

**PRIMENA ARDUINO PLATFORME NA AUTOMATIZACIJU
NAVODNJAVANJA VOĆNJAKA - DEO 2: SOFTVER
AN APPLICATION OF ARDUINO PLATFORM ON AUTOMATION OF
ORCHARD IRRIGATION - VOLUME 2: SOFTWARE**

Miroslav Marković, Nebojša Mitrović, Filip Vasić, Đorđe Mihailović
Visoka škola strukovnih studija Arandelovac

REZIME: Rad prezentuje implementaciju jednog sistema za automatizaciju navodnjavanja voćnjaka. Sistem meri temperaturu, vazdušni pritisak, vlažnost vazduha i vlažnost zemljišta, i u zavisnosti od izmerenih vrednosti kontroliše navodnjavanje. Celokupan sistem je povezan na internet. Izmerene vrednosti se prikazuju na namenskoj veb stranici. Za implementaciju je korišćena hardverska platforma Arduino sa odgovarajućim senzorima temperature, pritiska i vlažnosti. Rad prikazuje karakteristike Arduino platforme, izradu sistema za akviziciju, kreiranje veb servisa na kome se prikazuju izmerene vrednosti, kao i instalaciju i podešavanje veb servera na kome se nalazi veb servis. Sistem je namenjen za primenu u voćnjacima, i jednostavan je za implementaciju i upotrebu.

KLJUČNE REČI: Arduino, automatizacija, veb servis, veb stranica.

ABSTRACT: This paper presents an implementation of a system of automation of orchard irrigation. The system measures temperature, air pressure, air humidity and soil humidity, and depending on the measured values controls the irrigation. The system is connected to the Internet. The measured values are shown on a dedicated web page. For the system implementation, the Arduino hardware platform is used in combination with sensors of temperature, pressure and humidity. The paper shows the characteristics of Arduino platform, the creation of acquisition system, cration of the web service intended to display measured values, and installation and setup of the web server which hosts the web service. The system is intended to be used in orchards, it is simple to implement and use.

KEY WORDS: Arduino, automation, web service, web page.

1. UVOD

U ovom radu predstavlja se automatski računarski sistem za merenje nadmorske visine, temperature, apsolutnog vazdušnog pritiska, vlažnosti zemljišta, kontrolu navodnjavanja zemljišta, i očitavanje izmerenih vrednosti na ekranu odabrane Arduino hardverske platforme, kao i u okviru namenske veb stranice. Prikazuje se način kako se vrši merenje, kako se izmerene vrednosti distribuiraju kroz mrežu, gde se čuvaju i kako se čine dostupnim krajnjem korisniku -- vlasniku voćnjaka. Takođe, rad predstavlja sve tehnologije korišćene pri izradi ovog sistema.

Ovaj rad, koji će predstaviti softverski aspekt projekta, je nastavak prethodnog rada [1] koji je predstavio hardverski aspekt. Ovakvo razdvajanje na dva dela je bilo neophodno zbog obimnosti projekta.

2. PROGRAMSKI KOD

Ova sekcija će objasniti i prikazati samo najbitnije delove programskog koda, dok će ostali biti samo pomenuti. Izabrano je da se opiše i prikaže ono što je najilustrativnije.

2.1 Uključivanje potrebnih biblioteka

Kao prvi korak u pisanju koda, treba uključiti biblioteku koje će se koristiti prilikom pisanja programa. Biblioteka Adafruit_CC3000.h omogućava korišćenje CC3000 štita za bežičnu konekciju sa internetom. Biblioteku LiquidCrystal.h koristi HD44780 16x2 LCD modul za ispis na ekranu. Senzor

vazdušnog pritiska i temperature BMP180 koristi Adafruit_BMP085.h biblioteku. Biblioteka DHT.h i DHT_U.h koristi DHT11 senzor vlažnosti vazduha.

Biblioteka SPI.h se koristi da bi se omogućila serijska veza sa računarom: samim tim, na serijskom monitoru se može pratiti tok izvršavanja programa i uočiti ukoliko dolazi do nepravilnosti u radu sistema. Biblioteka SPI.h i delovi programa gde se ona koristi nisu neophodni za neometan rad platforme. U ovom koraku kreira se i funkcija za resetovanje Arduino modula koja će kasnije biti potrebna tokom izvršavanja programa.

Ovo je prikazano sledećom linijama koda:

```
1. #include <Adafruit_CC3000.h>
2. void(* resetFunc) (void) = 0;
```

Na sličan način se uvezu sve ostale pomenute biblioteka.

2.2 PODEŠAVANJA BEŽIČNE MREŽE I ADRESE SERVERA

U narednom koraku definišu se podešavanja bežične mreže na koju se sistem povezuje, kao i pinovi na koje se povezuju ostale komponente. Potrebno je uneti SSID (naziv mreže), lozinku i sigurnosni protokol bežične mreže. Pinovi 14 i 15 na koje su povezani E i RS pinovi displeja releji su u stvari analogni pinovi A0 i A1. Kada se analogni pinovi koriste kao digitalni, pin A0 se definiše kao pin 14, A1 kao pin 15, i tako redom do A5 kao pin 19.

```
1. #define ADAFRUIT_CC3000_IRQ 3 // Mora biti 2 ili 3
2. #define ADAFRUIT_CC3000_VBAT 5 // Moze bilo koji pin
3. #define ADAFRUIT_CC3000_CS 10 // Moze bilo koji pin
```

```

4. // UNO za hardverski SPI koristi pinove SCK = 13, MISO = 12, MOSI
   = 11
5.
6. #define WLAN_SSID "View" // SSID mreze na koju se povezuje
7. #define WLAN_PASS "3b4e273144ab" // lozinka mreze na koju se
   povezuje
8. #define WLAN_SECURITY WLAN_SEC_WPA2 // sigurnosni protokol
   mreze
9.
10. #define RS_PIN 15 // Definiseмо analogni A1 pin da radi kao digi-
    talni 15
11. #define E_PIN 14 // Definiseмо analogni A0 pin da radi kao digitalni
    14
12. #define D4_PIN 6
13. #define D5_PIN 7
14. #define D6_PIN 8
15. #define D7_PIN 9
16.
17. #define DHT_pin 2
18. #define DHT_tip DHT11
19.
20. #define vlaznost_zemljista_PIN A3
21.
22. #define navodnjavanje_PIN 16 // Definiseмо analogni A2 pin da radi
    kao digitalni 16

```

Takođe, deklarise se promenljiva adresa [] u koju se upisuje link web stranice, i promenljiva ip u koju će kasnije tokom izvršavanja programa biti upisana IP adresa servera na kojoj se web stranica nalazi. Deklarisu se promenljive u koje će se upisivati izmerene vrednosti, kao i promenljiva id_arduino koja služi da se utvrdi koji je Arduino poslao podatke u slučaju da postoji više mernih uređaja.

2.3 Početna podešavanja i beskonačna petlja

U setup() funkciji postavljaju se početna podešavanja. Isključuje se modul za SD karticu tako što se postavi da radi kao izlaz i dodeli mu se vrednost HIGH. Zatim se postavljaju vrednosti pina relejnog modula koji kontroliše navodnjavanje, u ovom slučaju HIGH znači da kroz kolo sekundara ne protiče struja, odnosno sistem navodnjavanja nije pokrenut. Funkcijom Serial.begin(115200) započinje se serijska komunikacija sa računarnom, brzinom od 115200 bita u sekundi. CC3000 WiFi štit se pokreće funkcijom cc3000.begin(), 16x2 LCD modul funkcijom lcd.begin(16,2), BMP180 senzor funkcijom bmp.begin() i DHT11 senzor funkcijom dht.begin().

Štit CC3000 pokušava da se poveže na mrežu i u slučaju uspešnog povezivanja ispisuje poruku na ekranu uređaja i serijskom monitoru. Ukoliko nema mreže, prelazise u "offline" režim, što se ispisuje i na ekranu uređaja (i na serijskom monitoru ukoliko je povezan sa računarnom), prolazi se kroz funkcije merenja, kontrole navodnjavanja i prikaza na ekranu uređaja, a zatim se restartuje i ponovo prolazi kroz setup funkciju.

U loop() petlji pozivaju se funkcije sledećim redosledom. Prvo se poziva funkcija kojom se dobijaju vrednosti merenja, zatim se vrši kontrola navodnjavanja, pripremaju se podaci za slanje na server, šalju se podaci na server i prikazuju se podaci merenja na ekranu uređaja. U nastavku teksta biće objašnjene najvažnije od pomenutih funkcija.

2.4 Merenje temperature, vlažnosti vazduha, vazdušnog pritiska i nadmorske visine

U funkciji merenje() dobijaju se podaci o izmerenim vrednostima sa BMP180 i DHT11 senzora, i te vrednosti se dodeljuju promenljivama, s tim da se vrednost apsolutnog vazdušnog pritiska deli sa 100 da bi se dobijena vrednost prikazala u hpa (mbar). Za nadmorsku visinu se unosi vrednost relativnog vazdušnog pritiska.

```

1. void merenje()
2. {
3.   temperatura_BMP = bmp.readTemperature();
4.   aps_pritisak_BMP = bmp.readPressure() / 100; // Deli sa 100 da bi
   dobili hPa(mbar)
5.   nadmorska_visina_BMP = bmp.readAltitude(bmp.readPressure() +
   3000);
6.   vlaznost_vazduha = dht.readHumidity();
7. }

```

2.5 Kontrola navodnjavanja

U funkciji kontrola_navodnjavanja() vrši se merenje vlažnosti zemljišta i proverava da li je ona opala ispod određene granice (u ovom slučaju 30 %). Pošto funkcija analogRead() vraća izmerene rezultate u opsegu od 0-1023, koristi se funkcija map() da bi se vrednost vlažnosti zemljišta prikazala u opsegu od 0-100%.

U slučaju da vlažnost zemljišta iznosi manje od 30% zatvaraju se kontakti na sekundarnom kolu releja, kroz strujno kolo koje pokreće sistem navodnjavanja kreće da protiče struja i uključi se električna pumpa zadužena za navodnjavanje na interval od 6 sekundi. Ako vlažnost zemljišta iznosi 30% ili više, kontakti na sekundaru releja ostaju otvoreni i električna pumpa je isključena. Na ekranu uređaja se ispisuje poruka da je izvršena kontrola navodnjavanja.

2.6 Slanje podataka i prikazivanje na ekranu

U funkciji priprema_podataka_za_slanje() se u promenljivo podaci smeštaju sve vrednosti promenljivih u obliku koji je pogodan za slanje u okviru POST zahteva [2]. Tokom smeštavanja sve promenljive se konvertuju u string tip podatka. Ukoliko je Arduino povezan sa računarnom ispisuje se vrednost promenljive podaci na serijskom monitoru.

```

1. void priprema_podataka_za_slanje()
2. {
3.   podaci = "t=" + String(temperatura_BMP) +
4.   "&p=" + String(vazdusni_pritisak_BMP) +
5.   "&l=" + String(nadmorska_visina_BMP) +
6.   "&z=" + String(vlaznost_zemljista) +
7.   "&v=" + String(vlaznost_vazduha) +
8.   "&n=" + String(navodnjavanje) +
9.   "&i=" + String(id_arduino);
10. Serial.println(podaci);
11. }

```

U funkciji slanje_podataka() preko promenljive adresa dolazi se do IP adrese servera. To se radi tako što se promenljivoj ip dodeli inicijalna vrednost 0; zatim se pomoću while petlje koja se izvršava sve dok je vrednost ip = 0, pomoću još jedne

while petlje koja se izvršava sve dok se pomoću metode `getHostByName(adresa, &ip)` ne dobije IP adresa servera i upiše u promenljivu `ip`.

Zatim se vrši provera da li se Arduino povezao sa web serverom koji čuva i prikazuje podatke. Ako je povezivanje uspešno, HTTP metodom POST šalju se serveru izmerene vrednosti, i dobija se poruka na serijskom monitoru da je POST zahtev poslat. Ako povezivanje nije uspešno, dobija se poruka na serijskom monitoru o neuspešnom povezivanju. Ako je zahtev poslat, na serijskom monitoru može se videti odgovor servera. Zatim se prekida veza sa serverom.

U funkciji `prikaz_na_ekranu()` naizmenično se prikazuju izmerene vrednosti. Kako je moguće na ekranu prikazati samo 16 karaktera u jednom redu, u slučajevima gde treba prikazati vazdušni pritisak i vlažnost zemljišta, pomoću `for` petlje i `scrollDisplayLeft()` metode pomeraju se karakteri u levo za po jedno mesto u sekundi dok se ne ispiše ceo naziv.

3. VEB SERVIS I STRANICE

Da bi se prikazale izmerene vrednosti na internetu, potrebno je kreirati veb sajt na kome će se prikazivati podaci koje Arduino šalje. Za kreiranje stranica sajta koristi se HTML jezik koji je specijalno namenjen opisu strukture veb stranica, CSS koji se koristi da se definiše izgled i formatira sadržaj u okviru HTML stranice, programski jezik PHP koji će omogućiti kreiranje dinamičkih HTML stranica, programski jezik JavaScript koji omogućava izvršavanje skripti na računaru korisnika, odnosno da se neki element promeni bez osvežavanja stranice, i MySQL sistem za upravljanje bazom podataka u kojoj se čuvaju podaci o korisnicima i izmerenim vrednostima.

Na računaru na kom se kreira sajt potrebno je instalirati XAMPP. To je besplatna softverska platforma koja sadrži interpretere za PHP i Perl, Apache HTTP server i MariaDB server bazu podataka [3].

Korisniku treba da budu dostupne stranica za prijavljivanje, stranica za registraciju, indeksna stranica gde će se prikazivati poslednje izmerene vrednosti, i stranice na kojima će se prikazivati grafici izmerenih vrednosti u poslednja 24 časa.

3.1 Stranica za prijavu korisnika

Kada se pristupi sajtu, na ekranu se pojavljuje forma za prijavu korisnika (Slika 1), gde se unose korisničko ime i lozinka. U slučaju da korisnik još uvek nema kreiran korisnički nalog, klikom na link „Registrujte se“ otvara se stranica za registraciju korisnika.



Slika 1: Stranica za prijavu korisnika

Nakon što se klikne na dugme za prijavu, POST metodom se iz tekstualnih polja uzimaju vrednosti korisničkog imena i lozinke, zatim se vrši provera da li u bazi postoji korisnik koji ima uneto ime i lozinku. Ukoliko takav korisnik postoji pokreće se sesija kojoj se dodeljuje vrednost ID (identifikaciono polje korisnika) i korisnik se preusmerava na indeksnu stranicu. Ukoliko ne postoji korisnik sa unetim korisničkim imenom i lozinkom, ispisuje se poruka o grešci.

3.2 Stranica za registraciju korisnika

Stranica za registraciju (Slika 2) korisnika služi da se podaci o novom korisniku unesu u bazu podataka. Na stranici za registraciju korisnika se nalazi forma sa poljima u koja je potrebno uneti korisničko ime, lozinku, ponovo uneti lozinku da bi se izvršila provera da nije došlo do greške prilikom kucanja lozinke, ID koji je definisan u programu Arduino mikrokontrolera (jedinstven je za svaki uređaj) i sigurnosni kod. Takođe se tu nalazi i tekstualni blok koji predstavlja uputstvo kako popuniti formu i link koji vodi nazad na stranicu za prijavu.



Slika 2: Stranica za registraciju korisnika

Broj korisnika koji se mogu registrovati sa istim ID kodom je deset, što znači da izmerene vrednosti sa jednog uređaja može da prati najviše deset korisnika. Nakon što se popune sva polja u formi i klikne za registraciju, u donjem levom uglu pojavljuje se poruka o uspešnoj registraciji ili napravljenoj nepravilnosti prilikom unošenja podataka.

3.3. Indeksna stranica

Na indeksnoj stranici (Slika 3) su prikazani podaci sa poslednjih merenja. U zaglavlju stranice, u levom uglu se nalazi dugme za meni, a u desnom za odjavu korisnika. Sadržaj stranice je prikazan u obliku tabele u kojoj se nalaze analogni pokazivači koji prikazuju poslednje izmerene vrednosti i plavi ekran koji je sličan ekranu koji se nalazi na samom uređaju i na kome su ispisani podaci o nadmorskoj visini i vremenu poslednjeg merenja. Za prikazivanje poslednjih izmerenih vrednosti u obliku analognih merača koristi se JavaScript dodatak HTML Canvas Gauges v2.0 [4]. Na dnu stranice prikazuje se trenutno vreme i datum.



Slika 3: Indeksna stranica

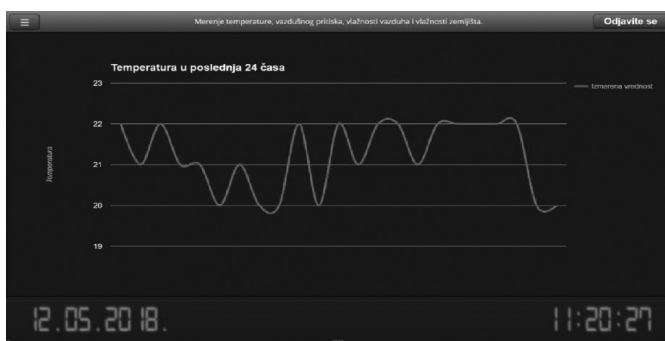
U slučaju da se stranici pristupa sa uređaja čija je širina ekrana manja od 720 piksela (npr. mobilni telefon) korisnik se preusmerava na stranicu za smartfone, gde su elementi stranice zbog bolje preglednosti postavljeni kao na Slici 4.



Slika 4: Izgled indeksne stranice na smartfonu

3.4 Stranica izmerenih vrednosti u poslednja 24 časa

Na stranici izmerenih vrednosti u poslednja 24 časa prikazuje se grafik odabrane vrednosti u navedenom vremenskom periodu. Za prikazivanje tog grafika JavaScript koristi dodatak Google Line Chart [5]. Na Slici 5 prikazan je grafik temperature u poslednja 24 časa.



Slika 5: Grafik izmerene temperature u poslednja 24 časa

4. BAZA PODATAKA

MySQL je najzastupljeniji sistem za upravljanje relacionim bazama podataka [6]. Dostupan je na većini operativnih sistema i omogućava pristup bazi u većini programskih jezika.

Instaliranjem programa XAMPP istovremeno se instalira i aplikacija phpMyAdmin, napisana u PHP programskom jeziku koja omogućava da se bazama podataka upravlja iz pregledača. Na indeksnoj stranici phpMyAdmin-a date su mogućnosti podešavanja skupa karaktera za konekciju sa serverom, jezika i grafičkog interfejsa (teme, font). Sa leve strane stranice nalazi se meni sa opcijom za kreiranje nove baze i spiskom postojećih baza podataka. Na vrhu stranice nalazi se meni sa opcijama za upravljanje bazama, SQL interpreter, podešavanja korisničkih naloga, uvoz - izvoz baze, phpMyAdmin podešavanja, itd. Na dnu stranice se nalazi MySQL konzola.

Naredbe i upiti mogu se unositi grafički, odabirom stavki iz brojnih podmenija, putem SQL interpretera ili MySQL konzole. PhpMyAdmin-u se pristupa unošenjem linka localhost/phpmyadmin.

Odabirom opcije New otvara se stranica na kojoj se unose naziv, set karaktera i kreira baza podataka. Baza će se nazvati `arduino_db`, set karaktera treba podesiti na `latin1_general_cs` i zatim kliknuti na Create. Pošto je baza kreirana, potrebno je kreirati tabele sa podacima o korisnicima i izmerenim vrednostima. U okviru `arduino_db` baze iz menija treba odabrati karticu SQL, koja otvara stranicu na kojoj se nalazi SQL interpreter i uneti sledeći query:

```
1. CREATE TABLE `korisnici`
2. (
3. `id_korisnika` int(11) NOT NULL AUTO_INCREMENT,
4. `ime` varchar(255) NOT NULL,
5. `lozinka` varchar(255) NOT NULL,
6. `id_arduino` int(11) NOT NULL,
7. PRIMARY KEY (id_korisnika)
8. );
9.
10. CREATE TABLE `merenja`
11. (
12. `id_merenja` int(11) NOT NULL AUTO_INCREMENT,
13. `id_arduino` int(11) NOT NULL,
14. `vreme_datum` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
15. `nadmorska_visina` int(11) NOT NULL,
16. `vazdusni_pritisak` int(11) NOT NULL,
17. `temperatura` int(11) NOT NULL,
18. `vlaznost vazduha` int(11) NOT NULL,
19. `vlaznost zemljista` int(11) NOT NULL,
20. `navodnjavanje` int(11) NOT NULL,
21. PRIMARY KEY (id_merenja)
22. );
```

Kada se izvrši ovaj query, u `arduino_db` bazi podataka se dobijaju dve tabele: `korisnici` i `merenja`.

5. PHP SKRIPTE

PHP je skriptni programski jezik prvenstveno namenjen razvoju Web sadržaja. Veliku popularnost je stekao zbog jednostavnosti pisanja koda i sintakse nasleđene iz programskog jezika C. Kod se izvršava na strani servera i generiše HTML

kod koji se prikazuje u okviru internet pregledača na računaru klijenta. Ima mogućnost proceduralnog i objektno-orijentisanog programiranja [7].

Pomoću PHP programskog jezika kreira se skripta za povezivanje sa bazom podataka, klasa korisnik, klasa merenje i kompletan logički deo sajta (navigacija stranica i sadržaja, trenutno vreme).

5.1. Datoteka za konekciju sa bazom podataka (konekcija.php)

Datoteka kojom se povezuje sa bazom se sastoji od try – catch naredbe. U metodu za povezivanje sa bazom podataka (new PDO) treba uneti podatke o nazivu baze (arduino_db), serveru gde je smeštena (localhost), korisničkom imenu (root) i šifri korisnika (nije postavljena) i dodeliti je promenljivoj konekcija.

Zatim iz bezbedonosnih razloga treba postaviti attribute PDO::ATTR_ERRMODE i PDO::ERRMODE_EXCEPTION da bi se u slučaju greške prekinula konekcija sa bazom. U slučaju da se iz nekog razloga ne ostvari konekcija sa bazom ispisuje se poruka o grešci. Ova datoteka naziva se konekcija.php i smešta se u folder klase.

```

1. <?php
2.
3. try
4. {
5. $konekcija = new PDO("mysql:host=localhost;dbname=arduino_
db", 'root', '');
6. $konekcija -> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMO
DE_EXCEPTION);
7. }
8. catch (PDOException $greska)
9. {
10. echo $greska -> getMessage();
11. }
12.
13. ?>
    
```

5.2 Klasa Merenje

Datoteka merenje.php se sastoji iz fajla za konekciju (konekcija.php) koji se učitava funkcijom include i klase Merenje koja sadrži konstruktorsku funkciju (__construct) koja omogućava rad sa bazom i funkciju za učitavanje poslednjih merenja, merenja u poslednja 24 časa, ažuriranje podataka o merenjima u tabeli baze i brisanje zapisa starijih od 24 časa. Na kraju ovog fajla prikazan je način kreiranja objekta \$merenje.

```

1. <?php
2.
3. include "konekcija.php";
4.
5. class Merenje
6. {
7. function __construct($konekcija)
8. {
9. $this -> db = $konekcija;
10. }
11. /*.....
12. Ostatak koda prikazan je kroz funkcije u narednim poglavljima
    
```

```

13. ....
14. */
15. }
16.
17. $merenje = new Merenje($konekcija);
18.
19. ?>
    
```

5.3 Funkcije za merenje

Funkcija PoslednjaMerenja selektuje poslednji unos merenih vrednosti iz tabele merenja i naredbom return vraća podatke o poslednjim merenjima u obliku niza. U slučaju da nema podataka o merenjima, funkcija postavlja pokazivače merenja na nulte vrednosti.

Funkcijom Poslednja24h selektuju se sve izmerene vrednosti u prethodna 24 časa, a pošto se novi rezultati merenja upisuju na svakih 60 sekundi, znači da postoji 1440 podataka o merenim vrednostima po danu. Zbog velikog broja podataka i nemogućnosti da se svi prikažu na grafiku, treba selektovati poslednji i svaki 15. prethodni upis: tako se postiže prikaz podataka u vremenskom intervalu od 15 minuta, odnosno 4 po času, što je više nego dovoljno s obzirom na brzinu kojom merene vrednosti mogu da variraju.

```

1. public function Poslednja24h($id_arduino, $vrednost)
2. {
3. $merenja24h = $this -> db -> prepare("SELECT *
4. FROM merenja
5. WHERE vreme_datum >= now() - INTERVAL 1 DAY
6. AND id_arduino = :id_arduino
7. ORDER BY id_merenja DESC
8. LIMIT 1440
9. ");
10. $merenja24h -> bindParam(':id_arduino', $id_arduino);
11. $merenja24h -> execute ();
12. $brojac = 1;
13.
14. while ($podatak = $merenja24h -> fetch(PDO::FETCH_ASSOC))
15. {
16. if ($brojac == 1)
17. {
18. $niz_podataka_24h[] = "[" . $podatak['vreme_datum'] . ", " .
19. $podatak[$vrednost] . "]" ;
20. }
21. if ($brojac == 15)
22. {
23. $brojac = 0;
24. }
25. $brojac ++;
26. }
27. @$podaci_24h = implode(',', $niz_podataka_24h);
28. return $podaci_24h;
29. }
    
```

Kada merni uređaj pošalje rezultate merenja, oni se upisuju u promenljive funkcije Azuriranje (\$id_arduino, \$temperatura, \$vlaznost_vazduha, \$vazdusni_pritisk, \$nadmorska_visina, \$vlaznost_zemljista, \$navodnjavanje). Zatim se vrednost promenljive \$navodnjavanje promeni u 1 ili 0 u zavisnosti da li je navodnjavanje uključeno ili isključeno. Nakon toga se podaci o pristiglim merenjima upisuju u tabelu merenja.

Funkcija Brisanje služi da obriše sve zabeležene merene vrednosti iz tabele merenja starije od jednog dana.

6. VEB SERVER

Web server je potreban zbog čuvanja izmerenih podataka i njihovog prikazivanja putem Web stranica. Za operativni sistem odabran je Ubuntu Server 16.04 Long Term Support x86 (namenjen računarima sa 32-bitnom arhitekturom). Ubuntu Server je široko rasprostranjen serverski sistem, veoma jednostavan za korišćenje, odličnog kvaliteta, sa dugotrajnom podrškom i besplatan za preuzimanje sa zvanične stranice [8].

6.1 Ubuntu Server i operativni sistem

Ubuntu Server izlazi u verzijama za x64, x86 i ARM procesore. Računar na koji se instalira server je starije generacije i koristi 32-bitni (x86) procesor, što je razlog da se odabere prethodno navedeni operativni sistem. Nakon što se preuzme operativni sistem, kreira se butabilan CD ili USB. Za kreiranje butabilne USB memorije koristi se besplatni program Rufus, dostupan za preuzimanje sa zvanične stranice [9].

Operativni sistem je jednostavan za korišćenje i samo je potrebno odabrati USB memoriju na koju treba smestiti instalacioni disk, postaviti da se disk kreira korišćenjem ISO slike i odabrati ISO sliku iz direktorijuma gde je smeštena prilikom preuzimanja. Potrebno je nekoliko minuta da se kreira instalacioni disk.

Kada se instalacija završi računara se ponovo pokrene i uradi se prijava sa korisničkim imenom i lozinkom. Posle toga potrebno je instalirati i podesiti DDNS klijenta, instalirati Webmin i otvoriti portove na ruteru.

6.2 DDNS podešavanja

Potrebno je otići na stranicu koja nudi besplatan dinamički DNS paket [10], registrovati se i izabrati besplatan hostname koji će se koristiti kao domen servera. Izabran je besplatni hostname (mitrovic.ddns.net) i ubuduće umesto IP adrese računara prilikom pristupa veb stranicama koristiće se ovaj hostname. Potrebno je na serverskom računaru koji koristi internet sa dinamičkom IP adresom instalirati program Dynamic Update Client. On radi tako što na određeni period proverava da nije došlo do promene IP adrese od strane provajdera.

Pre instaliranja Dynamic Update Client mora se instalirati build-essential paket koji sadrži program za instaliranje Dynamic Update Client programa (`sudo apt install build-essential -y`). Da bi se instalirao Dynamic Update Client potrebne su administratorske privilegije; to se postiže tako što se unese komanda „`sudo su`“ koja zahteva unos lozinke.

Dalji tok instalacije i podešavanja su prikazani u narednim koracima: postavlja se folder u koga se preuzima klijent (`cd /usr/local/src/`); preuzimanje započinje pomoću wget programa (`wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz`); raspakuje se preuzeti paket (`tar xf noip-duc-linux.tar.gz`); otvori se raspakovani folder (`cd noip-2.1.9-1/`).

Na kraju, instalira se Dynamic Update Client (`make install`). Kada se instalacija završi potrebno je sa administrator-

skim privilegijama kreirati konfiguracioni fajl (`/usr/local/bin/noip2 -C`).

Pre kreiranja konfiguracionog fajla potrebno je odgovoriti na pitanja o podacima sa noip (korisničko ime, lozinka) i na koliko minuta klijent treba da proverava da li je promenjena IP adresa (odabrano je 5 minuta). Da bi se Dynamic Update Client automatski pokretao prilikom pokretanja servera potrebno je kreirati noip skriptu u `init.d` folderu (`sudo nano /etc/init.d/noip`).

6.3 Webmin

Webmin je program koji omogućava udaljeni pristup i pojednostavljuje korišćenje računara sa Linux i Unix operativnim sistemima. Umesto da se koristi terminal za izmenu konfiguracionih fajlova, instaliranje programa, kreiranje naloga, podešavanje servera elektronske pošte, i slično, Webmin pruža mogućnost da se iz pregledača udaljenog računara upravlja serverom [11].

Da bi se instalirao Webmin pomoću apt programa, mora se izmeniti lista repozitorijuma sa kojih se mogu preuzimati i ažurirati paketi. To se radi pomoću nano editora (`sudo nano /etc/apt/sources.list`).

Takođe treba preuzeti i instalirati svoj GPG ključ sa kojim je repozitorijum potpisan:

1. `wget http://www.webmin.com/jcameron-key.asc`
2. `sudo apt-key add jcameron-key.asc apt-key add jcameron-key.asc`

Zatim se ažurira sistem da bi se uključio Webmin repozitorijum (`sudo apt update`), i instalira Webmin (`sudo apt install webmin -y`).

Kada se završi instalacija, na ekranu se ispisuje poruka da se može pristupiti serveru sa udaljenog računara (Slika 6).

```
Output
Webmin install complete. You can now login to
https://your_server_ip:10000 as root with your
root password, or as any user who can use `sudo`.
```

Slika 6: Poruka o uspešnoj instalaciji Webmin

6.4 Ostali koraci

Preostali koraci (Port Forwarding, dodavanje validnog sertifikata, prebacivanje veb stranica na server, prebacivanje baze podataka) koje treba odraditi kako bi se kompletirala softverska implementacija biće samo napomenuti usled nedostatka prostora.

Da bi se moglo pristupiti serveru sa interneta, potrebno je otvoriti portove na ruteru. Postupak otvaranja portova za veći modela rutera može se pronaći na sajtu Port Forward [12].

Nakon što se pristupi Webmin-u, u pregledaču se dobija poruka da konekcija sa Webmin stranicom nije sigurna (https precrtan i crvenom bojom), uz preporuku da se stranica zatvori, ili da se nastavi uz rizik da podaci budu kompromitovani.

To se dešava jer je Webmin podešen da koristi HTTPS [13] protokol, ali koristi nepouzdan sertifikat, potpisan samo svojim privatnim ključem.

U ovom trenutku na serveru nema podataka tako da se može nastaviti uz prihvatanje rizika. Koristiće se besplatan sertifikat kompanije Let's Encrypt koji je ponuđen kao opcija u okviru Webmin-a.

Stranice se prebacuju na server tako što se iz menija Others odabere File Manager, zatim se redom otvaraju folderi /var/www/html, iz menija odabere dodavanje datoteka u trenutni folder (Upload to current directory), odaberu datoteke i folderi sa lokalnog računara, sačekava da se datoteke opreme i nakon toga se vidi folder (arduino) u okviru menadžera datoteka.

Nakon što se u prethodnom koraku prebace sve Web stranice na server, potrebno je sa lokalnog računara preuzeti bazu podataka, a zatim je sačuvati na serveru. Sa lokalnog računara se pristupa stranici phpMyAdmin, zatim se odabere baza (arduino_db), iz menija izabere Export, i klikne na Go.

7. ZAKLJUČAK

U ovom radu prikazana je izrada funkcionalnog sistema zasnovanog na Arduino platformi koji vrši obradu podataka u realnom vremenu, i u zavisnosti od dobijenih vrednosti aktivira uređaje (preciznije, uključuje i isključuje navodnjavanje) i te vrednosti prikazuje krajnjem korisniku. Cilj ovog rada je da se pokaže da se primenom računarskih tehnologija mogu olakšati i automatizovati svakodnevni poslovi koje obavlja jedan voćar.

8. LITERATURA

- [1] M. Marković, N. Mitrović, N. Grujić, Đ. Mihailović, "Primena Arduino platforme na automatizaciju navodnjavanja voćnjaka -- Deo 1: Hardver", Info M 68/2018, str. 29-34
- [2] POST: [https://en.wikipedia.org/wiki/POST_\(HTTP\)](https://en.wikipedia.org/wiki/POST_(HTTP))
- [3] XAMPP: <https://en.wikipedia.org/wiki/XAMPP>
- [4] HTML Canvas Gauges v2.0: <https://github.com/Mikhus/canvas-gauges>

- [5] Google Line Chart: <https://developers.google.com/chart/interactive/docs/gallery/linechart>
- [6] MySQL: <http://www.oracle.com/technetwork/database/mysql/index.html>
- [7] PHP: <http://php.net/>
- [8] Zvanična stranica za preuzimanje Ubuntu Server OS: <https://www.ubuntu.com/download/server>
- [9] Program za kreiranje butabilnih USB diskova Rufus: <https://rufus.akeo.ie/>
- [10] Besplatan dinamički DNS paket: <https://www.noip.com/>
- [11] Webmin wiki: <https://doxfer.webmin.com/Webmin/Introduction>
- [12] Uputstvo za otvaranje portova: <https://portforward.com/>
- [13] HTTPS: <https://en.wikipedia.org/wiki/HTTPS>



Prof. Dr Miroslav Marković, Visoka škola strukovnih studija Aranđelovac (VŠAR)
Kontakt: miroslav.markovic@vsar.edu.rs
Oblast interesovanja: embedded sistemi, algoritmi i optimizacija



Nebojša Mitrović Visoka škola strukovnih studija Aranđelovac (VŠAR)
Kontakt: nebojsa.mitrovic@vsar.edu.rs
Oblast interesovanja: embedded sistemi, računarske mreže



Filip Vasić, Visoka škola strukovnih studija Aranđelovac (VŠAR)
Kontakt: nebojsa.grujic@vsar.edu.rs
Oblast interesovanja: programiranje, algoritmi



Prof. Dr Đorđe Mihailović, Visoka škola strukovnih studija Aranđelovac (VŠAR)
Kontakt: djordje.mihailovic@vsar.edu.rs
Oblast interesovanja: e-poslovanje, informacijski sistemi

