0 (*Serial 0*) se uporablja za povezovanje mikrokrmilnika z računalnikom, UART2 oziroma serijska povezava 2 (*Serial 2*), je zasedena z GPS-modulom. Tretjo oziroma UART1-serijsko povezavo (*Serial 1*), namenjeno SIM800C-modulu, smo omogočili na način, da se je odredil pin v konfiguracijski datoteki `HardwareSerial.cpp`.



Slika 2: Fritzing-shema priključitve senzorskih modulov na glavni modul

3 Končna aplikacija

Na podlagi raziskanih možnosti knjižnic uporabljenih komponent sistema, njihovih demo-programov in izbrane spletne platforme za prikaz podatkov smo kreirali končno aplikacijo oziroma algoritem, ki zajete podatke iz senzorjev shranjuje na SD-kartico in po želji uporabnika hkrati pošilja tudi na spletno platformo. Algoritem temelji na uporabi konfiguracijske

tipke, uporabniškem vmesniku ter Adafruit IO-spletni platformi (*Input-Output*). Vmesnik je narejen na osnovi Wi-Fi-upravitelja povezav (*Wi-Fi Connection Manager*), kateremu smo naknadno dodali polja za vnos lastnih parametrov, tako da je možno dodatno k upravljanju Wi-Fi-povezav nastavljati še vse (pre)ostale parametre. Merilnik se tako nastavlja s pomočjo klika na tipko, ki pokliče upravitelja povezav; ko končamo z vnosom parametrov, se le-ti trajno zapišejo v bliskovni pomnilnik (*flash memory*). Merilnik se nato ponovno zažene in ob zagonu naloži nove parametre, ki zamenjajo stare oziroma privzete parametre.

Čeprav je merilnik večnamenski in za splošno rabo, je bil njegov prvotni namen sledenje kontejnerjem – posledično je algoritem nekoliko prilagojen temu namenu.

Ob zagonu merilnik namreč prične z zajemanjem okoljskih parametrov in GPS-koordinat. Slednje zapisuje v tekstovno datoteko na SD-kartico – to počne periodično, saj je to njegov osnovni namen. Pošiljanje podatkov na spletno platformo je neobvezno; če to možnost vseeno omogočimo, imamo na voljo dva načina prenosa podatkov, in sicer s pomočjo Wi-Fi ali GPRS-omrežja.

Z uporabo upravitelja povezav lahko nastavimo vse parametre omenjenega merilnika, kar med drugim vključuje:

- periodo shranjevanja podatkov na SD kartico in pošiljanja podatkov na spletno platformo, iz senzorjev in GPS-modula,
- določamo vrsto omrežne povezave, Wi-Fi ali GPRS, in pripadajoče poverilnice, ter
- parametre za povezovanje s spletno platformo Adafruit IO.

Polja za določene nastavitve v upravitelju povezav so prikazane na sliki 3.

Prijavite se v omrežje

http://192.168.4.1/wifi?

Configuration

ITK Home	🔒 92%
264090	🔒 46%
5AB99D	🔒 14%

SSID

Password

Samo ena povezava sme biti izbrana. Poleg tega je še treba določiti MQTT podatke za povezovanje s Adafruit IO platformo ter intervala objavljanja podatkov.
(Za vklop Wi-Fi povezave, izberite dostopno točko iz zgornjega seznama, zaključite kvadratik ter nastavite naslove IP, DNS in prehod (gateway) na dnu.)

WiFi povezava

Nastavitve GPRS povezave
(Za vklop GPRS povezave, vpišite poverilnice operaterja ter zaključite kvadratik.)

GPRS povezava

GPRS APN:

GPRS uporabniško ime:

GPRS geslo:

Podatki za MQTT povezavo na Adafruit IO
(Pridobite jih ob registraciji na spletni platformi)

AIO strežnik (Server):

AIO strežniška vrata (Server Port):

AIO uporabniško ime:

AIO ključ:

Vir (feed) za temperaturo:

Vir (feed) za vlago:

Vir (feed) za višino:

Vir (feed) za tlak:

Vir (feed) za GPS:

Slika 3: Konfiguracijske opcije merilnika v upravitelju povezav

Merilnik lahko deluje v treh režimih:

1. podatke zapisuje na SD-kartico,
2. podatke zapisuje na SD-kartico in jih pošilja na spletno platformo z uporabo Wi-Fi omrežja in

3. podatke zapisuje na SD-kartico in jih pošilja na spletno platformo z uporabo GPRS omrežja.

Končna aplikacija deluje v odvisnosti od parametrov, ki jih uporabnik nastavi z uporabo upravitelja povezav. Takšna fleksibilnost je ena od bistvenih ciljev zadanega projekta, saj omogoča, da lahko merilnik enostavno konfiguriramo za različne projekte in ga enostavno povežemo v različna Wi-Fi omrežja.

Po številnih testiranjih različnih upraviteljev povezav smo se odločili za tistega, ki ponuja največ možnosti. Daleč najbolj pomembna je ta, da omogoča preprosto dodajanje novih elementov, kot so razna vnosna polja, gumbi in ostali HTML-elementi. Poleg izbire Wi-Fi-omrežja, kar je privzeto edina možnost teh upraviteljev, omogoča tudi vstavljanje lastnih HTML-elementov. Posledično lahko z njim naredimo še mnogo več.

Uporabili smo ESP_WiFiManager-knjižnico, ki jo je razvil Khoi Hoang. Ta razvijalec si je dodajanje lastnih HTML-elementov zamislil tako, da jih najprej definiramo s funkcijo *ESP_WMPParameter()*, nato pa jo dejansko vstavimo med ostalo kodo z uporabo funkcije *ESP_wifiManager.addParameter()*. Na ta način smo na dokaj preprost način ustvarili vse elemente upravitelja povezav in omogočili konfigurabilnost merilnika [9].

4 Spletna platforma

Zapisovanje podatkov na SD-kartico predstavlja zanesljivo obliko shranjevanja podatkov, vendar je za konkuriranje podobnim izdelkom na trgu zaželeno, da imamo vpogled oz. nadzor nad podatki v realnem času. To omogoča, da lahko pravočasno reagiramo, ko pride do odstopanja v izmerjenih oz. zajetih podatkih, hkrati pa z le-temi lahko manipuliramo. Odstranjevanje kartice iz modula in priklopljanje na računalnik predstavlja nezadostno rešitev, saj je slednje zamudno delo, sam koncept pa zastarel. Ker se v zadnjem obdobju koncepta IoT (*Internet of Things*) in računalništvo v oblaku (*Cloud Computing*) močno razvijata, se

posledično razvijajo platforme za shranjevanje in prikaz podatkov v realnem času.

Merilnik smo povezali z Adafruit IO spletno platformo, katere delovanje temelji na dobro poznanih in strojno nezahtevnih protokolih, poudarek pa je na enostavnosti uporabe. Omenjeno platformo smo izbrali zato, ker jo ponuja podjetje Adafruit, katerega moduli sestavljajo naš merilnik. Drugi razlog je brezplačna uporaba, z možnostjo plačljive nadgradnje. Tretji razlog pa je možnost prenosa zbranih podatkov s platforme v eni sami datoteki. Dodatno lahko omenimo preglednost elementov na nadzorni plošči, ki vizualno atraktivno prikazujejo stanje merilnika.

Adafruit IO je platforma, zasnovana za prikaz in interakcijo s podatki IoT-projektov. Z drugimi besedami lahko rečemo, da je to storitev za komuniciranje z mikrokrmilnikom preko spleta. Ta storitev je narejena s strani podjetja Adafruit, predvsem za namen interakcije njihovih komponent z uporabo interneta s komponentami, ki jih prodajajo. Za vsak njihov modul, ki ima možnost povezovanja na internet, izdelajo ustrezno knjižnico, ki omogoča interakcijo z omenjeno spletno platformo. Z uporabo MQTT protokola pa je možno s spletno platformo povezati tudi module drugih proizvajalcev. Adafruit.io je storitev v oblaku, kar pomeni, da jo ponudnik izvaja, in nam z njo ni treba upravljati. Namenjena je predvsem shranjevanju in pridobivanju podatkov, omogoča pa sledeče:

- prikaz podatkov v realnem času ali na spletu,
- omogoča internetno povezavo s projektom: npr. krmiljenje motorjev, zajemanje podatkov senzorjev, vkapljanje in izklapljanje komponent,
- možnost povezovanja projekta s spletnimi storitvami, kot so: Twitter, RSS (*Really Simple Syndication*) viri, vremenske storitve, itn.,
- možnost povezovanja projekta z drugimi napravami z internetom in
- storitev je brezplačna, z možnostjo plačljive nadgradnje [3].

Na sliki 4 je prikazana naša različica nadzorne plošče merilnika, kjer smo uporabili 3 različne tipe elementov, in sicer: zemljevid (*map*), 4 grafične prikazovalnike (*gauge*) in 4 črtne grafikone (*line chart*).



Slika 4: Nadzorna plošča (dashboard) na platformi Adafruit IO

5 Zaključek

Raziskava trga je pokazala, da konkurence ne manjka, kar pomeni, da z novim produktom ni enostavno prodreti na trg. Je pa tudi pokazala, da so obstoječe rešitve precej drage. Z ugodno ceno, zanesljivostjo in multifunkcionalnostjo, bi lahko bila naša naprava konkurenčna in atraktivna.

Uporabljeni moduli, narejeni s strani priznanega podjetja Adafruit, se lahko iz kitajskih spletnih trgovin dobijo po daleč nižjih cenah. Z uporabljenimi smo vseeno podprli Adafruit, na podlagi njihovih modulov in knjižnic pa smo kreirali lasten sistem, ki lahko v nadaljnji fazi gre v masovno proizvodnjo s cenejšimi različicami modulov ali pa lastnim vezjem, izdelanim iz čipov brez razširitvenih vezij.

Prav tako smo z uporabo Adafruitove spletne platforme merilniku dodali veliko možnosti glede nadzora in manipulacije s podatki – torej povsem brezplačno. Izdelan uporabniški vmesnik je uporaben, pregleden in ima moderen dizajn. Za zahtevnejše uporabnike obstaja tudi plačljiva različica, ki omogoča neomejeno število virov podatkov, kar pomeni, da lahko merilnik nadgradimo s teoretično neomejenim številom senzorjev.

Konfiguracijski uporabniški vmesnik oziroma upravitelj povezav je bistven del sistema, saj je z njegovo uporabo mogoče merilnik nadgraditi in tako razširiti njegove funkcionalnosti. Z dokaj preprostim dodajanjem HTML-elementov lahko v upravitelju dodamo nastavitve za dodatne senzore in module in tako do omenjenih nastavitvev preprosto dostopamo s pomočjo spletnega vmesnika. Med množico upraviteljev smo veliko časa namenili izbiri pravega, ki projektu brezdvomno omogoča že vse predhodno navedeno.

Izziv nam je bil prav tako programsko povezati vse komponente v celoto, saj je pri posameznih testiranjih modulov po navadi potrebno spremeniti le nekaj vrstic demo-primerov kode, da bi uspešno prišli do zelenih rezultatov. Ko pride do prepletanja kode in prilagajanja omenjenih demo-primerov, je potrebno vložiti veliko časa, napora in truda, da pridemo do funkcionalno zadovoljivih rezultatov. Še več časa in truda pa bi potrebovali, če bi kodo dodatno optimirali glede na porabo električne energije. Znotraj implementacije je bila najzahtevnejša Adafruitova MQTT-knjižnica, saj njeni objekti temeljijo na povezovanju s platformo na fiksno odrejenih oz. trdo kodiranih (*hardcoded*) poverilnicah, naš projekt pa temelji na spremenljivih poverilnicah, ki jih uporabnik vnese v uporabniški vmesnik upravitelja povezav. Ker se pošiljanje podatkov lahko nemoteno izvaja preko Wi-Fi-ja in GPRS-povezave, je bilo treba zagotoviti, da se povezavi ne prepletata. Tako smo implementirali algoritem, ki preverja, katero vrsto komunikacije je uporabnik izbral tekom konfiguracije modula, da se neuporabljena povezava izklopi in posledično varčuje energija.

Algoritem merilnika je sicer uporaben in opravlja svojo osnovno nalogo, a se merilnik v trenutni fazi napaja kabelsko in ne baterijsko. Optimizacija kode v smislu varčevanja energije predstavlja naslednji večji izziv, saj takšna optimizacija vključuje implementacijo varčevalnih metod krmilnika, upravitelja povezav, Wi-Fi in GPRS modula.

6 Literatura

- [1] P. Marko, Razvoj strojno-programске opreme za modularen merilnik fizikalnih veličin : magistrsko delo, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor 2021. Dostopno na: <https://dk.um.si/IzpisGradiva.php?id=78341&lang=slv>
- [2] Adafruit HUZZAH32 - ESP32 Feather. Adafruit Industries. 2020. Dostopno na: <https://learn.adafruit.com/adafruit-huzzah32-esp32-feather?view=all>
- [3] Adafruit IO. Adafruit Industries. 2020. Dostopno na: <https://learn.adafruit.com/welcome-to-adafruit-io/what-is-adafruit-io>
- [4] Adafruit Ultimate GPS featherwing. Adafruit Industries. 2020. Dostopno na: <https://learn.adafruit.com/adafruit-ultimate-gps-featherwing?view=all>
- [5] Adafruit Adalogger FeatherWing. Adafruit Industries. 2020. Dostopno na: <https://learn.adafruit.com/adafruit-adalogger-featherwing?view=all>
- [6] Bosch BME280. Bosch Sensortec GmbH. 2020. Dostopno na: <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/>
- [7] SIM800C GSM-GPRS modul. Dostopno na: <https://simcom.ee/modules/gsm-gprs/sim800c/>
- [8] Fritzing orodje za risanje CAD shem. Dostopno na: <https://fritzing.org/>
- [9] ESP_WiFiManager razvijalca khoi-prog. Dostopno na: https://github.com/khoih-prog/ESP_WiFiManager