Programiranje uređaja za Internet stvari

Hager, Mislav

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:360243

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-20**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

MISLAV HAGER

PROGRAMIRANJE UREĐAJA ZA INTERNET STVARI

Završni rad

Pula, rujan 2016.

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

MISLAV HAGER

PROGRAMIRANJE UREĐAJA ZA INTERNET STVARI

Završni rad

JMBAG: 0303046026, redoviti student
Studijski smjer: Informatika

Predmet: Napredne tehnike programiranja
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Tihomir Orehovački

Pula, rujan 2016.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Mislav Hager, kandidat za prvostupnika Informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

U Puli, _____.

Student

_____

IZJAVA

o korištenju autorskog djela

Ja, Mislav Hager dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom Programiranje uređaja za internet stvari koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____.

Potpis

_____

Sadržaj:

1. Uvod 1
2. Internet stvari 2
2.1 Funkcionalni pogled Interneta stvari 2
3. Raspberry Pi 4
3.1 Software 4
3.2 Sklopovlje 5
3.3 Pokretanje računala i namještanje operativnog sustava 8
4. Podešavanje DHT22/AM2302 senzora 11
4.1 Instaliranje odgovarajuće Python biblioteke i iščtavanje podataka 13
5. Ažuriranje podataka online pomoću ThingSpeak platforme 15
6. Blynk - Android platforma za Internet stvari 22
7. Zaključak 29
Literatura 30
Popis slika 31
Popis kodova 31
Sažetak 32
Summary 33

1. UVOD

U današnje vrijeme, broj korisnika Interneta kontinuirano se povećava te je život bez njega postao nezamisliv. Internet je, između ostaloga, omogućio brži i efikasniji prijenos podataka te ne samo povezivanje ljudi već i stvari (uređaja). Danas je sve češći pojam Internet stvari koji u svojoj definiciji opisuje povezanost uređaja i komunikaciju između njih. Umreženi uređaji i senzori postali su dio naše svakodnevnice, a samim time može se reći da po tom pitanju postaju dio novog evolucijskog koraka u razvoju Interneta, postaju dio Interneta stvari.

Glavna zamisao rada je bila osposobiti senzor i uspostaviti komunikaciju sa bilo kojim mikroprocesorskim računalom i uz upotrebu do sad stečenog znanja ostvariti legitiman primjer poimanja Interneta stvari. Kao najbolje alternativno rješenje za izradu rada uzeo sam Raspberry Pi 2 Model B. Raspberry Pi se koristi za učenje programiranja, elektronike, robotike itd. Platforma se stalno razvija i napeduje, od hardwarea do softwarea. Velika je pomoć što ima dosta snažnu bazu podataka online i podršku za samostalan rad. Nedostaci su stvarno minorni u odnosu na sve prednosti i mogućnosti koje pruža. DHT22/AM2302 senzor za mjerenje temperature i vlažnosti zraka koji sam također zajedno s Rapberry Pi-em uzeo kao jedno od najboljih mogućih rješenja za izradu rada, poslužit će kao primjer „stvari“ koja će biti isprogramirana u kontekstu Interneta stvari.

Kroz ovaj rad opisati ću puku koncepciju Interneta stvari. Prvo kroz uvod vidjeti što je to Internet stvari i njegov funkcionalni pogled. Zatim opisati mini računalo Raspberry Pi te vidjeti njegove funkcionalnosti i komponente, opisati sklopovlje i vidjeti korake u podizanju njegova sustava. Također opisati i senzor te njegovo osposobljavanje i testiranje. Nakon osposobljenog senzora vidjeti kako se uz pomoć Python koda podaci ažuriraju i grafički prikazuju na web servisu. Naravno i onda za kraj proći kroz još jednu vjerodostojnu platformu Interneta stvari i pomoću JavaScript-a ostvariti povezanost uređaja i putem Androida.

2. Uvod u Internet stvari

Internet stvari (eng. Internet of Things, skr. IoT) opisuje povezanost uređaja - bilo kojeg uređaja - na internet koristeći uklopljene sustave i senzore da komuniciraju, skupljaju i razmjenjuju podatke između sebe. Sa Internetom stvari, svijet je širom otvoren, nudeći beskonačnu virtualnu povezanost bilo to kod kuće ili na poslu. Tehnologija je danas umetnuta u svakodnevne fizičke objekte (uređaje). Te objekte možemo naučiti da reagiraju na našu prisutnost, pokret, glasovne naredbe, pokret oka pa čak i na autonomna fiziološka ponašanja kao što su otkucaiji srca.
IoT kombinira povezanost sa senzorima, uređajima i ljudima, te uključuje interakciju između čovjeka i stroja, softwera i hardwera. Sa napredcima u umjetnoj inteliganciji i strojnom učenju, ova vrsta interakcije može uključiti uređaje da anticipiraju, reagiraju, odgovarjau i poboljšavaju fizički svijet. Tako je primjerice već danas moguće da hladnjak pošalje obavijest o namirnicama koje nedostaju kako bismo ih kupili prilikom povratka kući. Ili recimo putem svog pametnog telefona upravljati klimatizacijom kod kuće, video nadzorom nadgledati prostor sa udaljene lokacije i sl.

2.1. Funkcionalni pogled Interneta stvari

IoT koncept temelji se na jedinstveno identificiranim objektima. U internet strukturi i IoT rješenja sastavljena su od komponenti kao što su:

- Modul za interakciju sa lokalim IoT uređajima (primjerice mobilni telefon koji se nalazi u neposrednoj blizini uređaja te se stoga može spojiti putem bežičnog sučelja).
- Modul za lokalnu analizu nalaza, oni trebaju IoT uređaje.
- Modul za komunikaciju s udaljenim IoT uređajem izravno putem interneta ili preko posrednika (proxy). Ovaj modul je odgovoran za dobivanje nalaza i slanje rezultata na udaljenim poslužiteljima za analizu i skladištenje.
- Modul za analizu i obradu određenih podataka aplikacije. On radi na aplikacijskom poslužitelju i služi svim kupcima. Uzima zahtjeve od mobilnih i web klijenata. Po primitku relavantnih IoT odgovora, početi će pronalaženje odgovarajućih algoritama za obradu podataka i generiranje rezultata, koji je tada predstavljen korisniku.

- Modul za povezivanje IoT informacije u poslovnim procesima. Ovaj modul će dobiti na važnosti s povećanom uporabom IoT podataka kao važnog čimbenika u svakodnevnom poslovanju ili poslovnim strategijama.
- Korisnička sučelja (mobilna ili web): vizualni prikaz mjerenih podataka u određenom kontekstu (npr na karti) i komunikacija s korisnicima [1].

Važno je naglasiti da jedan od ključnih faktora za uspjeh IoT-a je odmaknuti se od zatvorenih sustava prema više otvorenim sustavima, baziranima na temelju otvorenog API-a (eng. *Application Programming Interface*) i standardnih protokola na različitim razinama sustava. Velik broj aplikacija je dostupno na tržištu, to pridonosi uspjehu pametnih telefona u industriji. Razvoj tako velikih broja aplikacija za pametne telefone je moguće zbog velikog broja programera, prvenstveno zbog uključivanja zajednice developera u cjelinu. Slično tome, što također za IoT potrebno je uspostaviti ekosistem koji će definirati otvorene API za programere i nuditi odgovarajuće kanale za isporuku novih aplikacija. Takva API sučelja imaju određenu važnost na razini modula za analizu i obradu određenih aplikacijskih podataka, tako da se dopušta programerima da utječu na osnovnu komunikacijsku infrastrukturu, te koristite i kombiniraju informacije generirane različitim IoT uređajima da proizvode nove, dodane vrijednosti [1].

3. Raspberry Pi

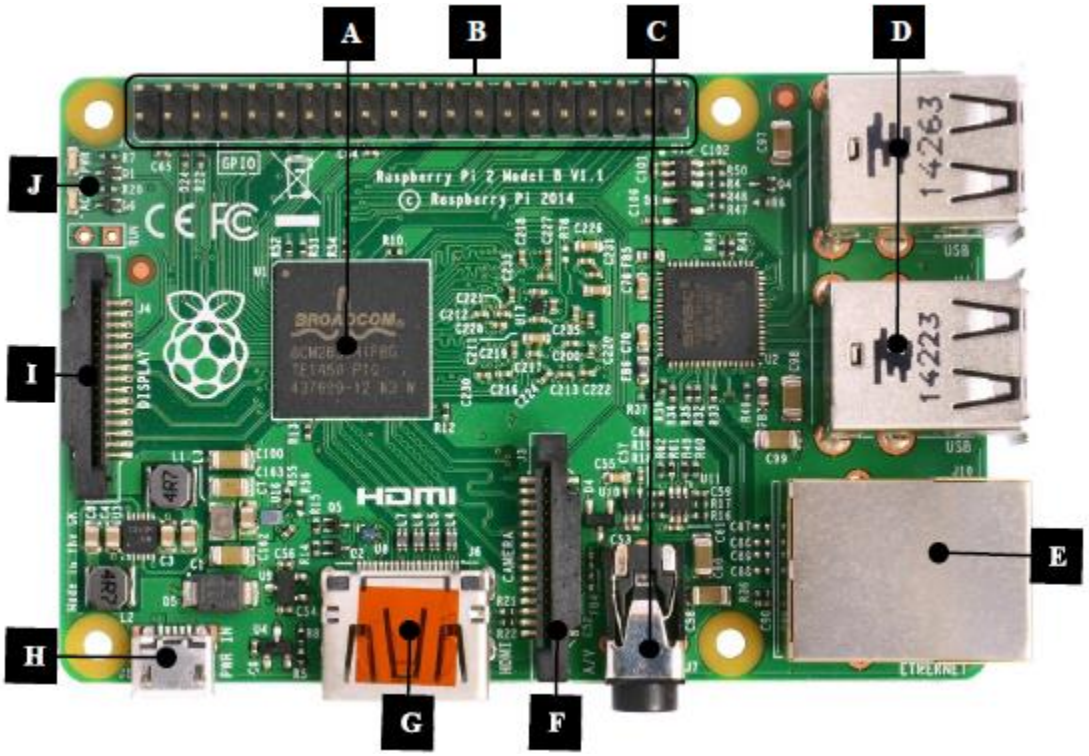
Raspberry Pi (skr. RPi) je računalo razvijeno u Cambridgeshireu u Velikoj Britaniji od strane Raspberry Pi Foundation. Raspberry Pi proizveden je s namjerom promicanja osnovnih računalnih znanosti u školama i kako bi mlade ljude potaknuo na bavljenje tehnologijom [2]. Raspberry Pi Foundation je dobrotvorna organizacija osnovana 2008. godine. Cilj organizacije i proizvođača bio je da uređaj bude nešto jeftiniji. Iz tog razloga određena je cijena od 25 dolara za najslabiji model. Za nešto jači model cijena iznosi otprilike 35 dolara. Prvi primjeri isporučeni su u veljači 2012. godine. Od samog početka prodaje pa do danas, Raspberry Pi je doživio izuzetnu popularnost.

3.1. Software

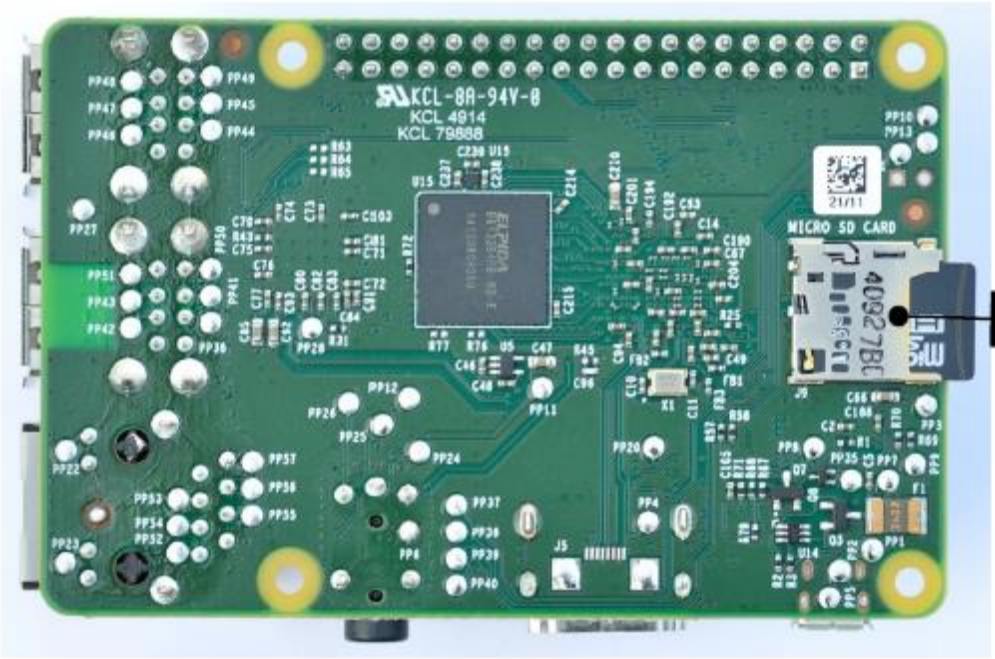
Za kontrolu resursa Raspberry Pi najčešće koristi Linux operativni sustav, a preporuča se Raspbian OS koji također koristim u izradi ovog završnog rada. Bazira se na Debian 7 „Eheezy“ Linux distribuciji, koja je dizajnirana za Raspberry Pi. Raspbian je vrlo popularan zbog njegovog jednostavnog načina održavanja te brzine i lakoće korištenja. Kao glavna prednost ovog OS-a ističe se mogućnost odabira obrazovnih materijala ili projekata koji su sadržani unutar distribucije. Samim time se omogućuje korisnicima da lako nauče osnove programiranja, uz mogućnost korištenja mnogih dodataka za hardver [3].
Ostale verzije OS-a su Noobs, Ubuntu Mate, Osmc, Openelec, Pinet. Dakako postoji i verzija Windows 10 operativnog sustava razvijenog za Raspberry Pi, a to je Windows 10 IOT Core. On se može pokrenuti samo na Raspberry Pi 2 model B i Raspberry Pi 3 model B.

Raspberry Pi ne dolazi s predinstaliranim operativnim sustavom, njega je potrebno instalirati pomoću odgovarajućeg programa na SD (eng. *Secure Digital*) karticu. Minimalna veličina SD kartice je 2GB, ali se preporuča veličina od 4GB ili više [4].

3.2. Sklopovlje



Slika 1. Gornja strana računala RPi 2.



Slika 2. Donja strana računala RPi 2.

A: Procesor

Raspberry Pi 2 koristi isti tip procesora kakav se može naći u uređajima kao što su Iphone 3G i Kindle 2, tako da su njegove mogućnosti usporedive s ovim uređajima. ARM-ovi čipovi općenito se rade u velikom broju arhitektura i s različitim jezgrama prilagođenim za pružanje različitih mogućnosti u različitim cjenovnim rangovima. ARM Cortex-A7 čip je četverojezgreni, 32-bitni, brzine 900 MHz, sustav na čipu, temeljen na ARMv7-A arhitekturi. Manji je, jednostavniji i energetski učinkovitiji u odnosu na prethodnike [5]. Uz to, Raspberry Pi 2 dolazi s 1 GB radne memorije.

B: GPIO (eng. *general-purpose input/ooutput*)
Model Raspberry Pi 2 ima 2 × 20 izvoda, što je standard još od modela B+. Ovi izvodi su fizičko sučelje između Raspberry-a i okoline, tj. omogućuju Raspberry-ju interakciju s fizičkim svijetom upravljanjem ulazima i izlazima. Najjednostavnije, može ih se promatrati kao prekidače koji se mogu uključivati i isključivati, u slučaju ulaza ili koje Raspberry može uključivati ili isključivati, što je slučaj kod izlaza. Ulazi ne moraju uvijek dolaziti od fizičkog prekidača, nego se mogu prikupljati sa senzora ili signala koji potječe s drugog računala. Izlaz također može biti bilo što, od paljenja LED diode do slanja signala ili podataka drugom uređaju. U slučaju da je Raspberry spojen na mrežu, moguće je iz daljine upravljati uređajima koji su priključeni na njega te dobivati podatke s tih uređaja. 26 od 40 izvoda GPIO-a su izvodi za ulaz i izlaz, dok ostali izvodi daju energiju i uzemljenje. U nastavku će biti objašnjeno kako koristiti izvode GPIO-a za očitavanje podataka sa senzora.

C: Kombinirani analogni video i audio izlaz
Na modelu Raspberry Pi 2, analogni audio i video izlazi su dostupni na standardnom 3.5 mm 4-polnom konektoru.

D: Vanjski USB priključci
Raspberry Pi 2 ima četiri USB 2.0 priključka, za razliku od modela koji su prethodili modelu B+, koji su imali samo dva. Također, pruža bolju podršku za priključivanje dodatnih perifernih uređaja bez značajnih smetnji u radu računala te zaštitu od prevelike struje. U slučaju povećane potrebe perifernog uređaja za energijom, koju ne može zadovoljiti računalo, može se koristiti vanjski izvor energije.

E: Ethernet ulaz

Model Raspberry Pi 2 ima, kao i modeli B i B+, standardni RJ45 Ethernet ulaz. Također, moguće je i Wi-Fi povezivanje na Internet pomoću USB priključka.

F: Priključak za serijsko sučelje kamere (eng. *Camera Serial Interface*, skr. CSI)

Pomoću ovog izlaza moguće je izravno povezati kameru na računalo.

G: HDMI (eng. *High-Definition Multimedia Interface*) izlaz

HDMI konektor omogućava digitalni video i audio izlaz. Računalo podržava četrnaest različitih video rezolucija, a HDMI signal je, pomoću vanjskih adaptera, moguće pretvoriti u DVI (koji koriste mnogi monitori), kompozitni signal (analogni video signal kojeg obično prenosi žuti RCA konektor) ili SCART (europski standard za povezivanje audio-vizualne opreme).

H: Ulaz za napajanje

Na računalima Raspberry Pi nema prekidača za uključivanje, nego se za napajanje koristi mikro USB konektor. On ne predstavlja još jedan USB ulaz, nego služi samo za napajanje. Ova vrsta konektora se koristi jer su jeftini i široko dostupni.

I: Priključak za serijsko sučelje zaslona (eng. *Display Serial Interface*, skr. DSI)

Na ovaj konektor se može priključiti ravni trakasti kabel koji se može koristiti za komunikaciju s LCD ili OLED zaslonima.

J: LED diode za prikaz stanja

Pet LED indikatorskih dioda pružaju vizualne podatke o stanju. LED diode vezane za mrežu nalaze se na samom Ethernet ulazu.

K: Utor za SD karticu

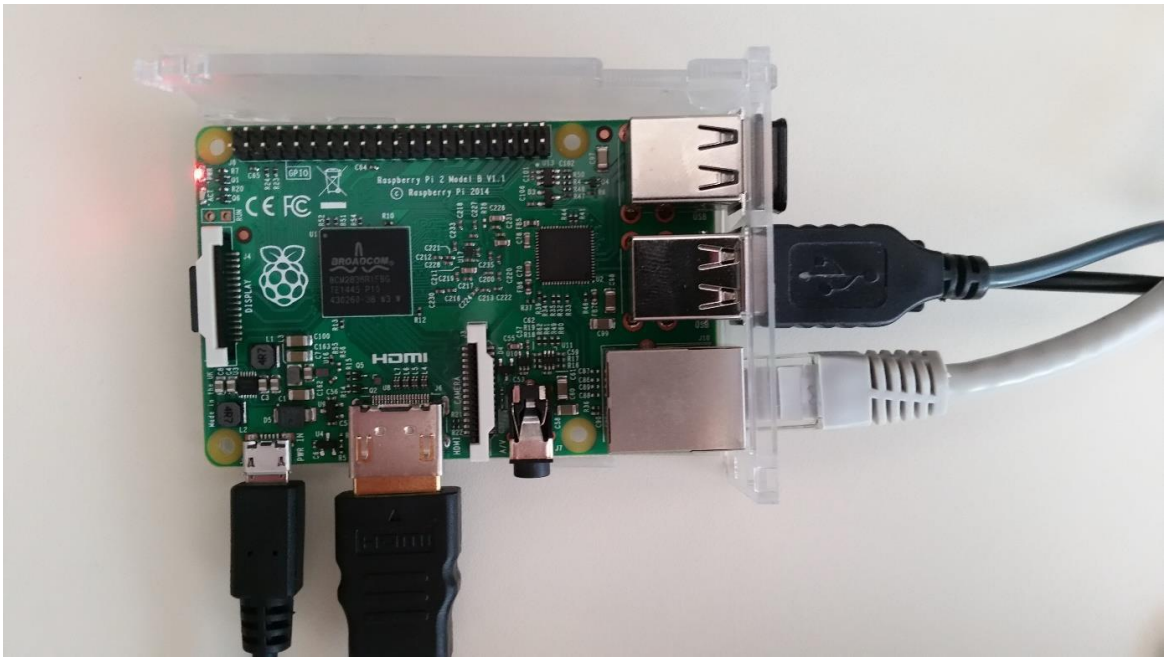
Pošto Raspberry Pi računala nemaju tvrdi disk, sve se pohranjuje na mikro SD karticu. Obzirom da se zalemljeni zglobovi utora za SD karticu mogu oštetiti ako se kartica slučajno savine, kvalitetna mogućnost je Raspberry Pi staviti u neku vrstu zaštitnog kućišta.

3.3. Pokretanje računala i namještanje operativnog sustava

Za pokretanje računala Raspberry Pi 2 potrebno je isto što i za pokretanje bilo kojeg drugog računala ili laptopa: SD kartica, monitor, Ethernet kabel, tipkovnica, miš i izvor napajanja. Kao što sam već naveo preporuča se SD kartica kapaciteta barem 4 GB, obzirom da najčešće korišteni operativni sustav, Raspbian, zauazima 1.8 GB memorije. Također, na kartici se mora nalaziti program NOOBS (eng. *New Out Of the Box Software*) koji omogućava odabir i instalaciju željenog operativnog sustava. Bilo koji zaslon ili TV koji ima HDMI ili DVI ulaz može poslužiti kao zaslon računala, dok se za pristup Internetu koristi standardni Ethernet kabel. Uz to, s Raspberry Pi-em radi bilo koja tipkovnica i miš s USB priključkom. Za napajanje računala koristi se mikro USB napajanje, napona barem 5V, uz 800 mA struje i snage 4 W. Nedovoljna snaga može uzrokovati čudno ponašanje uređaja.

Nakon prikupljanja sve navedene opreme, potrebno ju je spojiti u računalo (Slika 3.) prema sljedećim uputama:

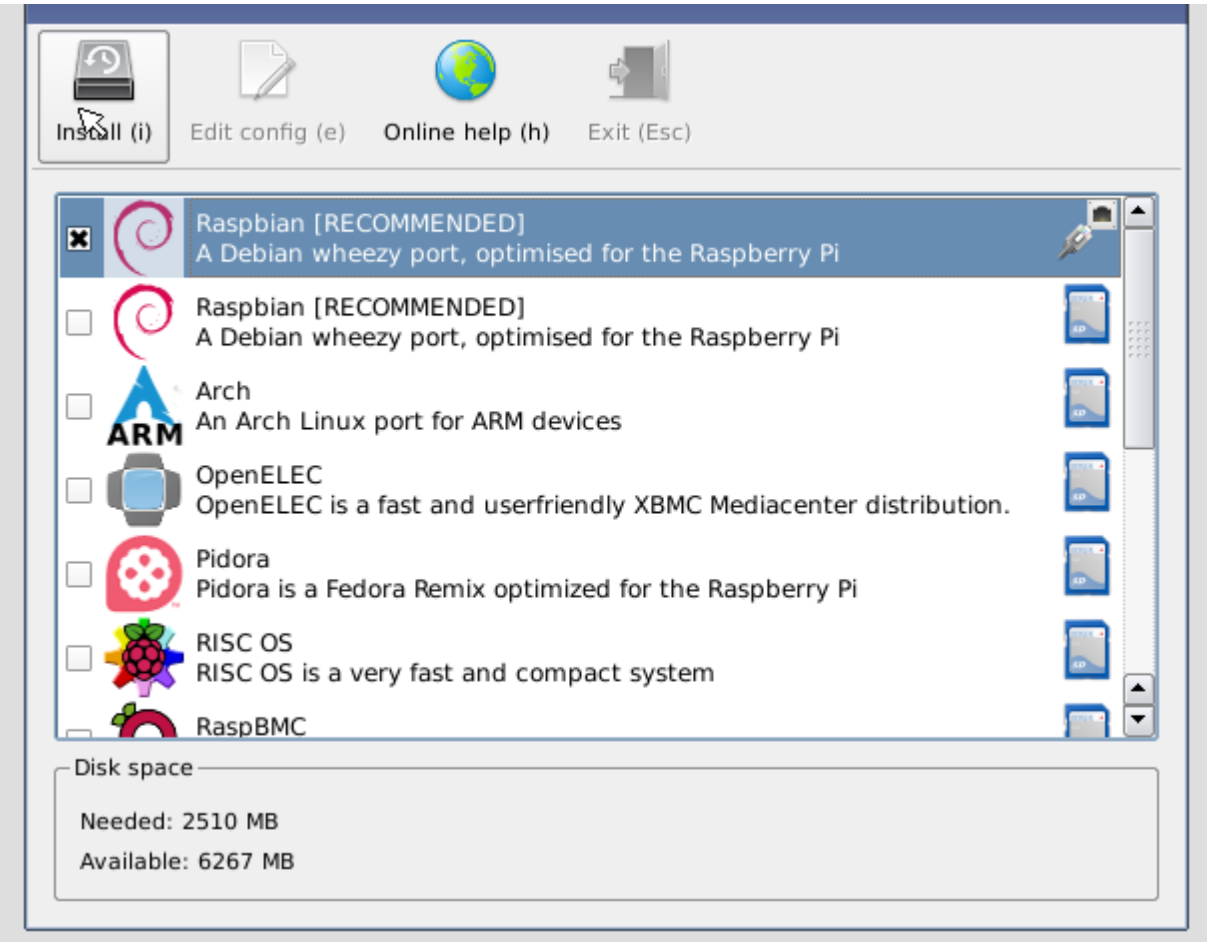
1. Za početak, potrebno je umetnuti SD karticu u utor za SD karticu na računalu, pri čemu ju je nemoguće pogrešno umetnuti.
2. U sljedećem koraku se priključuje tipkovnica i miš u USB priključke računala.
3. Zaslon ili TV mora biti uključen kako bi mogli odabrati ispravan ulaz (pr. HDMI 1, DVI itd.)
4. Zatim treba priključiti HDMI kabel iz računala na zaslon ili TV.
5. Ako je potreban Internet, priključuje se Ethernet kabel u Ethernet ulaz, koji se nalazi pored USB priključaka.
6. Na kraju se priključuje mikro USB napajanje, što rezultira paljenjem i pokretanjem Raspberry Pi računala. [6]



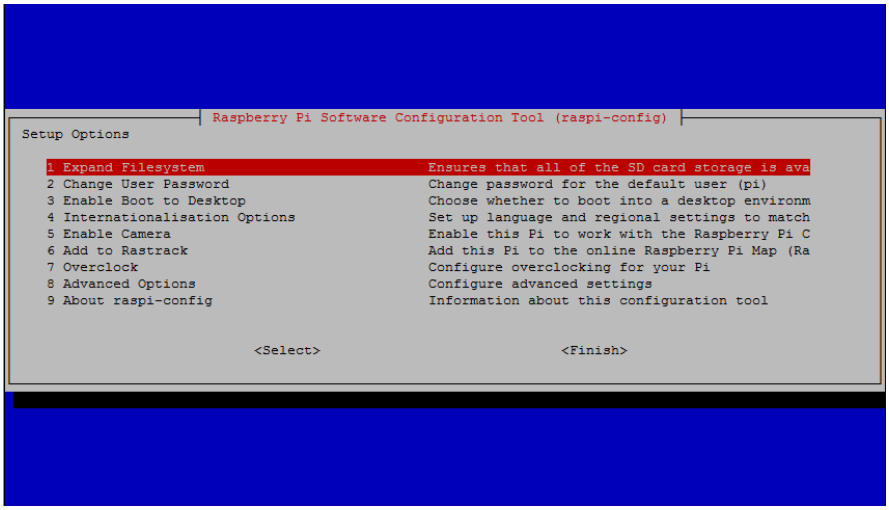
Slika 3. RPi 2 sa priključenim svim perifernim jedinicama,
Ethernet kabelom i napajanjem.

Kako bi se on lakše instalirao, odmah nakon pokretanja računala pokreće se i upravitelj za instalaciju NOOBS. Njega je moguće besplatnim preuzimanjem NOOBS-a i raspakiravanjem dobivene .zip datoteke na praznu, odnosno formatiranu SD karticu. Kod prvog pokretanja računala pojavljuje se prozor sa popisom operativnih sustava koje je moguće instalirati (Slika 4.). Raspberry Pi računala primarno koriste operativne sustave temeljene na Linux-u, a najavljeno je da će Raspbery Pi 2 moći koristiti i vlastitu verziju Windows 10 operativnog sustava. Ipak, trenutno najčešće korišten i operativni sustav preporučen od strane „Raspberry Pi Foundation“ je Raspbian. Za njegovu instalaciju potrebno je samo označiti kućicu s lijeve strane i kliknuti ikonu „Install“. Nakon završetka procesa instalacije, pokreće se izbornik za konfiguraciju (raspi-config) Raspberry Pi-a (Slika 5.) [7]. Ovdje je moguće namjestiti vrijeme i datum za određenu regiju, promijeniti lozinku korisnika (zadano korisničko ime je „pi“, a lozinka „raspberry“), promijeniti postavke vezane za jezik i regionalne postavke, omogućiti Pi-u da radi s Raspberry Pi kamerom, povećati brzinu procesora (eng. overclock) itd.[8] Izbornik za konfiguraciju je moguće pokrenuti bilo kada upisivanjem „sudo raspi-config“ u naredbenu liniju operativnog sustava. „Sudo“ je program za računalne operativne sustave slične Unix-u, koji omogućava korisnicima pokretanje

programa sa sigurnosnim dopuštenjima drugog korisnika, uključujući administratora sustava.



Slika 4. Prozor s popisom operativnih sustava koje je moguće instalirati.



Slika 5. Prozor s izbornikom za konfiguraciju.

4. Podešavanje DHT22/AM2302 senzora

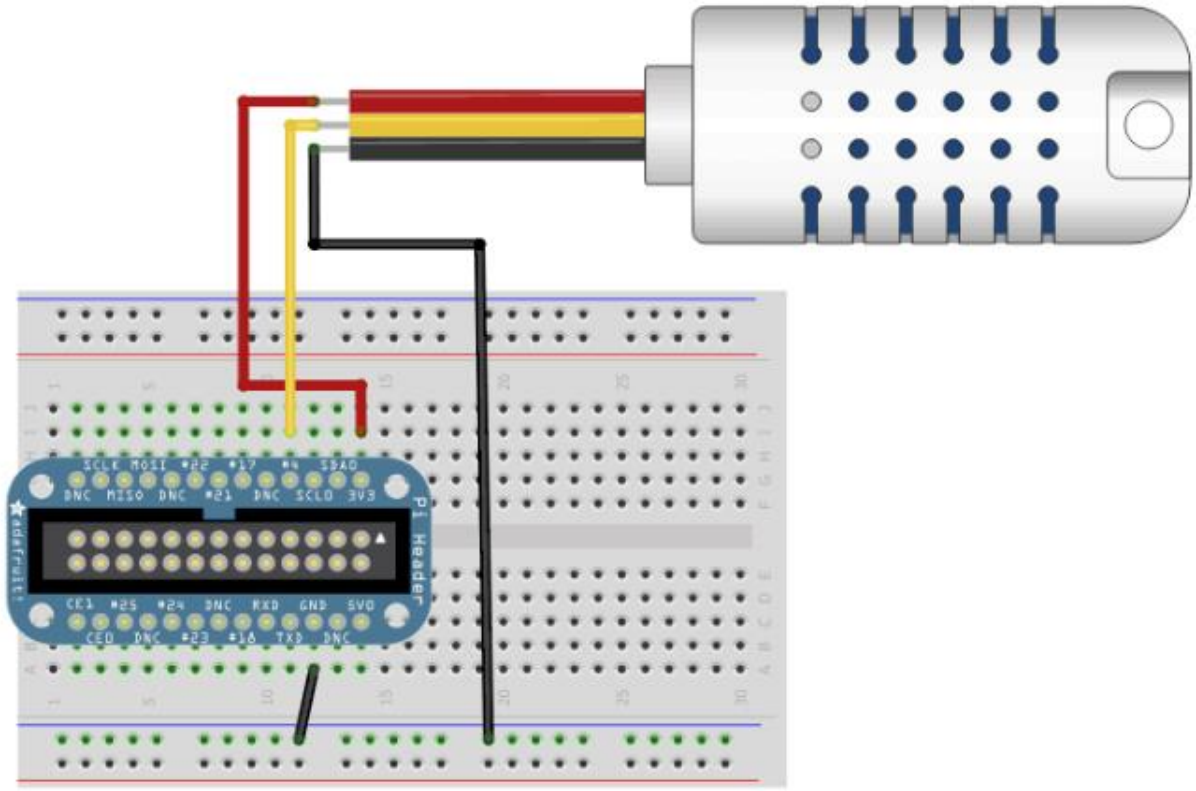
DHT22/AM2302 je digitalni senzor za mjerenje temperature i vlažnosti. DHT22 senzor temperature i vlažnosti je trenutno najjeftiniji senzor dostupan na tržištu koji pruža kalibrirane digitalne izlaze za temperaturu i relativnu vlažnost zraka. AM2302 je žičana verzija već spomenutog senzora DHT22 u praktičnom plastičnom kućištu (Slika 7.). Može mjeriti temperaturu do 80° C, a vlažnost zraka u opsegu od 0-100%, što ga čini idealnim za unutrašnju upotrebu.

Model	AM2302	
Power supply	3.3-5.5V DC	
Output signal	digital signal via 1-wire bus	
Sensing element	Polymer humidity capacitor	
Operating range	humidity 0-100%RH;	temperature -40~80Celsius
Accuracy	**humidity +-2%RH**(Max +-5%RH);	temperature +-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius
Repeatability	humidity +-1%RH;	temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH	
Long-term Stability	+-0.5%RH/year	
Interchangeability	fully interchangeable	

Slika 6. Tablični prikaz specifikacije AM2302 senzora.

Slika 7. AM2302 senzor.

Kako bih ga osposobili, treba ispravno spojiti žice senzora s pinovima Raspberry Pi-a. To je izvedivo samo preko tzv. ekperimentalne pločice (eng. *Breadboard*) iz razloga da se ostvari ispravan strujni krug.



Slika 8. Prikaz spojenih žica senzora s pinovima RPi-a [9].

Dok crvena i crna žica koje su spojene na pinove pod brojem 1 i 6 služe isključivo za paljenje i pokretanje senzora, žuta žica je spojena na GPIO4, pin pod brojem 7 koji služi za komuniciranje s računalom i ima svrhu dohvaćanja podataka sa senzora.

4.1. Instaliranje odguvarajuće Python biblioteke i iščitavanje podataka

Jedan i vjerojatno najjednostavniji način za preuzimanje svih tipova koda i open source projekata je GitHub. GitHub je hosting servis za Git repozitorije koji omogućuje korištenje svih funkcionalnosti Gita. Odnosno to je mjesto gdje svaki programer može podijeliti svoj izvorni kod s ostatkom svijeta. Također, da su Git repozitoriji popriično zastupljeni u Raspberry Pi projektima vidjet ćemo se u nastavku gjde će se koristiti Git razvijen od strane hardverske open source kompanije Adafruit.

Sljedeći kod unesen u terminalnom prozornu na Raspbian OS-u, obuhvaća terminalne naredbe koje podrazumjevaju preuzimanje Python koda sa GitHub-a, preuzimanje i instaliranje biblioteke Adafruit_Python_DHT.

```
sudo apt-get install git
cd~
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo apt-get update
sudo apt-get install build-essential python-dev
sudo python setup.py install
```

Nakon instalacije, slijedi testiranje biblioteke ulaženjem u mapu *examples* i pokretanjem koda u Python datoteci *AdafruitDHT.py*.

```
cd examples
sudo ./AdafruitDHT.py 2302 4
```

Možemo primjetiti da se u nabrebi *sudo ./AdafruitDHT.py 2302 4* nalaze još dva parametra od kojih je *2302* senzor koji se koristi, a *4* predstavnja GPIO koji se koristi za dohvaćanje podataka sa senzora. Uspješno testiran kod bih trebao ispisati temperaturu i vlažnost u terminalnom prozoru Raspbian OS-a(Slika).

Slika 9. Prikaz uspješno testiranog koda.

Kod 1. Prikaz kompletnog koda u Python datoteci AdafruitDHT.py.

```
import sys
import Adafruit_DHT


# Parse command line parameters.
sensor_args = { '11': Adafruit_DHT.DHT11,
                '22': Adafruit_DHT.DHT22,
                '2302': Adafruit_DHT.AM2302 }
if len(sys.argv) == 3 and sys.argv[1] in sensor_args:
    sensor = sensor_args[sys.argv[1]]
    pin = sys.argv[2]
else:
    print('usage: sudo ./Adafruit_DHT.py [11|22|2302] GPIOpin#')
    print('example: sudo ./Adafruit_DHT.py 2302 4 - Read from an AM2302
connected to GPIO #4')
    sys.exit(1)

# Try to grab a sensor reading.  Use the read_retry method which will retry up
# to 15 times to get a sensor reading (waiting 2 seconds between each retry).
humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

# Un-comment the line below to convert the temperature to Fahrenheit.
# temperature = temperature * 9/5.0 + 32

# Note that sometimes you won't get a reading and
# the results will be null (because Linux can't
# guarantee the timing of calls to read the sensor).

if humidity is not None and temperature is not None:
    print('Temp={0:0.1f}*  Humidity={1:0.1f}%'.format(temperature, humidity))
else:
    print('Failed to get reading. Try again!')
    sys.exit(1)
```

5. Ažuriranje podataka online pomoću ThingSpeak platforme

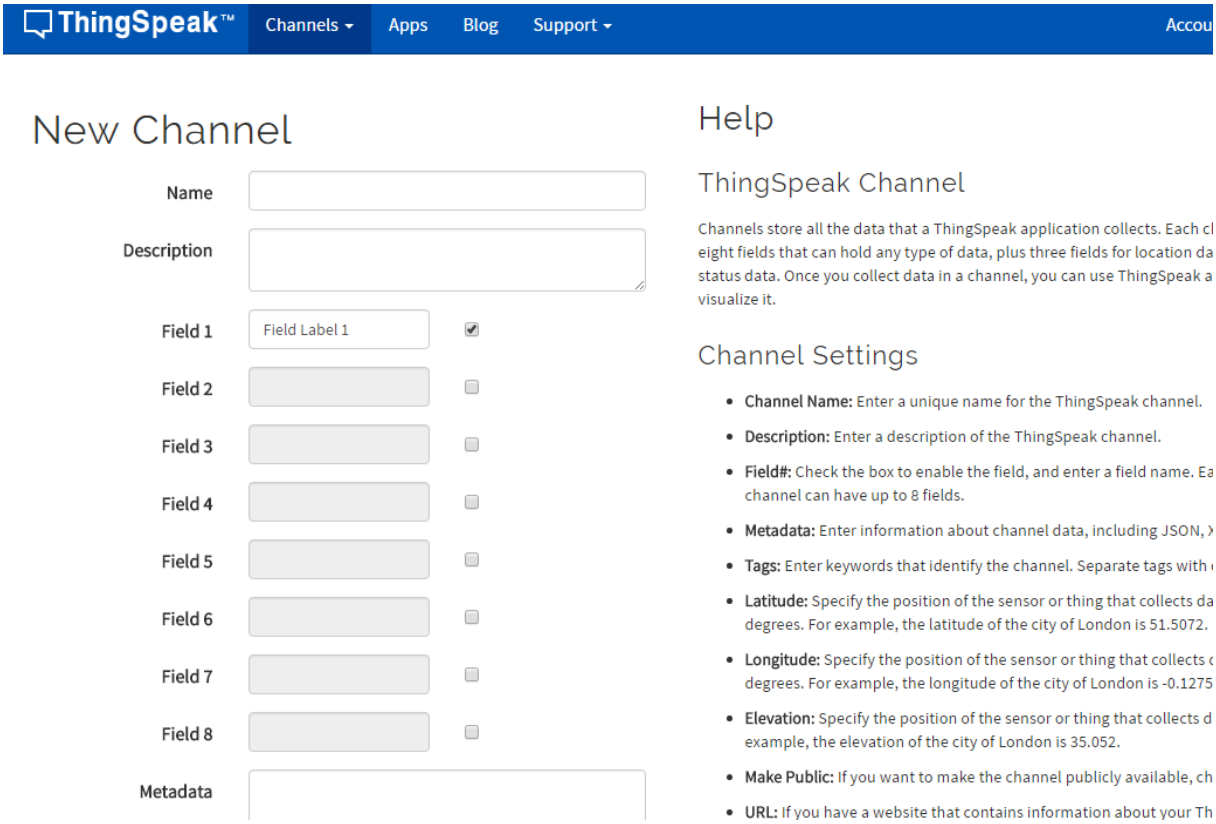
ThingSpeak je jedana od platformi za Internet stvari koja pomaže prikupiti i pohraniti podatke senzora na cloud. Također pomaže razviti aplikacije za Internet stvari i uključuje aplikacije koje pomažu analizirati i vizualizirati podatke senzora.[10]

Glavni element ThingSpeak platforme su *kanali* koji sadrže polja podataka te lokacijska i statusna polja.

ThingSpeak API je također dostupan na GitHub-u i uključuje kompletni ThingSpeak API za procesiranje HTTP zahtjeva, pohranjivanje numeričkih i alfanumeričkih podataka, obradu numeričkih podataka, praćenje lokacije i ažuriranje statusa [10].

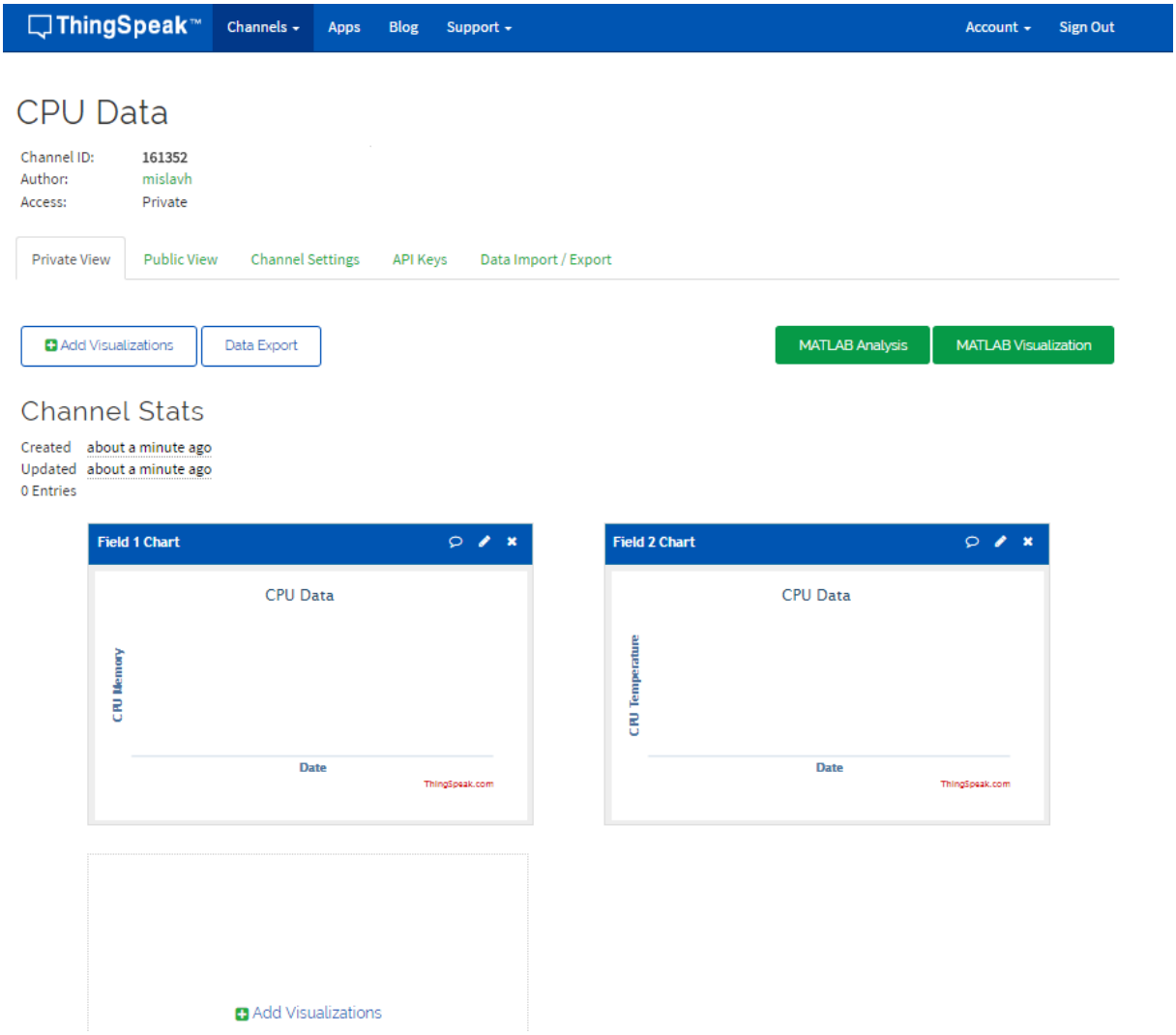
Kako GitHub ima mnošto raznovsnih projekata pogotovo kad je RPi u pitanju, pokraj podataka sezora za temperaturu i vlažnost, na ThingSpeak su prvo ažurirani podaci o temperaturi procesora nađenog koda Thingspeak_CPU_Python-Code.

Prvi korak odmah nakon naravno registracije na ThingSpeak je kreiranje kanala za podatke.



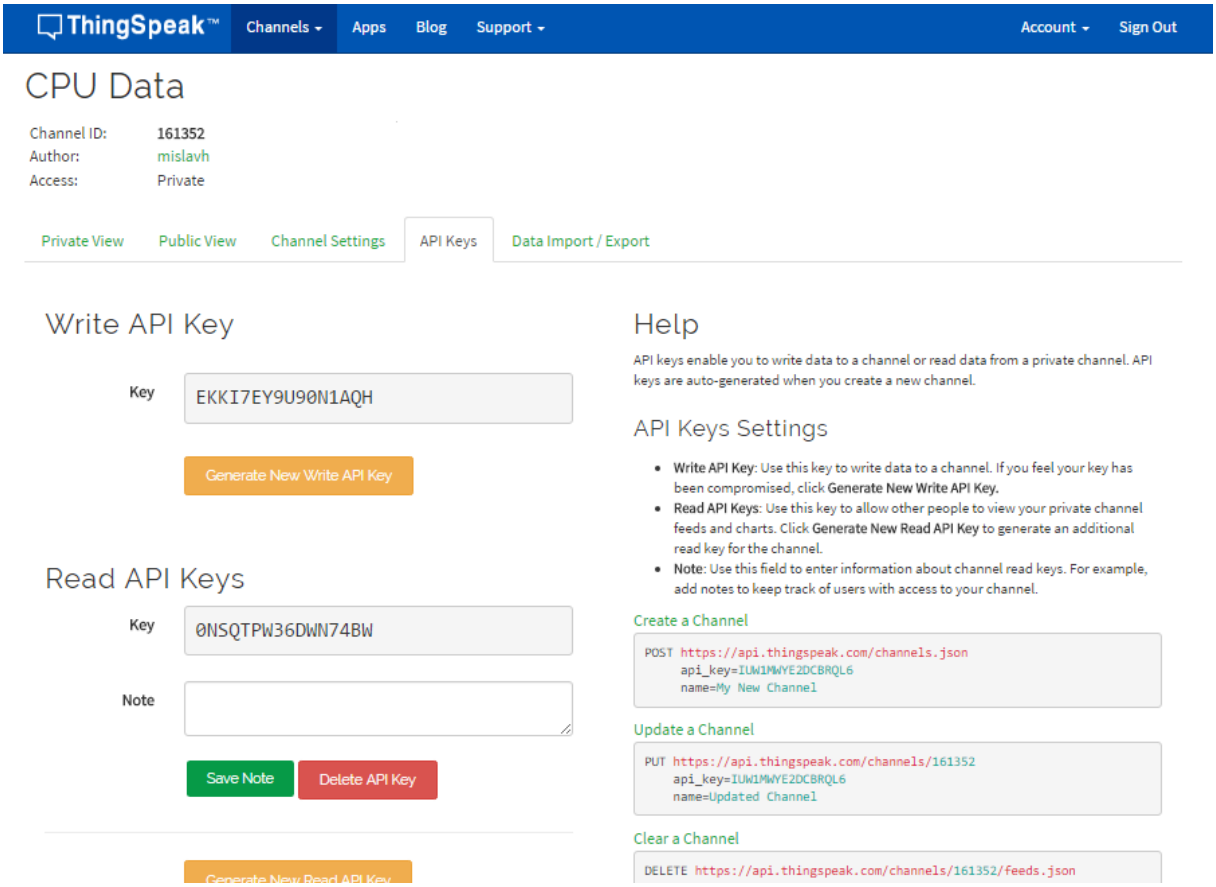
Slika 10. Kreiranje ThingSpeak kanala.

Nakon kreirana dva nova polja za testiranje temperature procesora, ThingSpeak kanal izgleda na sljedeći način:



Slika 11. Prikaz dva kreirana polja za testiranje.

Idući korak je generiranje API ključa koji je neophodan za ažuriranje podataka. API ključ se zatim uključuje u Python kod i na taj način pokretanjem koda će se ažurirati podaci senzora na ThingSpeak kanal.



Slika 12. Generiranje ThingSpeak API ključa.

Nakon generiranog API ključa, jedna od stvari koja je potebna za ažuriranje temeprature procesora na ThingSpeak je uključivanje istog u kod. Opet se koriste terminalne naredbe kako bih se preuzeo kod sa GitHub-a i nakon preuzimanja slijedi otvaranje Python datototeke i ubacivanje API ključa na predviđeno mjesto.

```
git clone https://github.com/sriharshakunda/Thingspeak_CPU_Python-Code

cd Thingspeak_CPU_Python-Code
sudo nano CPU_Python.py
```

Otvorena CPU_Python.py datoteka, odnosno cijeli potrebni kod izgleda na ovaj način:

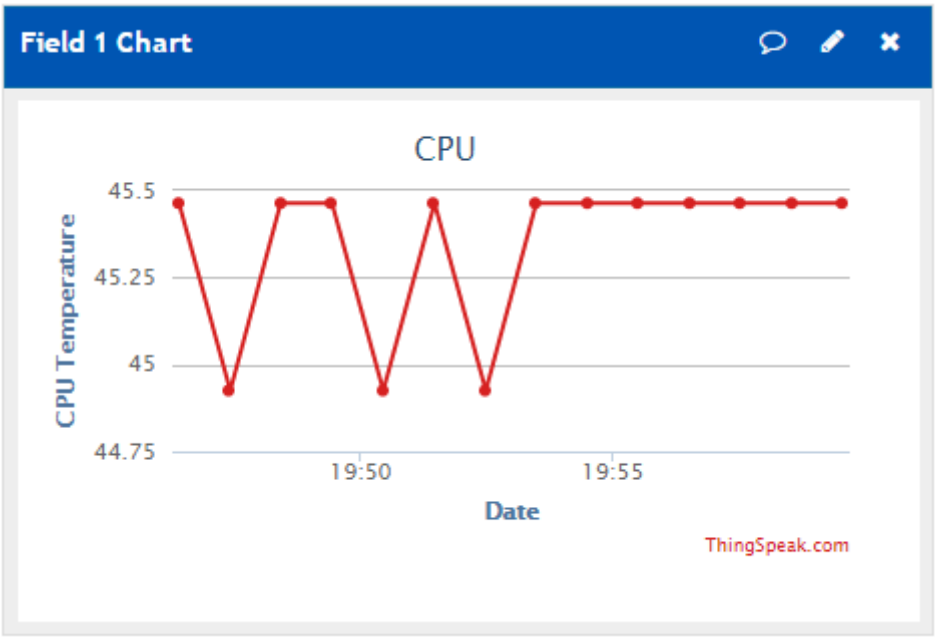
Kod 2. Prikaz koda CPU_Python.py za ažuriranje temeprature procesora na ThingSpeak kanal

```
import httplib, urllib
import time
sleep = 60 # how many seconds to sleep between posts to the channel
key = 'EKKI7EY9U90N1AQH'  # Thingspeak channel to update

#Report Raspberry Pi internal temperature to Thingspeak Channel
def thermometer():
    while True:
        #Calculate CPU temperature of Raspberry Pi in Degrees C
        temp = int(open('/sys/class/thermal/thermal_zone0/temp').read()) / 1e3 # Get Raspberry Pi CPU temp
        params = urllib.urlencode({'field1': temp, 'key':key })
        headers = {"Content-type": "application/x-www-form-urlencoded","Accept": "text/plain"}
        conn = httplib.HTTPConnection("api.thingspeak.com:80")
        try:
            conn.request("POST", "/update", params, headers)
            response = conn.getresponse()
            print temp
            print response.status, response.reason
            data = response.read()
            conn.close()
        except:
            print "connection failed"
        break
#sleep for desired amount of time
if __name__ == "__main__":
        while True:
                thermometer()
                time.sleep(sleep)
```

Inače preporučeni kompajler za CPU_Python.py je Python2.7 koji će se i koristiti pri unosu naredbe za izvršavanje koda.

```
sudo python2.7 CPU_Python.py
```

I nakon kompajliranja koda, te uspostavljanja veze sa ThingSpeak serverom putem API ključa podaci o temperaturi procesora su uspješno preneseni.



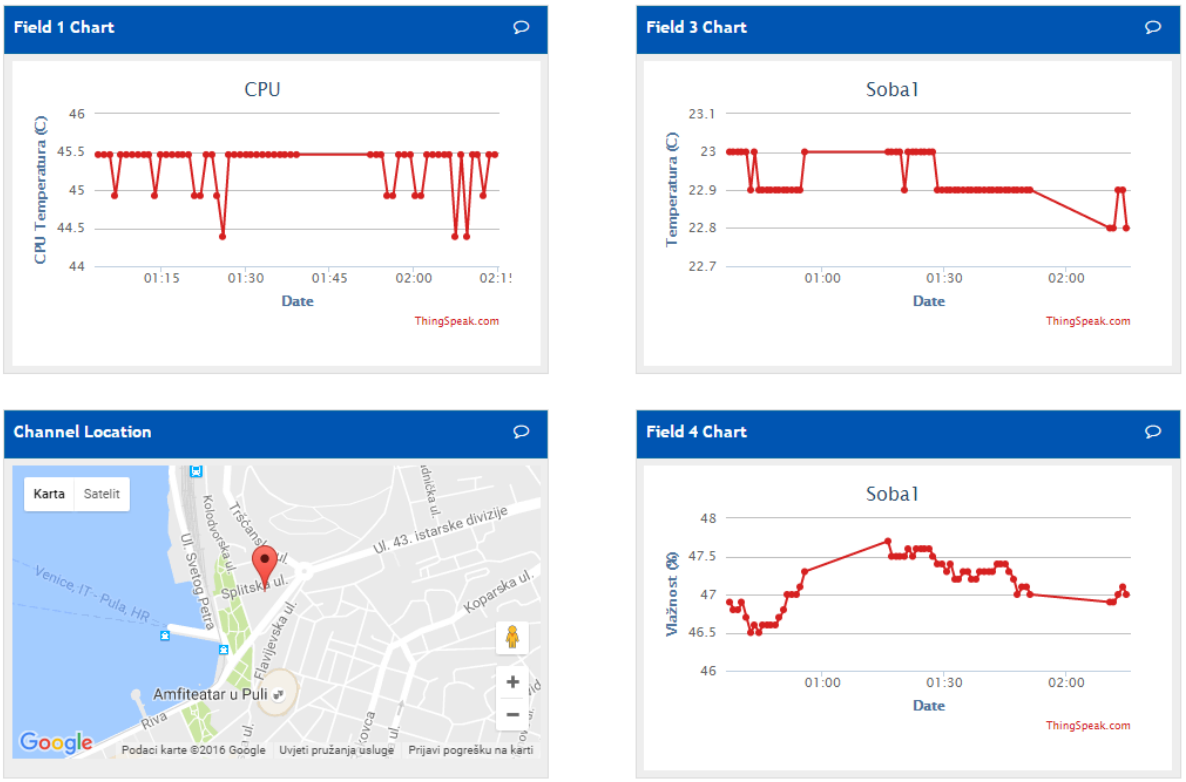
Slika 13. Prikaz temperature procesora na ThingSpeak kanalu.

Poslje provedenog testiranja nad temperaturom procesora potrebno je podatke temperature i vlažnosti zraka prenjeti na ThingSpeak kanal. Na GitHub-u ne postoji odgovarajući kod za ažuriranje podataka pomoću AM2302 senzora na ThingSpeak kanal, iz istog razloga je potrebno osloniti se na vlastito znanje programiranja i pokušat prilagodit kod za pristup ThingSpeak kanalu putem API ključa. Nakon proučavanja već korištenih kodova uspješno se pomoću Adafruit_Python_DHT biblioteke prilagođava kod i dolazi se do zadovoljavajućeg rješenja.

Kod 3. Prikaz koda za ažuriranje podataka temperature i vlažnosti na ThingSpeak poutem API ključa.

```
import sys
import httplib, urllib
import RPi.GPIO as GPIO
import Adafruit_DHT
from time import sleep
# Parse command line parameters.
sensor_args = { '11': Adafruit_DHT.DHT11, '22': Adafruit_DHT.DHT22, '2302':
Adafruit_DHT.AM2302 }
if len(sys.argv) == 3 and sys.argv[1] in sensor_args:
 sensor = sensor_args[sys.argv[1]]
 pin = sys.argv[2]
else:
 print 'usage: sudo ./Adafruit_DHT.py [11|22|2302] GPIOpin#'
 print 'example: sudo ./Adafruit_DHT.py 2302 4 - Read from an AM2302 connected to GPIO #4'
 sys.exit(1)

# Try to grab a sensor reading.  Use the read_retry method which will retry up
# to 15 times to get a sensor reading (waiting 2 seconds between each retry).

while True:
 humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

# Un-comment the line below to convert the temperature to Fahrenheit.
# temperature = temperature * 9/5.0 + 32

# Note that sometimes you won't get a reading and the results will be
# null (because Linux can't guarantee the timing of calls to read the
# sensor).  If this happens try again! Pushing data to Thingspeak python

 if humidity is not None and temperature is not None:
     print 'Temp={0:0.1f}%  Humidity={1:0.1f}%'.format(temperature, humidity)
 else:
     print 'Failed to get reading. Try again!'
     sys.exit(1)
 params = urllib.urlencode({'field3': temperature, 'field4':
humidity,'key':'EKKI7EY9U90N1AQH'})
 headers = {"Content-type": "application/x-www-form-urlencoded","Accept":"text/plain"}
 conn = httplib.HTTPConnection("api.thingspeak.com:80")
 conn.request("POST", "/update", params, headers)
 response = conn.getresponse()
 print response.status, response.reason
 data = response.read()
 conn.close()

 sleep(60)
```

Za uspostavljanje veze u kodu, pomoću Adafruit_Python_DHT biblioteke u odgovarajuće varijable bilo je potrebno spremiti podatke senzora i prenjeti ih kao argumente fukcije za ažuriranje podataka. Podaci temperature i vlažnosti su spremljeni u polja 3 i 4, to sve je stavljeno u beskonačnu pelju, koja će iščitavat podatke sve dok je RPi uključen ili dok se manualno ne prekine. Također je u polje 2 na ThingSpeak kanalu dodan Google maps.



Slika 14. Konačni prikaz ThingSpeak kanala sa ažuriranim podacima.

6. Blynk- Android platforma za Internet stvari

Blynk je android platforma koja je isključivo dizajnirana za Internet stvari. Osim što može prikazivati podatke senzora, te ih pohranjivati i vizualizirati, također može kontrolirati hardver na daljinu.

Postoje tri glavne komponente u platformi:

- Blynk App - omogućuje kreiranje sučelja za projekte koristeći raznovrsne widgete. Riječ je uglavnom o *drag n drop* widgetima
- Blynk Server – odgovoran za svu komunikaciju između pametnih telefona i računala poput RPi.
- Blynk Libraries – za sve platforme poput RPi-ja, omogućuju komunikaciju s poslužiteljem i obrađuju sve ulazne i izlazne naredbe[11].

Ono po ćemu se Blynk još ističe su Virtualni Pinovi koji mogu slati podatke iz mikrokontrolera u Blynk App i primati ih nazad. Sve što se spoji na hardver bit će u mogućnosti komunicirati s Blynk-om. Virtualni Pinovi imaju mogućnost slanja podataka sa aplikacije, obraditi ih u mikrokontroleru i onda si poslat nazad na smartphone. Isto tako ne bih valjalo izostaviti to da jedini način da se vaš smartphone poveže na RPi je preko Auth Token-a. Auth Token ima istu ulogu kao API ključ iz prethodnog poglavlja, a definira se kao jedinstveni identifikator koji je potreban za povezivanje hardvera na smartphone [11].

Da bih se omogućilo normalno komuniciranje sa Virtualnim Pinovima i općenito sa Blynk-om uređaji bih trebali podržavati JavaScript. Iz tog razloga na RPi je potrebno instalirat Node.js koji je open-source, cross-platforma JavaScript runtime okruženja, koristi se za razvijanje raznolikih izbor alata i aplikacija.

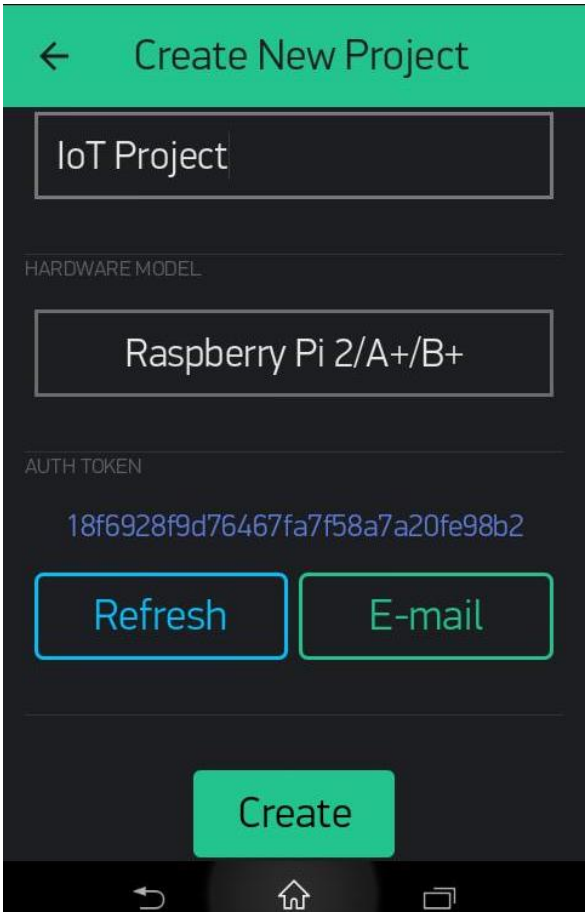
Kod instalacije Node.js-a prvo se mora dodati paket repozitorija na RPi, a zatim instalirati

```
curl -sLS https://apt.adafruit.com/add | sudo bash

sudo apt-get install node
```

Nakon instalacije i potvrde da je Node.js instaliran na RPi i RPi spojen na internet, tada se može preuzeti i instalirati Blynk biblioteka za JavaScript.

```
git clone https://github.com/vshymanskyy/blynk-library-js
cd blynk-library-js
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install build-essential
sudo npm install -g npm
sudo npm install -g onoff
sudo npm install -g blynk-library
```

Idući korak nakon instalacije Blynk biblioteke je registriracija na Blynk i pokretanje novog projekta na Blynk-u kako bih se generirao Auth Token.

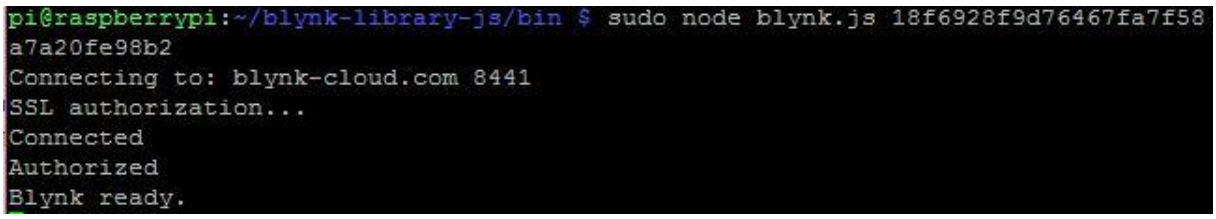


Slika 15. Kreiranje novog projekta u Blynku i generiranje Auth Tokena.

Bitno je naznačit da nakon dobivenog Auth Token treba dodati varijablu okruženja (eng. *Enviroment variable*) koja treba pokazivati na mjesto gdje npm pohranjuje globalno instalirane module[12]. Zatim treba pomoću Node.js-a pokrenuti blynk.js JavaScript datoteku sa Auth Tokenom kao parametrom.

```
export NODE_PATH=/usr/local/lib/node_modules

sudo node blynk.js 18f6928f9d76467fa7f58a7a20fe98b2
```

Nakon ispravno unesene komande zajedno sa Auth Tokenom terminal bih trebao ispisati:

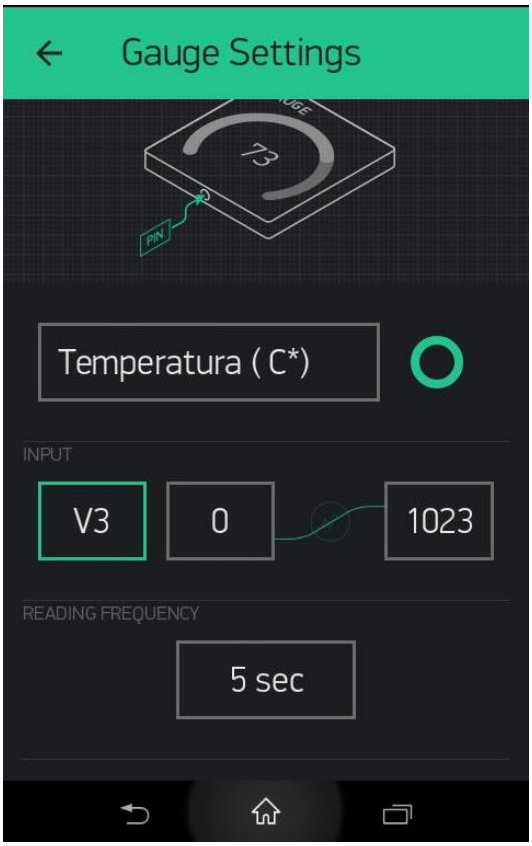


Slika 16. Povezivanje RPi-a s cloudom Blynk-a.

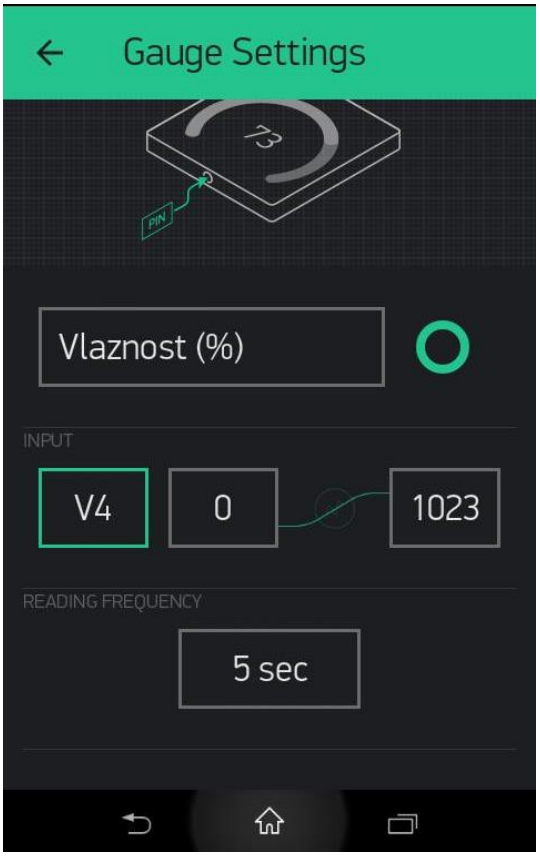
S obzirom da je RPi uspješno spojen na Blynk, može se preuzeti i instalirati odgovarajuće biblioteke za iščitavanje podataka senzora.

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.46.tar.gz

tar zxvf bcm2835-1.46.tar.gz

cd bcm2835-1.46

./configure

make

sudo make check

sudo make install

sudo npm install -g node-dht-sensor
```

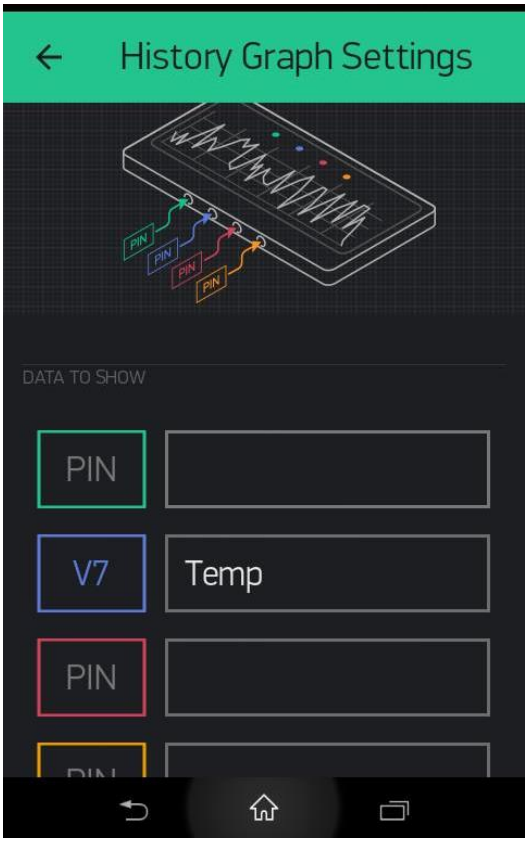
Instalacija nije uobičajena jer se biblioteka bcm2835 mora dodatno raspakirati, što malo odužuje proces. U svrhu daljnjeg nastavka potrebno je u Blynk App-u *drag and drop-at* widgete te im odrediti Virtualne Pinove.



Slika 17. Dodavanje Gauge widgeta Temperatura na Virtualni Pin 3.



Slika 18. Dodavanje Gauge widgeta Vlaznost na Virtualni Pin 4



Slika 19. Dodavanje History Graph widgeta Temp na Virtualni Pin 7

Poslje uspješno dodanih widgeta i Virtualnih Pinova dolazi se i do JavaScript programa za ažuriranje podataka. Pinovi koje su odabrani su V3,V4 i V7 i preko njih će se vršit prijenos podataka pomoću Auth Tokena.

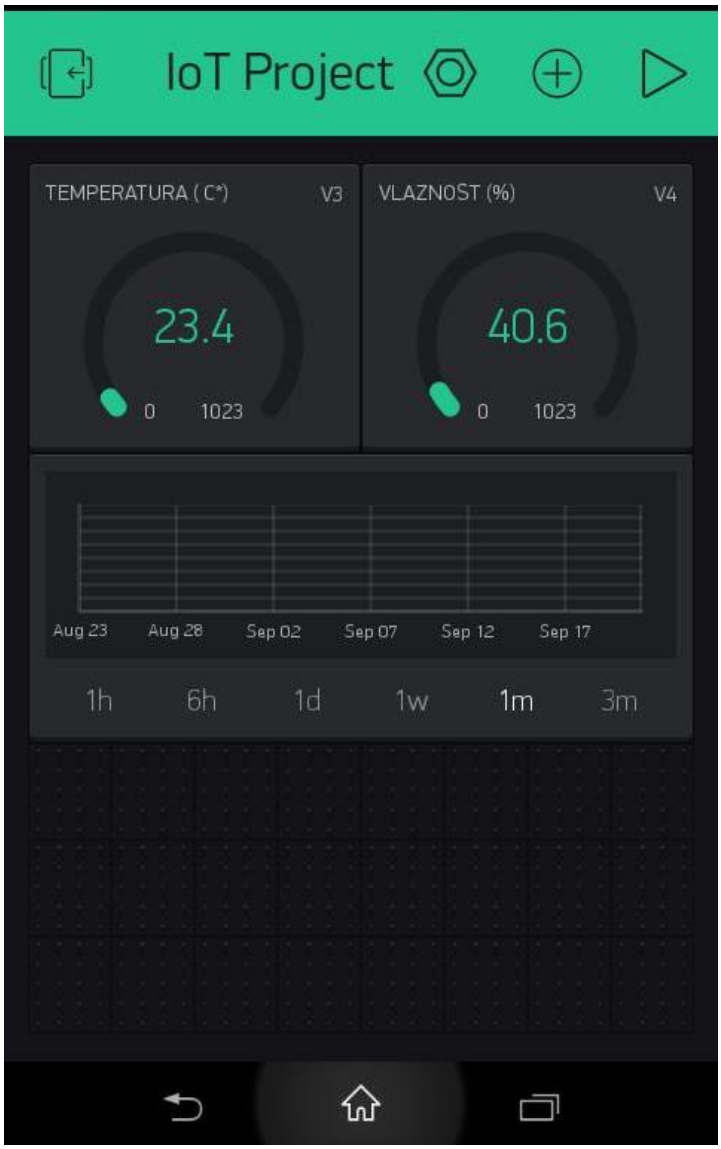
Kod 4. JavaScript datoteka blynk_s.js

```
var blynkLib = require('blynk-library');
var sensorLib = require('node-dht-sensor');

var AUTH = '18f6928f9d76467fa7f58a7a20fe98b2;

// Setup Blynk
var blynk = new blynkLib.Blynk(AUTH);

// Setup sensor, exit if failed
var sensorType = 22; // 11 for DHT11, 22 for DHT22 and AM2302
var sensorPin  = 4;  // The GPIO pin number for sensor signal
if (!sensorLib.initialize(sensorType, sensorPin)) {
    console.warn('Failed to initialize sensor');
    process.exit(1);
}
// Automatically update sensor value every 2 seconds
setInterval(function() {
    var readout = sensorLib.read();
    blynk.virtualWrite(7, readout.temperature.toFixed(1));
    blynk.virtualWrite(3, readout.temperature.toFixed(1));
    blynk.virtualWrite(4, readout.humidity.toFixed(1));

    console.log('Temperature:', readout.temperature.toFixed(1) + 'C');
    console.log('Humidity:   ', readout.humidity.toFixed(1)    + '%');
}, 2000);
```

Primarne stvari koje su se morale izmjeniti u JavaScript kodu su varijabla *var AUTH* u koju se logično morao unijeti Auth Token, *var sensorType* kojoj se vrijednost morala postaviti na 22, a treća je bila prilagodba funkcije *blynk.virtualWrite* koja kao prvi argument uzima broj Virtualnog Pina, a kao drugi željeni podatak senzora bila to temperatura ili vlažnost zraka.

Nakon izvršenog uvida u kod i prilagodbe koda može se započeti ažuriranje podataka na Blynk App pokretanjem blynk_s.js odnosno izvršenjem naredbe:

```
sudo NODE_PATH=/usr/local/lib/node_modules node ./blynk_s.js
```



Slika 20. Konačni izgled aplikacije Blynk na Androidu

7. Zaključak

Internet stvari koji povezuje uređaje i fizičke objekte iz okoline u globalnu mrežu temeljenu na protokolu IP, čini infrastrukturu za razvoj novih mobilnih i web usluga.
Evolucija društva u digitalno društvo i nastanak digitalne ekonomije, pod utjecajem jačanja i širenja društvenih mreža, danas služi kao temelj na kojemu se može razvijati koncept Interneta stvari. Tehnološki napredak i razvoj senzora, aktuatora, skladišta podataka i mrežnih sposobnosti pruža nam uvid u neke od mogućih scenarija budućeg razvoja i primjene Interneta stvari.
Ovim završnim radom prikazao sam postupke ažuriranja podataka senzora web servisom i Android platformom za Internet stvari.
Postavljeni cilj završnoga rada je ostvaren, senzor je uspješno isprogramiran te je napravljena i njemu svojstvena aplikacija na Android OS-u. Samim time, omogućeno je lako i jednostavno pregledavanje podatkovnog sadržaja senzora, a sama uporaba aplikacija nije ni malo komplicirana.
Također bit Interneta stvari je ostvarena, uređaji su međusobno povezani, a podaci su online. Projekti izrađeni u sklopu Interneta stvari su vrlo raznoliki i primjenu pronalaze u mnogo domena: edukacija, zdravstvo i medicina, promet, domaćinstvo, poljoprivreda, trgovina, zabava i dr. Većina tih projekata ima zajednički cilj, a to je olakšati čovjeku život.
Internet stvari nekome može djelovati zastrašujuće ili prijeteće jer nas približava automatiziranom, robotskom svijetu. No, realnost je da smo već počeli automatizirati sve više aspekata radnog mjesta, a dom je sljedeći korak. To ne znači da će stvari razmišljati za nas ili donositi odluke umjesto nas. Internet stvari nam jednostavno omogućuje da se usredotočimo na veće projekte od paljenja svjetla ili provjeravanja temperature hladnjaka.

Literatura:

[1] Smith, Ian G. (2012) „Internet of Things 2012 New Horizons“. Halifax, UK., 2012.

[2] Raspberry Pi, About. https://www.raspberrypi.org/about/

(datum pristupa 15.9.2016.)

[3] Linux User, Top 4 Raspberry Pi OS. http://www.linuxuser.co.uk/reviews/top-4-raspberry-pi-os (datum pristupa 17.9.2016.)

[4] Raspbian. http://www.raspbian.org (datum pristupa 18.9.2016.)

[5] http://en.wikipedia.org/wiki/ARM_Cortex-A7 (datum pristupa 15.9.2016.)

[6] Raspberry Pi Foundation: https://www.raspberrypi.org/help/quick-start-guide/ (datum pristupa 18.9.2016.)

[7] Raspberry Pi Foundation: https://www.raspberrypi.org/help/noobs-setup/ (datum (datum pristupa 16.9.2016.)

[8] Raspberry Pi Foundation:

https://www.raspberrypi.org/documentation/configuration/raspi-config.md(datum pristupa 18.9.2016.)

[9] Adafruit Industries; https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/wiring Adafruit Industries (datum pristupa 18.9.2016.)

[10] Mathworks: https://www.mathworks.com/help/thingspeak/ (datum pristupa 16.9.2016.)

[11] Blynk: http://docs.blynk.cc/#getting-started-getting-started-with-the-blynk-app-4-auth-(datum pristupa 17.9.2016.)

[12] Instuctables: Blynk + JavaScript in 20 minutes

http://www.instructables.com/id/Blynk-JavaScript-in-20-minutes-Raspberry-Pi-Edison/ (datum pristupa 19.9.2016.)

Popis slika:

Slika 1. Gornja strana računala RPi 2 5
Slika 2. Donja strana računala RPi 2 5
Slika 3. RPi 2 sa priključenim svim perifernim jedinicama, Ethernet kabelom i napajanjem 9
Slika 4. Prozor s popisom operativnih sustava koje je moguće instalirati 10
Slika 5. Prozor s izbornikom za konfiguraciju 10
Slika 6. Tablični prikaz specifikacije AM2302 senzora 11
Slika 7. AM2302 senzor 11
Slika 8. Prikaz spojenih žica senzora s pinovima RPi-a 12
Slika 9. Prikaz uspješno testiranog koda 13
Slika 10. Kreiranje ThingSpeak kanala 15
Slika 11. Prikaz dva kreirana polja za testiranje 16
Slika 12. Generiranje ThingSpeak API ključa 17
Slika 13. Prikaz temperature procesora na ThingSpeak kanalu 19
Slika 14. Konačni prikaz ThingSpeak kanala sa ažuriranim podacima 21
Slika 15. Kreiranje novog projekta u Blynku i generiranje Auth Tokena 23
Slika 16. Povezivanje RPi-a s cloudom Blynk-a 24
Slika 17. Dodavanje Gauge widgeta Temperatura na Virtualni Pin 3 25
Slika 18. Dodavanje Gauge widgeta Vlaznost na Virtualni Pin 4 26
Slika 19. Dodavanje History Graph widgeta Temp na Virtualni Pin 7 26
Slika 20. Konačni izgled aplikacije Blynk na Androidu 28

Popis kodova:

Kod 1. Prikaz kompletnog koda u Python datoteci AdafruitDHT.py 14
Kod 2. Prikaz koda CPU_Python.py za ažuriranje temeprature procesora na ThingSpeak kanal 18
Kod 3. Prikaz koda za ažuriranje podataka temperature i vlažnosti na ThingSpeak poutem API ključa 20
Kod 4. JavaScript datoteka blynk_s.js 27

Sažetak

Internet stvari, kao koncept, postoji već duže vrijeme, no tek je u zadnjih nekoliko godina došlo do porasta primjene ove tehnologije. Razlog tome je napredak informacijske i komunikacijske tehnologije. Internet stvari povezuje uređaje i fizičke objekte iz okoline u globalnu mrežu temeljenu na IP protokolu. Tehnološki napredak i razvoj senzora, aktuatora, skladišta podataka i mrežnih sposobnosti pruža nam uvid u neke od mogućih scenarija budućeg razvoja i primjene Interneta stvari. Glavna zamisao rada je bila osposobiti senzor i uspostaviti komunikaciju sa mikroprocesorskim računalom. Zatim uz uspješno dobivene podatke senzora, pomoću računala iste podatke ažurirati na Web i Android platformu za Internet stvari. Kao moguće rješenje za postizanje konačnog cilja rada, koristi se računalo Raspberry Pi 2 i digitalni senzor za temperaturu i vlažnost. U radu je prikazan i objašnjen cjelokupan postupak pripreme i korištenja senzora i računala Raspberry Pi, od instalacije operativnog sustava na računalo do izrade programa i provedbe prikupljanja podataka.

Ključne riječi: Internet stvari, senzor, programiranje, Raspberry Pi

Summary

As a concept, Internet of Things has been around for a number of years, but its expansion has only started a few years ago. The reason for this are the advances in the information and communication technology. Internet of Things connects devices and physical objects from the environment in a global network based on IP protocol. Together with technological advancements in the shape of sensors, actuators, data warehouses and networking give us a glimpse into what the future of Internet of Things could look like. The main idea of this paper was to enable sensor to communicate with microprocessor computer. Afterwards with successfully obtained sensor data, use a computer to update the same data at the Web and Android platform for Internet of things. As a possible solutions used for achieving the main goal of this paper are Raspberry Pi 2 and digital temperature and humidity sensor. This paper presents the entire set up and use procedure of sensor and Raspberry Pi computer, from installation of the operating system to coding and data gathering.

Keywords: Internet of Things, sensor, programming, Raspberry Pi