

Razvoj merne stanice sa web interfejsom, upotrebom otvorenog hardvera u konceptu Industrije 4.0

Ivan Gutai, *Member, IEEE*, Platon Sovilj, *Member, IEEE*, Đorđe Novaković, *Member, IEEE*, Marina Subotin, *Member, IEEE*, Nemanja Gazivoda, *Member, IEEE*, Bojan Vujičić, *Member, IEEE*

Apstrakt — Upotreba otvorenog hardvera pored toga što ubrzava razvoj prototipova uređaja, omogućava i kreiranje korisničkog interfejsa koji je prilagođen prenosnim uređajima kao što su tableti i mobilni telefoni. Upotrebom Espressif ESP32 razvojnog sistema je omogućeno merenje različitih senzora, kreiranje bežične pristupne tačke (eng. Access Point-a), kreiranje web korisničkog interfejsa a zatim i pristup mernom sistemu sa prenosnog uređaja koji poseduje Wi-Fi. U radu je opisan edukativan primer kreiranja merne stanice, kao i konfigurisanja razvojnog okruženja za ESP32 razvojne sisteme. Pored razvojnog sistema, od hardvera je korišćen senzor BME 280 i 3.7 V litijumske baterije dimenzija 18650.

Ključne reči— ESP32, prenosni uređaji, web interfejs, BME280, 3.7 V 18650 baterije, Metrologija, Industrija 4.0

I. UVOD

Merno-informacioni sistemi se koriste u velikom broju i često su podeljeni u grupe, u zavisnosti od toga šta mere. Upotrebom razvojnog sistema zasnovanog na ESP32 čipu [1], se navedenim sistemima proširuju mogućnosti. Na jednom mestu se skupljaju podaci i na neprimetan način se sistem integriše u već postojeću računarsku mrežu ili da se kreira nova, u kojoj će se pratiti rezultati merenja. Prilikom razvoja mikroprocesorskih merno-akvizicionih sistema, može se desiti da velika količina truda i vremena ode u pravcu kreiranja korisničkog interfejsa, koji će biti kako moderan, tako i funkcionalan. Sada je vreme kada su ljudi navikli da većinu informacija preuzimaju sa mobilnih telefona i tableta, pa se kreiranje web korisničkog interfejsa nameće kao efektivno rešenje, koje omogućava bržu izradu prototipa. Zajednice i otvorenog softvera i otvorenog hardvera na nedeljnom nivou omogućavaju korisnicima praktične primere koda kao i iscrpnu dokumentaciju. Da bi se uspešno pratili primeri sa onlajn izvora informacija kao što su [2], [3] ili [4]. Neophodno je znanje programskih jezika C, C++ i JavaScript, kao i napredni nivo korišćenja HTML5 i CSS3. Wi-Fi funkcionalnost kod ESP32 je samo jedna od naprednih mogućnosti, a treba istaći da poseduje i Bluetooth 4.2 i BLE

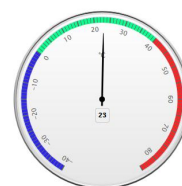
(Bluetooth Low Energy), što ostavlja mogućnosti za dalje proširivanje sistema.

II. KORIŠĆENI HARDVER

Senzor koji meri temperaturu, vlažnost vazduha i pritisak je BME280 [5]. Postoje primeri koda koji prikazuju i nadmorski visinu, što nije metrološki korektno zato što ima dosta pretpostavki. Na osnovu promene izmerenih parametara se može zaključiti, tj. izračunati kolika je promena visine, pod uslovom da smo tog dana u sistem upisali kolika je trenutna vrednost pritiska vazduha na najbližoj obali mora, kako se navodi u specifikaciji proizvođača. Upotrebom dve Panasonic NCR18650B 3.7 V baterije kapaciteta po 3400 mAh, se omogućava prenosivost prototipa. Kompletan interfejs je web, a upotreba kapacitivnog senzora TTP223 predstavlja „dugme“ koje čovek treba da pritisne. U trenutnoj verziji firmvera pritisak, tj. dodir dugmeta predstavlja način na se inicira novo merenje, van definisane učestanosti merenja. Postoji mnogo verzija razvojnih sistema zasnovanih na ESP32 čipu, a korišćen je ESP32-DevKitC [6].

III. POSTAVLJANJE KODA ZA FRONT END NA ESP32

4 MB eksterne memorije poznatijeg kao SPIFFS (Serial Peripheral Interface Flash File System) [7], predstavlja dovoljnu količinu memorije na koju se mogu smeštati svi fajlovi koji omogućavaju iscrpno merenje web interfejsa. Opcija da se radi spajanje stringova i da se HTML, CSS i JavaScript sastavljaju liniju po liniju je moguća, ali je vrlo spora i mogućnost za grešku je neprihvatljivo visoka. Upotrebom SPIFFS-a i programerskog alata iz internet pretraživača na računaru (dugme F12), u realnom vremenu možemo da vršimo test sistema i da dobijemo pregled šta se dešava na front end-u naše aplikacije. Tačnost senzora je ona koju je definisao proizvođač, a mi možemo na front end-u npr. samo da smanjimo opseg skale radi lakše preglednosti. Da bi korisnici lakše prešli u Industriju 4.0, upotrebom modernih tehnologija ipak imamo mogućnost da im prikazemo nešto na šta su decenijama navikli, a to je analogna skala. Na slici 1 je prikazan pun opseg za temperaturu, od -40°C do 85°C koju je definisao proizvođač.



Sl 1. Analogna skala za prikazivanje izmerene temperature

Ivan Gutai – Fakultet tehničkih nauka, Novi Sad, Srbija (e-mail: gutai@uns.ac.rs).

Platon Sovilj – Fakultet tehničkih nauka, Novi Sad, Srbija (e-mail: platon@uns.ac.rs).

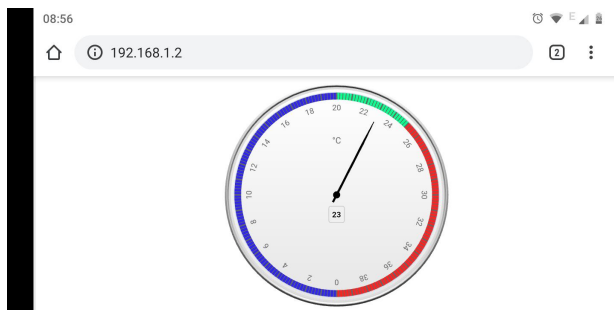
Đorđe Novaković – Fakultet tehničkih nauka, Novi Sad, Srbija (e-mail: djordjenovakovic@uns.ac.rs).

Marina Subotin – Fakultet tehničkih nauka, Novi Sad, Srbija (e-mail: marina.bulat@uns.ac.rs).

Nemanja Gazivoda – Fakultet tehničkih nauka, Novi Sad, Srbija (e-mail: nemanjagazivoda@uns.ac.rs).

Bojan Vujičić – Fakultet tehničkih nauka, Novi Sad, Srbija (e-mail: bojanvuj@uns.ac.rs).

Uređaj uvek treba prilagoditi zahtevima korisnika, pa nema potrebe konstantno težiti da se prikazuju i nepotrebne vrednosti. Npr. ako će prototip ovog uređaja meriti temperaturu optimalnih uslova za rad programera, koji po subjektivnom osećaju može biti u opsegu između 20°C i 25°C, nema smisla prikazivati minus deo skale, kao ni onaj deo preko 40°C. Na slici 2 je prikazan interfejs, posmatran iz Google Chrome pretraživača, sa Android mobilnog telefona, rezolucije ekrana 720 px x 1520 px, u horizontalnom režimu.



Sl 2. Interfejs prikazan na ekranu Android mobilnog telefona

IV. ASINHRONI JAVASCRIPT ZAHTEVI

U web programiranju u tzv. full stack developmentu, spoj između back end-a i front end-a ume često da se iskomplikuje. Potrebno je sa front end-a slati zahteve, a back end mora biti u stanju da na adekvatan način odgovara. Jedna od neprihvatljivih stvari na front end-u je neprestano učitavanje stranice, što se može nekad uočiti čestim treptanjem. U sistemu u kom pratimo razne parametre, najpogodniji način je da se svaki deo učitava, tj. osvežava zasebno, bez potrebe za uzastopnim osvežavanjem cele stranice. AJAX (Asynchronous JavaScript And XML) [8] zahtevi predstavljaju rešenje navedenog problema. Ukoliko na stranici imamo 20 različitih panela, a želimo da se neki ažuriraju na 5 s, neki na 20 minuta, a neki na 8 sati, AJAX zahtevi to omogućuju. Npr. treba samo voditi računa da back end i front end budu skladno podešeni, tj. da ne merimo na back end-u temperaturu jednom u 24 h, a da na front end-u podesimo da se vrednost osvežava na svakih 5 s. Još jedna prednost asinhronih zahteva je što nema blokiranja ostalih funkcija, dok se jedna izvršava.

V. ESP32 FUNKCIONALNOSTI

ESP32 ima dva Wi-Fi režima, STA (Station) i AP (Soft Access Point). U STA režimu se ESP32 povezuje u kućnu ili korporativnu Wi-Fi mrežu, na isti način kako se i mobilni telefoni povezuju. Uređaj dinamički dobije IP adresu, a zatim mu se pristupa sa ostalih uređaja. AP režim je kompleksniji za konfigurisanje, ali njegova prednost je što daje mogućnost da se na lokacijama gde nema Wi-Fi mreže, kreira sopstvena, npr. u automobilu. Postoji mogućnost zadavanja fiksne IP adrese, kao i opsega adresa koje će biti dodeljivane uređajima koji će se povezivati na navedeni merno-informacioni sistem. Za sve koje počinju sa korišćenjem razvojnih sistema kompanije Espressif, dobar početak je ESP8266. Reč je o modelu koji je postao deo IoT (Internet of Things) ekosistema, od 2014. je prisutan na tržištu i postoji izuzetno velika baza znanja i dosta kvalitetnih primera koda. Nakon dve godine stiže nešto skuplji, ali i moderniji je ESP32, a

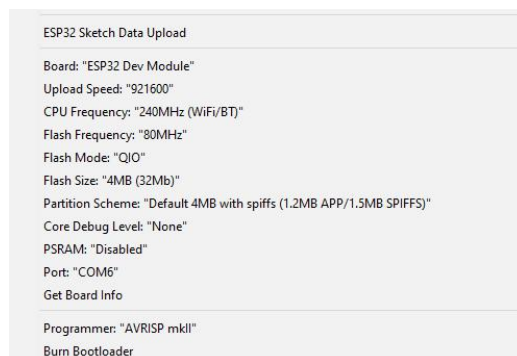
jedini nedostatak je što postoji manji broj primera koda. Jedan od važnijih parametara koje treba istaći je i cena ovakvog razvojnog sistema, koja je ispod 10 €. U Tabeli I su navedeni važniji parametri aktuelnih razvojnih sistema, zasnovanih na otvorenom hardveru [9].

TABELA I
Parametri aktuelnih razvojnih sistema, zasnovanih na otvorenom hardveru

	Arduino Nano	Arduino Mega	Arduino Due	Espressif ESP32
Napon	5 V	5 V	3.3 V	3.3 V
Količina fleš memorije	32 KB	256 KB	512 KB	448 KB
Eksterni SPI RAM	/	/	/	4 MB
SRAM	2 KB	8 KB	96 KB	520 KB + 16 KB (RTC)
Takt	16 MHz	16 MHz	84 MHz	80 MHz - 240 MHz
EEPROM	1 KB	4 KB	/	/
Broj jezgara	1	1	1	2
Wi-Fi	/	/	/	802.11 b/g/n
Arhitektura	8-bit	8-bit	32-bit	32-bit
Približna potrošnja u režimu mirovanja	10 mA	70 mA	110 mA	100 mA

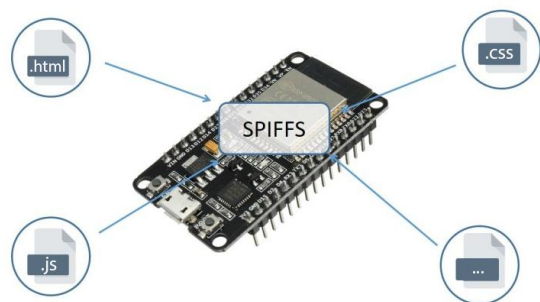
VI. PRILAGODAVANJE RAZVOJNOG OKRUŽENJA ZA ESP32

Kao razvojno okruženje, uz manje modifikacije se može koristiti Arduino IDE (Integrated Development Environment). Postavljanje koda na korišćeni ESP32-DevKitC radi dobro ukoliko je izabrana ploča „ESP32 Dev Module“, a svi ostali parametri su podešeni kao na slici 3.



Sl. 3 Parametri ESP32 u Arduino IDE

Deo „ESP32 Sketch Data Upload“ omogućava korišćenje SPIFFS-a, kao memorije gde se snimaju: html, css, js ili fajlovi sa bilo kakvom ekstenzijom. Na slici 4 je ilustrovano kako se navedeni fajlovi mogu postaviti na memoriju mikrokontrolera, na identičan način kao što bi se fajlovi postavljali na web hosting ili bilo koju verziju fajl sistema.



Sl. 4 Ilustracija postavljanja fajlova na memoriju mikrokontrolera

Za svaki fajl mora da se navede na kojoj lokaciji će se nalaziti. Svi fajlovi će se nalaziti unutar „data“ foldera, tj. index.html će biti na lokaciji: „data/index.html“, itd. Navedene funkcionalnosti omogućavaju kompletno pisanje i testiranje koda u omiljenom razvojnom okruženju, kao što je npr. Microsoft Visual Studio Code. Jedini nedostatak trenutne verzije SPIFFS-a je to što je podržana ravna, (eng. flat) raspodela fajlova, bez mogućnosti upotrebe foldera. Dok ne stigne novija verzija gde će ovaj manji nedostatak biti otklonjen, svi fajlovi se nalaze u „data“ folderu.

PROTOTIP UREĐAJA

U trenutnoj verziji firmvera merenje temperature se vrši na svakih petnaest minuta, dok je ostavljena mogućnost pritiska da dugme radi iniciranja novog merenja. Uređaj radi u STA režimu, a baterije ukupnog kapaciteta od 6800 mAh omogućavaju višednevni rad, u aktivnom režimu rada. Na testu, uređaj je dobio adresu 192.168.1.2, a pošto se cela Wi-Fi mreža nalazi iza proksi servera, na lokalnom računaru i na mobilnom uređaju je bilo potrebno podesiti da se za lokalne adrese ne koristi proksi.

Uređaj koristi indikator napunjenosti baterije oznake 1S, što označava da je baterija puna na 100 %, kada napon na njoj iznosi 4.2 V. Indikator ima potrošnju od približno 5 mAh i ima 4 podoeka.

Svi režimi rada ESP32 su: *Active*, *Modem-sleep*, *Light-sleep*, *Deep-sleep* i *Hibernation*. Na osnovu rezultata sistem testa i na osnovu potrebnog odnosa performansi i trajanja baterije, u nekoj od narednih iteracija će biti primenjen jedan od režima uštede energije, što ostavlja mogućnost da se i do deset puta smanji potrošnja energije.

Testiranje je izvršeno na Google Chrome i Mozilla Firefox internet pretraživačima na Android 9 i na Windows 10 operativnim sistemima.

ZAKLJUČAK

Programiranje, kao i prelazak u Industriju 4.0 bi trebalo da bude u istoj meri i izazovno i zabavno, a najvažnije odluke ipak treba da donosi čovek. Takođe, uređaji treba da se uklape u sisteme koje smo navikli da koristimo. Uređaj koji je povezan na korporativnu ili kućnu Wi-Fi mrežu predstavlja

komotan način da se u novom tabu internet pretraživača otvori nešto ozbiljno, kao što je nadzor merno-informacionog sistema.

U prototip uređaja je ugrađeno jedno kapacitivno dugme, koje će u svim verzijama firmvera služiti kao nešto sa čime mora da se potvrdi neka akcija. Npr. ako u web interfejsu ostavimo mogućnost promene nekog od parametara, kao što je tačno vreme u sistemu, bitno je da pored toga što smo mi ili neko od korisnika sistema „kliknuli“ u web interfejsu, radnju potvrdimo i na samom hardveru.

U elektrotehnici većito postoji podela, na softveraše i hardveraše. Web dizajn je specifična nauka, koja se često svodi na zanat i hardveraši u većini slučajeva ne smatraju ozbiljno softver developere, kao ni front end developere. U ovom momentu postavljanje fajlova potrebnih za web aplikaciju na memoriju razvojnog sistema predstavlja relativno novu stvar, a ona u velikoj meri olakšava softverašima da koriste hardver samo u onoj meri koliko im je to potrebno.

Hardveraši, spremite se za web dizajn na mikrokontrolerima.

ZAHVALNICA

Ovaj rad je delom podržan od strane projekta ELEMEND (šifra projekta: 585681-EEP-1-2017-EL-EPPKA2-CBHE-JP).

LITERATURA

- [1] https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [2] <https://www.sitepoint.com/best-javascript-charting-libraries>
- [3] <https://lastminuteengineers.com>
- [4] <https://randomnerdtutorials.com>
- [5] <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280>
- [6] <https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/get-started-devkita.html>
- [7] <https://github.com/me-no-dev/arduino-esp32fs-plugin>
- [8] <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>
- [9] <https://www.arduino.cc>

ABSTRACT

Usage of open hardware speeds up the development of prototypes and provides the web interface which is accommodated to portable devices such as mobile phones or tablets. Using Espressif ESP32 development system enables measurement using various sensors, the creation of Wi-Fi Acces Point, the creation of a web user interface, and access to measuring system from some Wi-Fi enabled portable devices. In this paper, an educational example is given for the creation of a measuring station and configuring an integrated development environment for ESP32 development boards. Besides the development system, hardware as a BME280 sensor and 3.7 V 18650 lithium batteries are used.

Development of measuring station with web interface using open hardware in the Concept of Industry 4.0

Ivan Gutai, Platon Sovilj, Đorđe Novaković,
Marina Subotin, Nemanja Gazivoda, Bojan Vujičić