PAMETNA BAŠTA

# Projektni zadatak

Potrebno je projektovati pametnu baštu korišćenjem više senzora, aktuatora i Raspberry Pi mikroračunara. Prikazati projektovanje pametne bašte korišćenjem Visio i Fritzing programa. U viziju je potrebno nacrtati pametnu baštu i prikazati pozicije na kojima bi bili postavljeni senzori, aktuatori i Raspberry Pi. Slika treba da sadrži legendu korišćenih pametnih uređaja. Fizičko povezivanje uređaja treba prikazati korišćenjem Fritzing programa. Svaki deo fizičkog povezivanja prikazati legendom.

Definisati modele uređaja i njihov scenario rada u pametnoj bašti. Svaki pametni uređaj treba da ima opis specifikacije.

Prikazati i objasniti kod komentarima.

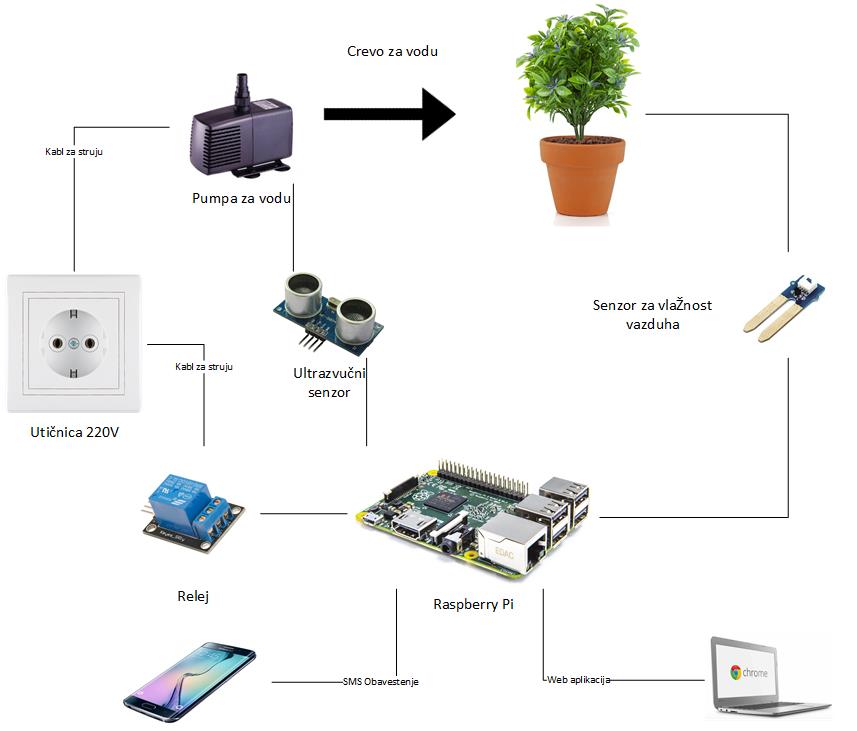
Video sa demonstracijom ovog primera možete pogledati na sledećem linku:

[https://www.youtube.com/watch?v=PD3OxoQnTkQ& feature=youtu.be](https://www.youtube.com/watch?v=PD3OxoQnTkQ&%09feature=youtu.be)

# Rešenje zadatka

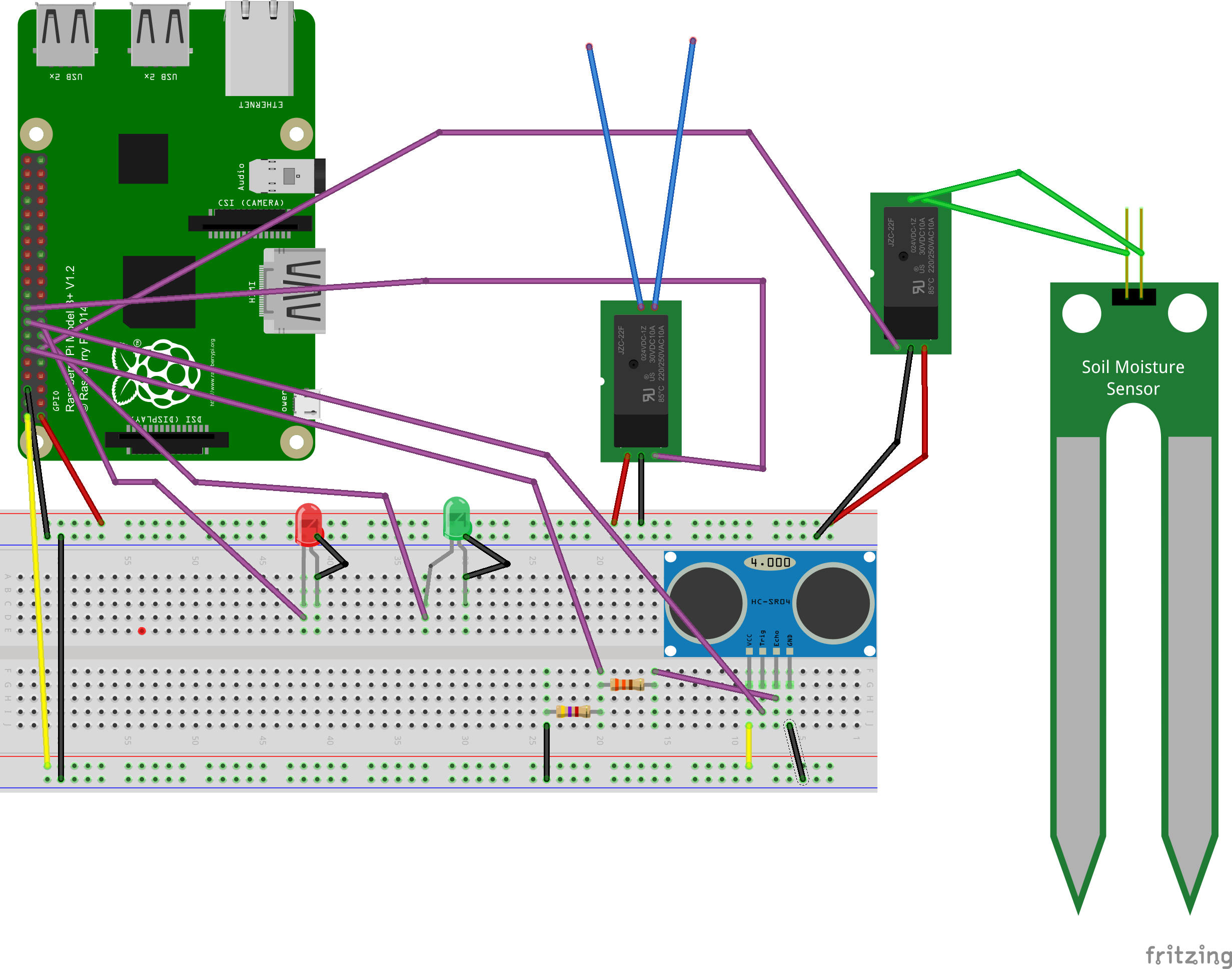
## Projektovanje

Na slici 1 prikazana bašta sa pozicijama pametnih uređaja.



Slika 1. Pametna bašta

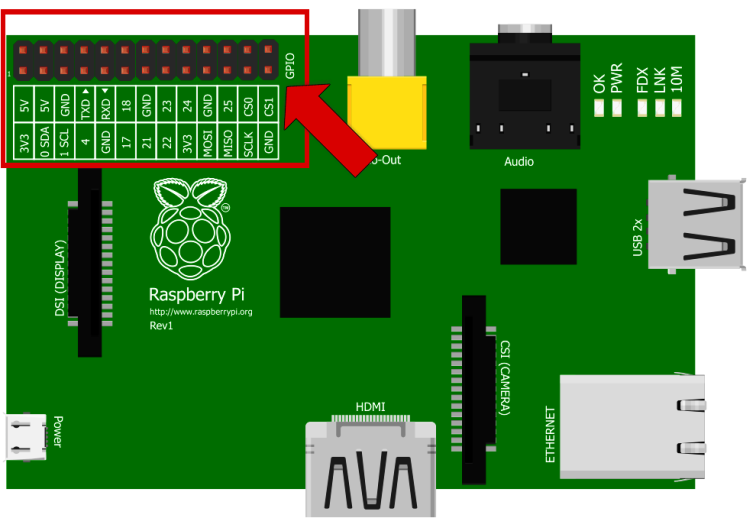
Fizičko povezivanje senzora i uređaja za pametnu baštu u Fritzing programu prikazano je na Slici 2.



Slika 2. Fizičko povezivanje senzora i uređaja

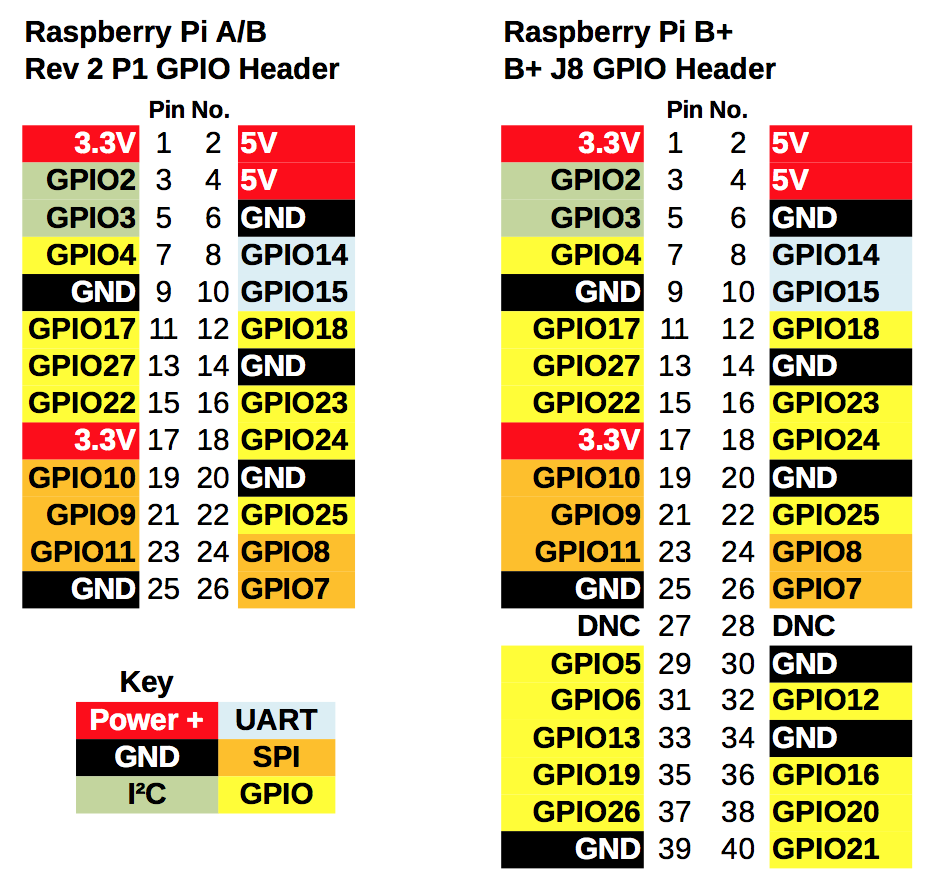
Crne linije na slici predstavljaju kablove koji se povezuju na ground izvod (negativni pol) na Raspberry Pi uređaju. Crvene linije na slici predstavljaju kablove koji se povezuju na izvod od 3.3 V na Raspberry Pi uređaju, žute linije predstavljaju kablove koji se povezuju na izvod od 5.0V na Raspberry Pi uređaju. Ljubičaste linije predstavljaju veze komponenata na protoploči sa odgovarajućim GPIO pinovima na Raspberry Pi uređaju. Plave linije predstavljaju povezivanje pumpe sa relejem.

Raspored pinova na Raspberry Pi B ili B+ modelu prikazani su na Slici 3.



Slika 3. Prikaz rasporeda pinova na Raspberry Pi B ili B+ modelu

Pin broj 1 je pin najbliži konektoru za napajanje (kod modela B i B+). Na slici 4 prikazana je šema rasporeda pinova.



Slika 4. Šema rasporeda pinova

### Pametni uređaji

Pametni uređaji potebni za realizaciju ovog primera dati su u tabeli 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Naziv komponente** | **Izgled komponente** | **Opis** | **Količina** |
| Raspberry Pi mikroračunar | 1914-01.jpg | Model **B** ili **B+** | 1 |
| Internet konekcija | 113 |  | 1 |
| Ultrazvučni senzor | 20130108065629!HC-SR04 | HC-SR04 | 1 |
| Senzor za vlažnost zemljišta | large_366_SoilMoisture1-450 | Sa LM393 comparator čipom | 1 |
| Pumpa za vodu | Aquarium-Water-Pump-001- | Atman AT-300 | 1 |
| Relej | $T2eC16N,!ykE9s7tvVK)BRpc21z9Wg--60_12 | 5V DC | 1 |
| Otpornici | https://nootropicdesign.com/ve/images/330_resistor.jpg | 5,1k Ω | 1 |
| Otpornici | https://nootropicdesign.com/ve/images/330_resistor.jpg | 330Ω | 1 |
| LED diode | http://www.greenprophet.com/wp-content/uploads/2012/09/LED-lights-health-hazard.jpeg | 3mm  boja po izboru | 2 |
| USB miš | http://www.zseries.in/electronics%20lab/cables/usb/pictures/usb%20mouse.jpg |  | 1 |
| USB tastatura | http://www.liquidware.com/system/0000/3480/Beagle_Mini_USB_Keyboard.jpg |  | 1 |
| Proto ploča | 64-02.jpg | Za povezivanje elektronskih elemenata | 1 |
| Kablovi | https://content.solarbotics.com/products/photos/03e0f1ccebb02b4dc5cc17e395d3049b/lrg/45040-dscn0624.jpg | Male-male | 2 |
| Kablovi | Untitled-2.png | Male-Female | 5 |
| MicroUSB punjač | http://www.world-of-accessories.co.uk/images/user/11-120313233630.jpg | 2A | 1 |
| HDMI kabl | http://skpang.co.uk/catalog/images/raspberrypi/hdmi_cable.jpg |  | 1 |

Tabela 1. Prikaz pametnih uređaja za realizaciju rešenja

## Scenario

Pametna bašta treba da omogući automatizovano zalivanje bašte, slanje obaveštenja o količini vode u kanisteru za zalivanje, kao i obaveštenja o vremenu svakog zalivanja putem SMS-a. Za realizaciju ovog projekta potrebo je 2 senzora, pumpa za vodu, relej, led diode, kao i web aplikacija sa SMS servisom. Senzor za vlažnost zemljišta bi se nalazio zaboden u zemljistu koje je potrebno zalivati, onog trenutka kada vlažnost zemljišta padne ispod normalne vlažnosti, aktivirao bi se relej koji pušta struju do pumpe za vodu i započinje zalivanje zemljišta. Ultrazvučni senzor sve vreme meri količinu vode koja se nalazi u kanisteru i ispisuje podatke na web aplikaciji. Prilikom svakog zalivanja bašte, vlasnik bi dobio obaveštenje putem SMS-a da je došlo do zalivanja. Takođe ukoliko količina vode u kanisteru padne ispod 20%, vlasnik će dobiti obaveštenje da treba da dopuni kanister jer je ostalo malo vode. (Ukoliko to ne uradi, dobijaće SMS obaveštenja na svakih 5h, da treba da dopuni kanister sa vodom.) Nakon zalivanja, senzor za vlažnost zemljišta će detektovati da je postignuta normalna vlažnost zemljišta a zatim će ugasiti relej koji će zaustaviti rad pumpe za vodu.

## Prikaz rešenja

Za realizaciju primera pametno taksi vozilo potrebno je koristiti bilo koji tekstualni editor (Notepad, Notepad++, Sublime ili slično). Kod prikazan u Prilogu 1 kucan je u Sublime-u.

Prilog 1. Prikaz koda **basta.py**

|  |
| --- |
| #!/usr/bin/python  #importovanje potrebnih biblioteka  **from subprocess import call**  **import time**  **import urllib**  **import RPi.GPIO as GPIO**  **#postavljanje pinova**  **GPIO.setmode(GPIO.BCM)**  **GPIO.cleanup()**  **GPIO.setwarnings(True)**  **GPIO.setup(17,GPIO.IN)** #senzor za vlaznost zemljista  **GPIO.setup(27,GPIO.OUT)** #zelena LED dioda  **GPIO.setup(22,GPIO.OUT) #**crvena LED dioda  **GPIO.setup (23, GPIO.OUT)** #trigger  **GPIO.setup (18,GPIO.IN)** #echo  **GPIO.output(22,GPIO.LOW)** #crvena LED dioda OFF  **GPIO.output(27,GPIO.HIGH)** #zelena LED dioda ON  **status = None**  **while True:**  **print "Provera kanistera"**  **GPIO.output (23,GPIO.LOW)**  **GPIO.output(23,GPIO.HIGH)** #slanje kratkog signala preko triggera  **time.sleep(0.00001)**  **GPIO.output(23,GPIO.LOW)**  **while GPIO.input (18) == 0:** #racunanje proteklog vremena  **pass**  **start = time.time()**  **while GPIO.input (18) == 1:**  **pass**  **stop = time.time()**  **voda = round(((30-((stop-start) \* 17000))/20)\*100,2**) #formula za racunanje % kolicine vode u kanisteru  **print voda**  **url= 'http://www.halotaxi.rs/test/testupisivodu.php?voda={0}'.format(voda) #slanje nivoa vode**  **urllib.urlopen(url)**  **print "Ocitavam vlaznost zemljista"**  **citac = GPIO.input(17)** #citanje vlaznosti zemljista  **print citac**  **if citac==0:**  **GPIO.output(22,GPIO.LOW)**  **GPIO.output(27,GPIO.HIGH)**  **if status == 1:**  **print "Poslato je da je vlaznost zemljista normalna!!"**  **else:**  **print "Vlaznost zemljista je normalna!"**  **urllib.urlopen('http://www.halotaxi.rs/test/testupisibastu.php?vlaznost=0')** #obavestenje o vlaznosti  **status=1**  **else:**  **GPIO.output(22,GPIO.HIGH)**  **GPIO.output(27,GPIO.LOW)**  **call(["python", "relejon.py"])** #pokretanje eksternog python fajla koji ukljucuje relej tj. pumpu za vodu  **print "Pokrecem pumpu..."**  **print "Zalivam cvece"**  **time.sleep(10)**  **call(["python", "relejoff.py"])** #pokretanje eksternog python fajla koji iskljucuje relej tj. pumpu za vodu  **print "Cekam da zemljiste upije vodu"**  **time.sleep(30)** #cekanje od 30 sekundi da zemlja upije vodu  **call(["python", "relejon.py"])**  **print "Pokrecem pumpu..."**  **print "Zalivam cvece"**  **time.sleep(10)**  **call(["python", "relejoff.py"])**  **print "Zalivanje zavrseno"**  **if status==0:**  **print "Poslato je da treba zalivanje!"**  **else:**  **print "PAZNJA: Vlaznost zemljista ispod normale, potrebno zalivanje!!!"**  **urllib.urlopen('http://www.halotaxi.rs/test/testupisibastu.php?vlaznost=1') #obavestenje o vlaznosti**  **status=0**  **time.sleep(1)** |

Prilog 2. Prikaz koda **relejon.py**

|  |
| --- |
| #!/usr/bin/python  #Importovanje potrebnih biblioteka  **import RPi.GPIO as GPIO**  **import time**  # Koriscenje BCM sistema  **GPIO.setmode(GPIO.BCM)**  # Lista pinova koje koristimo  **pinList = [24]**  # postavljanje pinova na LOW  **for i in pinList:**  **GPIO.setwarnings(False)**  **GPIO.setup(i, GPIO.OUT)**  **GPIO.output(i, GPIO.HIGH)**  **def trigger() :**  **for i in pinList:**  **GPIO.output(i, GPIO.LOW)**  # GPIO.cleanup()  **break**    **try:**  **trigger()**      **except KeyboardInterrupt:**  **print " Quit"**  # Resetovanje GPIO podesavanja  **GPIO.cleanup()** |

Prilog 3. Prikaz koda **relejoff.py**

|  |
| --- |
| #!/usr/bin/python  #Importovanje potrebnih biblioteka  **import RPi.GPIO as GPIO**  **import time**  # Koriscenje BCM sistema  **GPIO.setmode(GPIO.BCM)**  # Lista pinova koje koristimo  **pinList = [24]**  # postavljanje pinova na LOW  **for i in pinList:**  **GPIO.setwarnings(False)**  **GPIO.setup(i, GPIO.OUT)**  **GPIO.output(i, GPIO.HIGH)**  **def trigger() :**  **for i in pinList:**  **GPIO.output(i, GPIO.HIGH)**  # GPIO.cleanup()  **break**    **try:**  **trigger()**      **except KeyboardInterrupt:**  **print " Quit"**  # Resetovanje GPIO podesavanja  **GPIO.cleanup()** |

Nakon završetka kucanja koda, fajl je potrebno sačuvati pod nazivom ***basta.py.*** Lokacija za čuvanje fajlova na računaru može biti bilo koja. U ovom primeru skripte su sačuvane u folderu Desktop na Raspian OS-u (Slika 5).

Capture.JPG

Slika 5. Promena direktorijuma

Nakon kucanja ove komande, promeniće se direktorijum, a putanja će izgledati kao na slici 6.

Capture2.JPG

Slika 6. Prikaz promene direktorijuma na Raspbianu

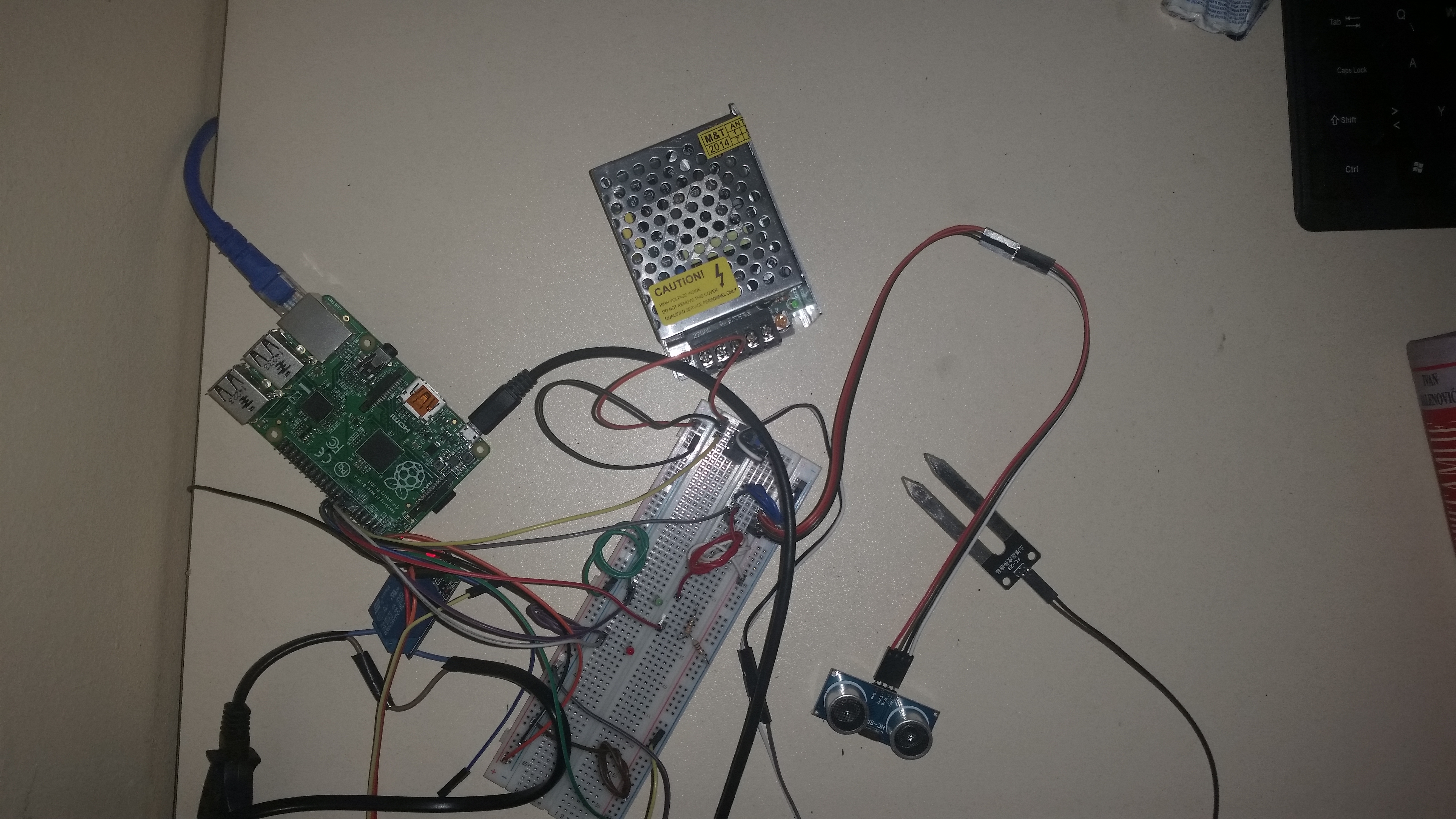
Za pokretanje programa kuca se komanda prikazana na slici 7.

bastapy

Slika 7. Pokretanje programa

Nakon kucanja ove komande započeće izvršavanje programa i očitavanje senzora za vlažnost zemljista kao i ultrazvučnog senzora sa određenim ispisom na ekranu. Sledi testiranje uređaja. Prekidanje izvršavanja programa se vrši komandom **Ctrl + c**.

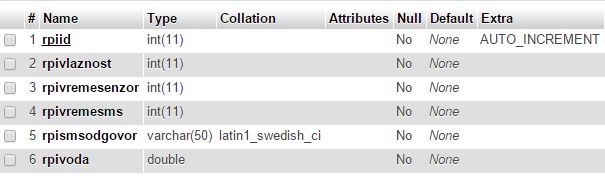
Prikaz povezanog RPi B+ povezanog na breadboard senzorima dat je na slici 8.



Slika 8. Prikaz povezanih uređaja

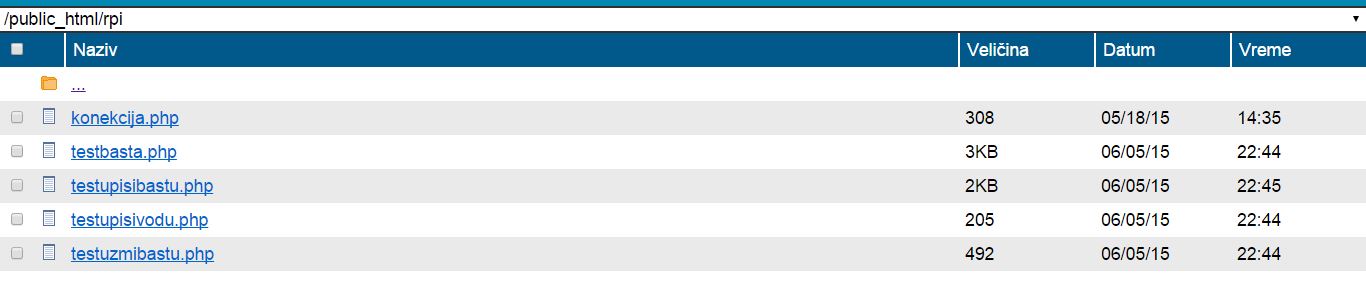
Za upravljanje realizovanim rešenjem razvijen je veb servis i veb aplikacija.

Da bi skripte radile, neophodno je pre pravilnog funkcionisanja programa napraviti bazu podataka i izvršiti povezivanje sa istom. Baza podataka treba da sadrži tabelu pod nazivom rpi i takođe da ima 6 kolonA, **rpiid**, **rpivlaznost, rpivremesenzor, rpivremesms, rpismsodgovor i rpivoda**. Potrebno je napraviti MySQL bazu na studentskom elab hostingu i kreirati prvi red u bazi. Ime baze je **rpi** (slika 9).

****

Slika 9. Kreiranje baze podataka

Potrebno je kreirane php skripte (prilog 4, prilog 5, prilog 6, prilog 7 i prilog 8) staviti na elab hosting (slika 10). Napomena: Moguće je koristiti bilo koj hosting.

****

Slika 10. Smeštanje php skripti na elab hosting

Prilog 4. **konekcija.php**

|  |
| --- |
| Ovaj fajl služi za povezivanje sa bazom podataka.  **<?php**  **$db\_server = "localhost";**  **$db\_db= "rpi";**  **$db\_user = "root";**  **$db\_pass = "";**  **$mysqli = new mysqli($db\_server, $db\_user, $db\_pass, $db\_db);**  **if ($mysqli->connect\_errno) {**  **printf("Konekcija neuspešna: %s\n", $mysqli->connect\_error);**  **exit();**  **}**  **$mysqli->set\_charset("utf8");**  **?>** |

Prilog 5. **testuzmibastu.php**

|  |
| --- |
| Ovaj fajl nam služi da isčitamo trenutno stanje senzora.  **<?php**  **include "konekcija.php";**  //uzimanje podataka iz baze  **$sql="SELECT \* FROM rpi";**  **$q=$mysqli->query($sql);**  **$red=$q->fetch\_object();**  //konvertovanje vremena  **$sms= date('d/m/Y - H:i:s', $red->rpivremesms);**  **$senzor= date('d/m/Y - H:i:s', $red->rpivremesenzor);**  //smestanje promenljivih u niz  **$odgovor = array(**  **'status' => $red->rpivlaznost,**  **'sms' => $sms,**  **'senzor' => $senzor,**  **'smsodgovor' => $red->rpismsodgovor,**  **'voda'=> $red->rpivoda**  **);**  //pravljenje JSON-a  **echo json\_encode($odgovor);**  **?>** |

Prilog 6. **testupisibastu.php**

|  |
| --- |
| **<?php**  **$vreme=time();**  **include "konekcija.php";**  **if(!isset($\_GET['vlaznost'])){** //provera prosledjenih parametara  **echo "Greska!";**  **} else {**  **$vlaznost=$\_GET['vlaznost'];**  **$sql="UPDATE rpi SET rpivlaznost='$vlaznost', rpivremesenzor='$vreme' WHERE rpiid=1";** //upis stanja senzora  **$q=$mysqli->query($sql);**  **if ($vlaznost==1){ //provera stanja senzora**  **$sql2="SELECT \* FROM rpi WHERE rpiid=1";**  **$q2=$mysqli->query($sql2);**  **$red2=$q2->fetch\_object();**  **$nivovode=$red2->rpivoda;**  **$vremedetekcije = date('d/m/Y - H:i:s', $red2->rpivremesenzor);**  **echo $vremedetekcije;**  **$vremezabrane=$red2->rpivremesms+18000;**  //zabrana slanja poruke na manje od 5 sati  **if ($vreme > $vremezabrane) {**      **$url = "https://student.elab.fon.bg.ac.rs/api/sms/v1";**  **$broj='0123456789';**  **$tekst='Cvece je zaliveno! Vreme zalivanja je: '.$vremedetekcije.'. Nivo vode u kanisteru je '.$nivovode.' %.';**  **if ($nivovode<20) $tekst='Cvece je zaliveno! Nivo vode u kanisteru je '.$nivovode.' %. Dopunite kanister sa vodom!!!';**  //SMS SERVIS  **$data = array(**  **'broj' => $broj,**  **'tekst' => $tekst**  **);**  **$data1=json\_encode($data);**  **$curl = curl\_init($url);**  **$headers = array(**  **'X-ELAB-SMS-Kljuc: ELAB SMS KLJUC',**  **'Content-type: application/json',**  **);**  **curl\_setopt($curl, CURLOPT\_HTTPHEADER, $headers);**  **curl\_setopt($curl, CURLOPT\_HEADER, false);**  **curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, false);**  **curl\_setopt($curl, CURLOPT\_POST, true);**  **curl\_setopt($curl, CURLOPT\_POSTFIELDS, $data1);**  **$json\_response = curl\_exec($curl);**  **$odgovor = json\_decode($json\_response);**  **if (isset($odgovor->greska)){**  **$odgovorbaza = $odgovor->greska;**  **} else {**  **$odgovorbaza = $odgovor->obavestenje;**  **}**  **curl\_close($curl);**  **$vremesms = time();**    **$sql3="UPDATE rpi SET rpivremesms='$vremesms', rpismsodgovor='$odgovorbaza' WHERE rpiid=1";**  **$q3=$mysqli->query($sql3);**    **}**  **}**  **}**  **?>** |

Prilog 7. **testupisivodu.php**

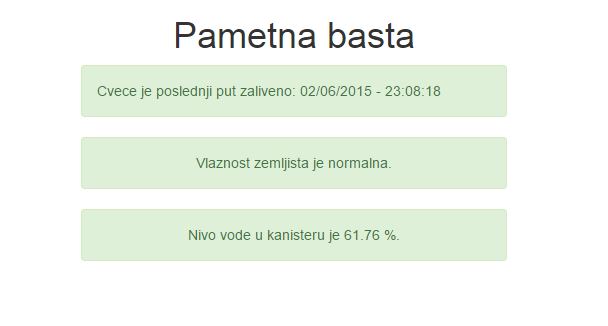
|  |
| --- |
| Ovaj fajl nam služi da isčitamo trenutno stanje nivoa vode u kanisteru.  **<?php**  **include "konekcija.php";**  **if(!isset($\_GET['voda'])){** //provera prosledjenih parametara  **echo "Greska!";**  **}else {**  **$voda=$\_GET['voda'];**  **$sql="UPDATE rpi SET rpivoda='$voda' WHERE rpiid=1**"; //upis prosledjenog parametra u bazu  **$q=$mysqli->query($sql);**  **}**  **?>** |

Prilog 8. Prikaz koda web aplikacije **testbasta.php**

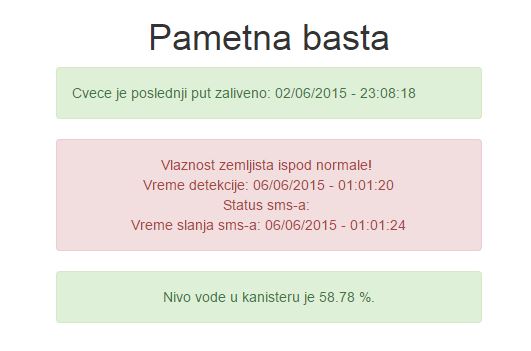
|  |
| --- |
| **<!DOCTYPE html>**  **<html>**  **<head>**  <!-- Stilizovanje -->  **<style type="text/css">**  **#voda {**  **text-align: center;**  **}**  **#detektcija{**  **text-align: center;**  **}**  **h1{**  **text-align: center;**  **}**  **</style>**  **<title>Pametna basta</title>**  <!-- Potrebni fajlovi -->  **<?php include 'skripte/funkcije.php'; ?>**  **<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">**  **<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>**  **<meta charset="utf-8">**  **<meta http-equiv="X-UA-Compatible" content="IE=edge">**  **<meta name="viewport" content="width=device-width, initial-scale=1">**  <!-- Skripta koja omogucava proveru statusa bez potrebe za osvezavanjem stranice -->  **<script type="text/javascript">**  **function proveriStatus(){**  **var timer = setInterval(function(){**  **$.ajax({**  **url: 'testuzmibastu.php',**  **dataType : 'json',**  **success: function(response) {**  **var status = response['status'];**  **var voda = response['voda'];**  **$('#voda').html('<div class="alert alert-success">Nivo vode u kanisteru je ' +voda+ ' %.</div>');**  **if (status==0) {**  **$('#detektcija').html('<div class="alert alert-success">Vlaznost zemljista je normalna.</div>');**    **}**  **if (status==1){**  **$('#detektcija').html('<div class="alert alert-danger">Vlaznost zemljista ispod normale! <br> Vreme detekcije: '+response['senzor']+'<br> Status sms-a: '+ response['smsodgovor']+' <br> Vreme slanja sms-a: '+response['sms']+'</div>'); // buffers automatically when created**  **document.getElementById('audio').play();**    **clearInterval(timer);**  **setTimeout(function(){**  **proveriStatus();**  **},1000);**  **}**    **}**  **})}, 1000)} ;**  **</script>**  <!-- Upit koji vraca poslednju detekciju senzora -->  **<?php**  **$sql="SELECT \* FROM rpi";**  **$q=$mysqli->query($sql);**  **$red=$q->fetch\_object();**  **$poslednjadetekcija = date('d/m/Y - H:i:s', $red->rpivremesenzor);**  **?>**  **</head>**  **<body onload="proveriStatus();">**  **<div class="container-fluid">**  **<div class="row">**  **<div class="col-lg-4 col-sm-4 col-xs-12"></div>**  **<div class="col-lg-4 col-sm-4 col-xs-12">**  **<h1>Pametna basta</h1>**  **<audio id="audio" controls="controls" hidden>**  **<source src="mp3/pozar.mp3" type="audio/mp3">**  **Your browser does not support the audio element.**  **</audio>**  **<div class="alert alert-success" id="poslednjadetekcija">Cvece je poslednji put zaliveno: <?php echo $poslednjadetekcija; ?></div>**  **<div id="detektcija"></div>**  **<div id="voda"></div>**  **</div>**  **<div class="col-lg-4 col-sm-4 col-xs-12"></div>**  **</div>**  **</div>**  **<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>**  **</body>**  **</html>** |

Web aplikacija pametne baste sastoji se iz 3 zelene kartice. Prva kartica omogucava uvid u poslednje zalivanje baste. Pored toga nalazi se i kartica gde se moze videti da li je vlaznost zemiljsta zadovoljavajuca. Ukoliko je vlaznost zemljista, druga kartica menja boju u crvenu i pruža podatke o detekciji (status, vreme, sms). Treca kartica u svakom trenutku omogucava uvid u nivo vode u kanisteru. Pored web aplikacije nalazi se i SMS servis koji obavestava korisnika svaki put kada je vlažnost zemljišta pala ispod normale tj. kad je cveće zaliveno. U SMS poruci se nalazi status senzora, vreme zalivanja kao i nivo vode u kanisteru. Ukoliko je nivo vode u kanisteru ispod 20%, u poruci se takodje ispisuje upozorenje da se kanister dopuni.

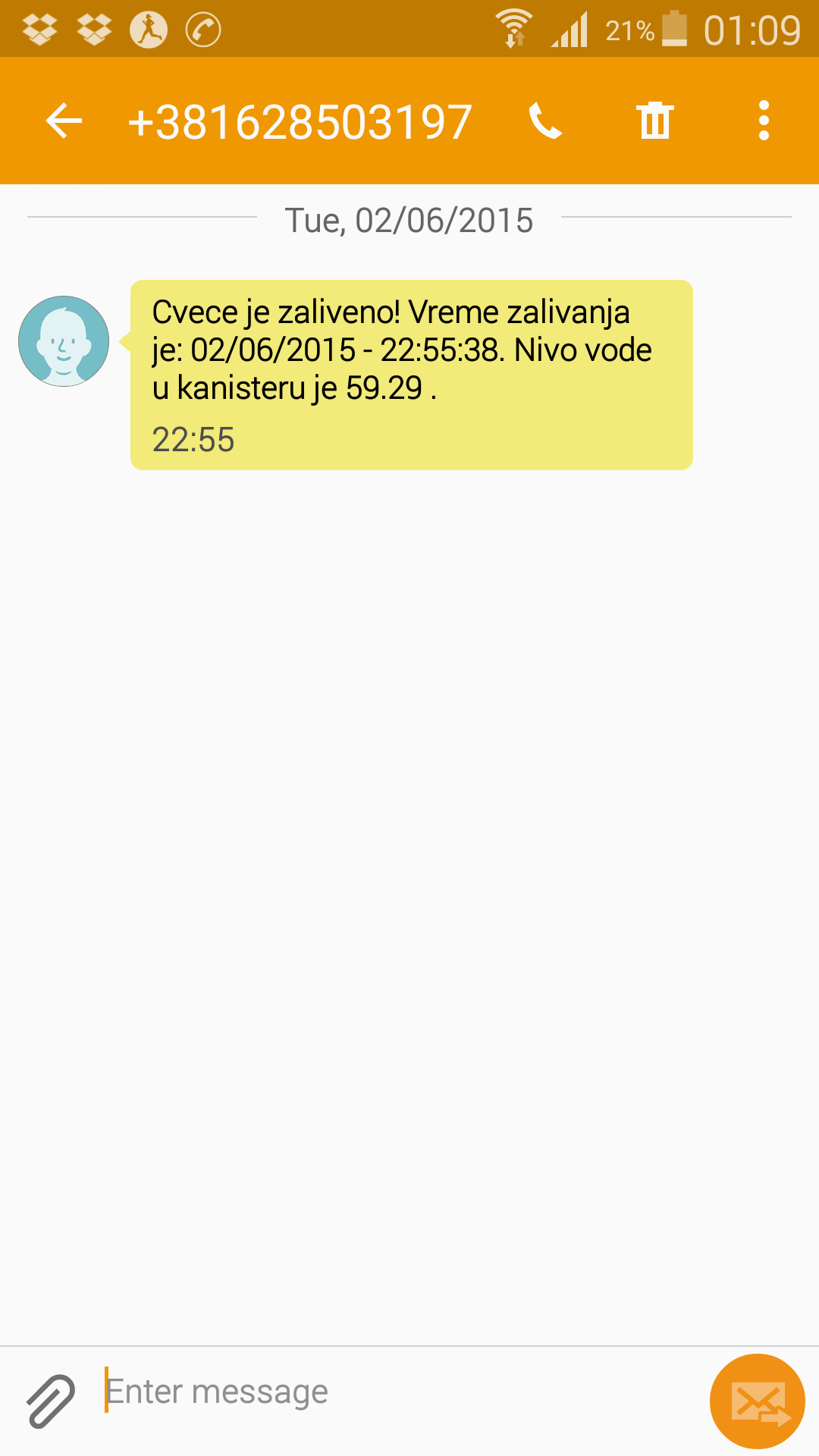
Prikaz veb aplikacije za upravljanje pametnim vozilom dat je na slici 10.



Slika 10. Prikaz veb aplikacije gde je vlažnost zemljista normalna



Slika 11. Prikaz veb aplikacije gde je vlažnost zemljista ispod normale



Slika 12. Prikaz sms poruke nakon zalivanja cveća