

Implementacija sustava pametnog doma

Grgić, Mika

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Karlovac University of Applied Sciences / Veleučilište u Karlovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:128:272399>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-20**



VELEUČILIŠTE U KARLOVCU
Karlovac University of Applied Sciences

Repository / Repozitorij:

[Repository of Karlovac University of Applied Sciences - Institutional Repository](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJ

VELEUČILIŠTE U KARLOVCU
STROJARSKI ODJEL
Stručni studij Mehatronike

Mika Grgić

**IMPLEMENTACIJA SUSTAVA
PAMETNOG DOMA**

Završni rad

Karlovac, 2016. godina.

VELEUČILIŠTE U KARLOVCU
STROJARSKI ODJEL
Stručni studij Mehatronike

Mika Grgić

**IMPLEMENTACIJA SUSTAVA
PAMETNOG DOMA**

Završni rad

mr.sc. Vedran Vyroubal

Karlovac, 2016. godina.

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu, uz stručno vodstvo mentora mr. sc. Vedrana Vyroubala, kome se srdačno zahvaljujem na ukazanoj podršci i mentorstvu.

Također se zahvaljujem svim kolegama i prijateljima na podršci tokom studija, a posebno kolegi Josipu Šeketi bacc. ing. na tehničkoj pomoći tokom izrade rada.

Od srca se zahvaljujem svojoj obitelji, koja je bila najveća potpora tokom studiranja i rada.

Mika Grgić



VELEUČILIŠTE U KARLOVCU

Stručni studij: Stručni studij Mehatronike.....

Usmjerenje: Mehatronika..... Karlovac, 22.09.2016.....

ZADATAK ZAVRŠNOG RADA

Student: Mika Grgić..... Matični broj: 01.126.10007.....

Naslov: Implementacija sustava pametnog doma.....

Opis zadatka:

Zadatak završnog rada je implementirati funkcionalni model pametnog doma. Za senzorsko-kontrolnu jedinicu koristi se mikrokontroler STM32F103C8T6 i ESP8266 kao bežični modul. Senzor koji se koristi je BMP280 koji služi za očitavanje temperature, vlage i pritiska.

Rad treba obuhvatiti sljedeće cjeline:

1. Opis mikrokontrolera i njegovih modula
princip rada, raspored pinova, opis I2C komunikacije, isplativost
2. Osnovne pretpostavke modela pametnog doma
izrada modela i objašnjenje izbora kod dizajna modela
3. Opis senzorsko-kontrolne jedinice
ulazno-izlazne karakteristike jedinice, implementacija servisnog terminala
4. Opis serverske infrastrukture
instalacija baze podataka i izrada sučelja
5. Radni primjer pametnog doma
prikazati interakciju sa serverom i između servera i senzorsko-ulazne jedinice na praktičnom primjeru

Zadatak zadan:

Rok predaje rada:

Predviđeni datum obrane:

15.7.2016.

26.9.2016.

19.12.2016.

Mentor:
mr.sc. Vedran Vyroubal

Predsjednik Ispitnog povjerenstva:
mr.sc. Marina Tevčić

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	II
POPIS TABLICA.....	IV
SAŽETAK.....	V
SUMMARY	VI
1. UVOD.....	1
2. ELEMENTI I PROTOKOLI SENZORSKE STANICE	3
2.1. I ² C Protokol.....	6
2.1.1. Terminologija.....	8
2.1.2. Signali na sabirnici.....	8
2.1.3. Serijski prijenos podataka	9
2.1.4. START i STOP stanje.....	10
2.1.5. Komunikacija s izvršiteljima sa 7-bitnim adresama	10
2.2. BME280 Senzor	13
2.2.1. „Sleep mode“	14
2.2.2. „Forced mode“	14
2.2.3. „Normal mode“	14
2.3. ESP8266 mikrokontroler.....	20
2.4. STM32F103 mikrokontroler	27
2.5. Servisni terminal	32
3. POSTAVLJANJE LAMP SERVERA.....	34
3.1. Linux OS	34
3.2. Apache server.....	35
3.2.1. Instalacija i konfiguracija na Ubuntu OS	35
3.3. PHP i MySQL	36
4. ZAKLJUČAK.....	39
LITERATURA.....	40

POPIS SLIKA

Slika 1. Primjer sustava pametnog doma	1
Slika 2. Dijagram toka za jedan ciklus očitavanja i slanja podataka senzorske stanice	4
Slika 3. Moduli senzorske jedinice	5
Slika 4. Primjer spoja I ² C sabirnice	7
Slika 5. Open-drain konfiguracija SDA pina kod I ² C protokola.....	9
Slika 6. Ciklus prijenosa podataka prema I ² C protokolu	9
Slika 7. Primjer odabira izvršitelja i registra na izvršitelju te pisanje u registar.....	11
Slika 8. Primjer iščitavanja više bajtova s početnim registrom na izvršitelju.....	12
Slika 9. BME280 Senzor.....	13
Slika 10. Raspored pinova BME280	15
Slika 11. Blok dijagram BME280	16
Slika 12. Mapa memorije senzora BME280	17
Slika 13. Struktura koja se koristi za iščitavanje memorije iz senzora	19
Slika 14. ESP8266 mikrokontroler.....	20
Slika 15. Raspored pinova ESP8266.....	21
Slika 16. Primjer spajanja na bežičnu mrežu i provjera dobivene IP adrese	24
Slika 17. Primjer dohvata HTML koda internet stranice	25
Slika 18. Podjela ARM procesora	27
Slika 19. Raspored pinova na 48-pinskoj verziji mikrokontrolera.....	28
Slika 20. Korišteni pinovi na STM32F103C8 mikrokontroleru	29
Slika 21. Glavni prozor KEIL integriranog okruženja.....	30
Slika 22. J-Link JTAG/SWD programator i debugger.....	31
Slika 23. Dijagram toka rada servisnog terminala	32
Slika 24. Testna stranica apache servera.....	36

Slika 25. PHP stranica za prihvatanje podataka.....	37
Slika 26. PHP stranica za dohvatanje podataka iz baze.....	38
Slika 27. Direktno isčitavanje baze preko „phpMyAdmin“ sučelja.....	38

POPIS TABLICA

Tablica 1. Popis posebnih adresa definiranih I ² C protokolom.....	11
Tablica 2. Karakteristike BME280 senzora	13
Tablica 3. Oznake pinova BME280	15
Tablica 4. Memorijske lokacije i tipovi podataka potrebnih kalibracijskih koeficijenata	17
Tablica 5. Oznake pinova ESP8266.....	21
Tablica 6. Primjer AT naredbi na ESP8266.....	22

SAŽETAK

Cilj ovog rada je bila implementacija modela pametnog doma. Poput osobnih računala, automatizacija doma (popularno nazvani pametnim domovima) duguje svoje podrijetlo industriji. Velika „mainframe“ računala su preci današnjih osobnih računala i industrijski automatizacijski sustavi su početna točka za razvoj automatizacije domova, dostupnoj široj potrošačkoj publici. Iako ova rastuća industrija ima neke boljke, poput smanjene ili nepostojeće sigurnosti, s porastom filozofije „Internet of Things“, svakako možemo očekivati da ovo tržište samo raste.

Ključne riječi: pametni dom, pametna kuća, automatizacija, Internet of Things, IoT

SUMMARY

The goal of this thesis was implementation of a functional model of a smart home. Just like personal computers, home automation (popularly called smart homes) owes its origins to the industry. Mainframe computers were predecessors of personal computers and industrial automation systems are basis for a smaller, consumer scale home automation. Although this booming industry has some ailments like reduced security or not at all, with the rise of „Internet of Things“ philosophy we certainly can expect this market only to grow.

Keywords: smart home, automation, Internet of Things, IoT

Sama ideja pametnog doma je već aktualna preko 30 godina, no implementacija je morala pričekati tehnološka rješenja koja su u današnje vrijeme pojavom jednostavnih i jeftinih mikrokontrolera, procesora i senzora postale stvarnost.

Svaki pametni dom se sastoji od slijedećih elemenata:

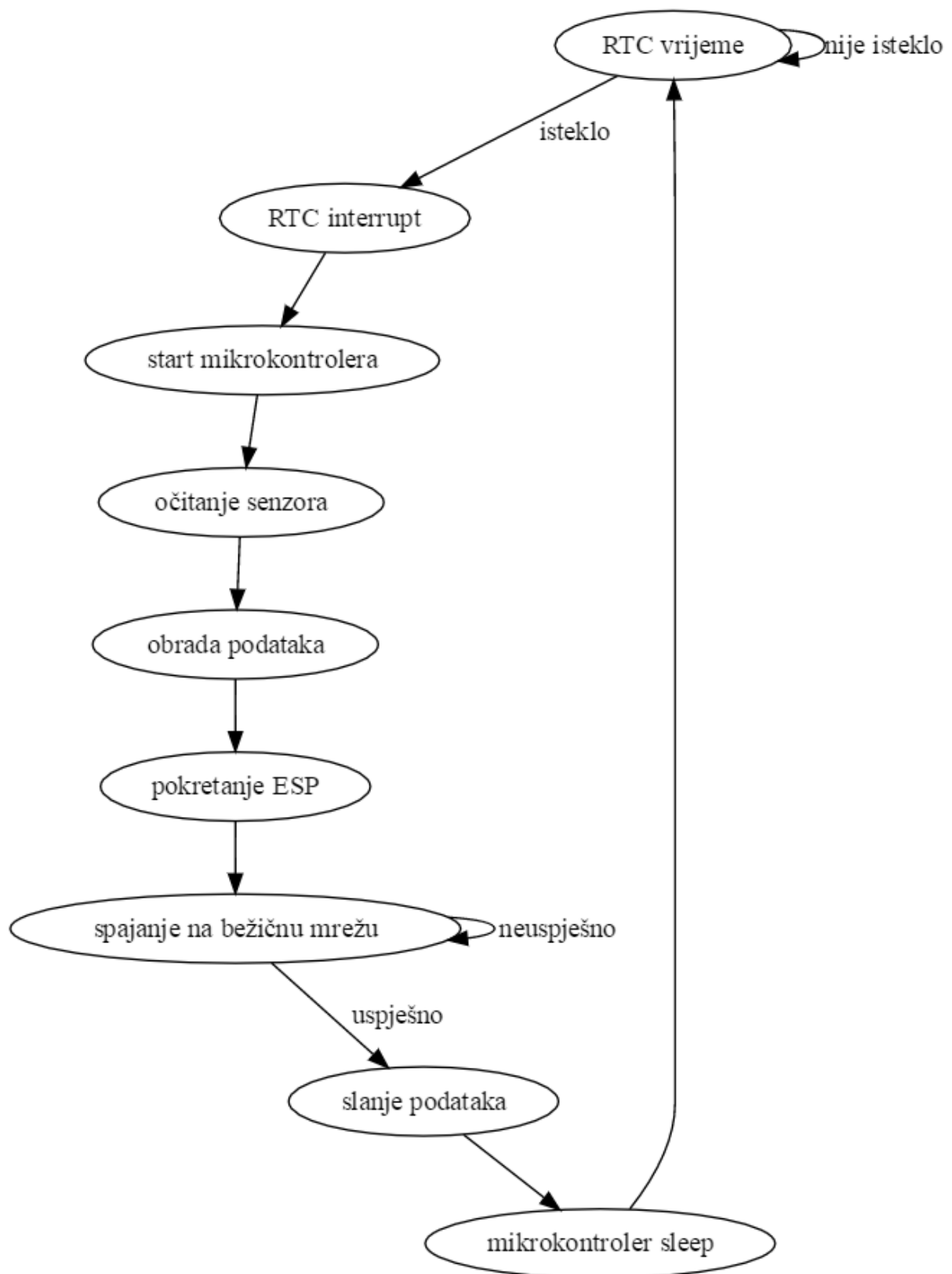
- **osjetilnih elemenata** – razni senzori koji prihvaćaju podatke iz okoline kao što su temperatura, svjetlina, vlažnost, CO₂, razina buke, razni detektori pokreta
- **kontrolera** – glavni zadatak je prihvati podataka sa senzora, slanje u bazu i slanje naredbi prema kontrolnoj jedinici
- **kontrolna jedinica** – upravlja raznim elementima u kućanstvu s ciljem postizanja zadanih vrijednosti, npr. kontroliranje temperature putem ventilacije, grijanja i klimatizacije
- **korisničko sučelje** – centralno mjesto sustava, korisnik može očitati, unositi i izmjenjivati vrijednosti po volji, i davati direktne naredbe. Korisničko sučelje može biti obično računalo

2. ELEMENTI I PROTOKOLI SENZORSKE STANICE

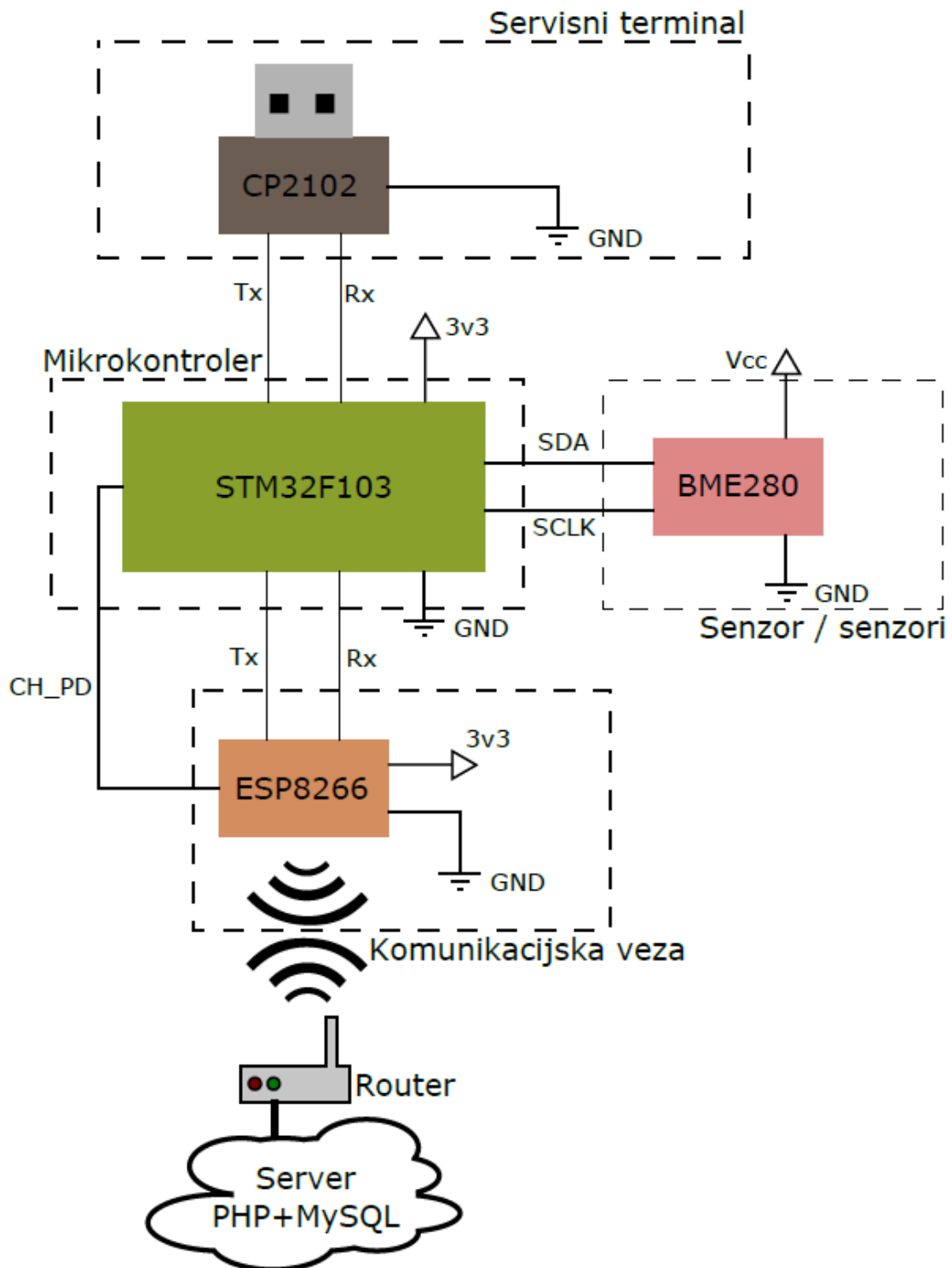
Model senzorske stanice se sastoji od četiri funkcionalna dijela:

- servisni terminal
- senzor ili senzori
- mikrokontroler koji vrši prikupljanje i obradu podataka sa senzora te za komunikaciju sa servisnim terminalom te slanje podataka na baznu stanicu
- komunikacijska veza sa baznom stanicom

Sama senzorska stanica se sastoji od senzorskog modula BME280 koji služi za očitavanje podataka iz okoline (tlak, vlažnost i temperatura), te ih STM32F103 mikrokontroler očitava preko I²C protokola. Samo očitavanje se obavlja samo jednom, zbog uštede energije. Kada STM32 očita podatke sa senzora, nakon obrade ih šalje preko serijskog porta koji je povezan s ESP8266 modulom, koji u ovom slučaju služi samo kao modul za bežičnu komunikaciju sa baznom stanicom. Predviđeno je da napajanje senzorske stanice bude baterija, tako da potrošnja energije mora biti minimalna. U svrhu toga, glavni upravljač koji raspolaže dostupnom energijom je sam mikrokontroler. Mikrokontroler je većinu vremena u stanju niske energije („sleep“ mode) te se budi nakon određenog vremena.



Slika 2. Dijagram toka za jedan ciklus očitavanja i slanja podataka senzorske stanice



Slika 3. Moduli senzorske jedinice

2.1. I²C Protokol

I²C protokol je 1982. godine kreirala kompanija Philips Semiconductor, i služi za kratkometražnu komunikaciju između integriranih krugova i mikroprocesora (obično unutar tiskane pločice). Na samu sabirnicu komponente se spajaju direktno, bez dodatnih komponenti (tzv. „Glue logic“). Broj komponenti je ograničen ukupnim kapacitetom koji smije biti na jednoj liniji sabirnice, a on iznosi 400 pF i adresnim prostorom koji je uobičajeno 7-bit ili 10-bit. Prva standardizirana verzija tog protokola se pojavila 1992. godine, a od 2006. godine je besplatan za korištenje i nije potrebna licenca.

Primjena I²C protokola:

- spajanje komponenta niske brzine na računalo
- komunikacija s EEPROM i NVRAM memorijama
- pristup D/A i A/D pretvornicima
- upravljanje LCD ekranima
- očitavanje vrijednosti sa senzora
- paljenje ili gašenje uređaja ili računalne periferije

Najbitnija prednost je da mikrokontroler može upravljati s mnogo senzora koristeći samo dva priključka, također komponente se mogu skidati i stavljati na sabirnicu po volji (tzv. „Hot-swapping“). Dok je najbitniji nedostatak brzina same sabirnice u odnosu na ostale.

Povijest I²C protokola:

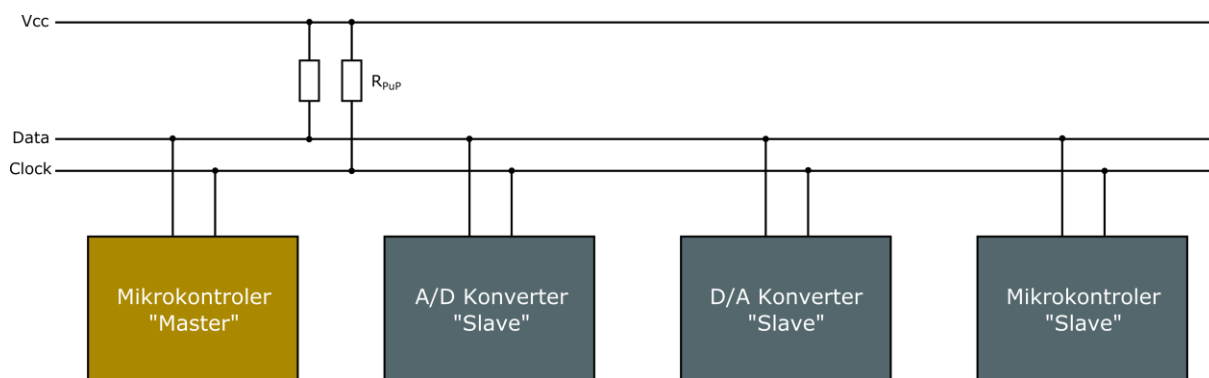
1982. – prvi dizajn, brzina sabirnice je 100 kbit/s, služila je za kontrolu raznih Philips-ovih čipova

1992. – prva standardizirana verzija, dodan je tzv. „Fast-mode“ koji omogućuje brži prijenos podataka (do 400 kbit/s)

1998. – verzija 2 standarda – još veće brzine (do 3,4 Mbit/s), te optimizirana potrošnja energije

2007. – verzija 3 standarda – dodan je „Fast –mode plus“ (Fm+) koji omogućuje brzinu prijenosa od 1Mbit/s

2012. – verzija 4 standarda koja je donijela veće promjene, do 5 Mbit/s brzine, te za „dana“ i „clock“ liniju nisu potrebni „pull-up“ otpornici.



Slika 4. Primjer spoja I²C sabirnice

Tipični integrirani sustav se sastoji od jednog ili više mikrokontrolera i perifernih uređaja (memorije, konverteri, proširivači ulazno-izlaznih pinova, LCD driveri, senzori, matrični prekidači, itd.). Zbog potrebe da komunikacija između njih bude što jeftinija i što manje kompleksna. Također, način komunikacija mora biti prilagodljiv, kako sporiji uređaji ne bi usporavali samu komunikaciju. Zbog postavljenih uvjeta, potrebna je serijska sabirnica. Definiranje sabirnice uvjetuje definiranje spajanja na sabirnicu, protokole, formate, adresiranje i tip adresa koje se koriste te se samim time definiraju pravila na sabirnici. I²C sabirnica je odgovor na postavljene uvjete.

I²C sabirnica koristi dvije žice: serijski podaci (SDA) i serijski takt (SCL). Svi uređaji, bili oni upravljači ili izvršitelji, su spojeni na sabirnicu sa te dvije žice (plus linija za napajanje i uzemljenje). Svaki uređaj može biti primatelj, pošiljalatelj ili oboje, ali ne istovremeno. Potrebno je napraviti promjenu moda uređaja u pitanju. Neki uređaji su upravljači: oni generiraju takt na sabirnici i započinju komunikaciju. Drugi uređaji su izvršitelji: oni odgovaraju na naredbe. Kako bi se ostvarila komunikacija s određenim uređajem, svaki izvršitelj mora imati adresu koja je jedinstvena na sabirnici. Upravljači na I²C sabirnici (obično mikrokontroleri) ne trebaju imati vlastitu adresu jer nijedan izvršitelj ne šalje naredbe upravljaču.

2.1.1. Terminologija

pošiljatelj - uređaj koji šalje podatke na sabirnici

primatelj - uređaj koji prima podatke na sabirnici

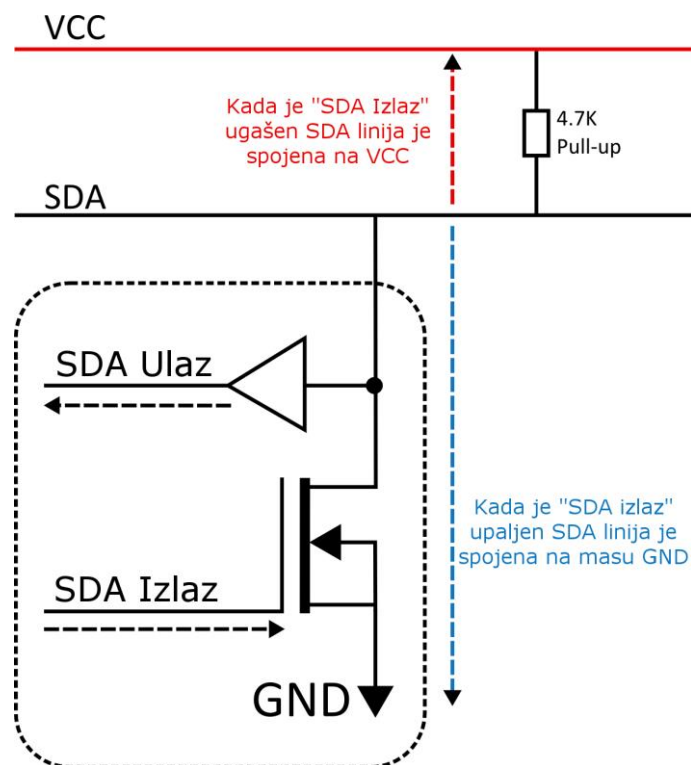
upravljač („Master“) - uređaj koji generira takt na sabirnici, započinje komunikaciju, šalje I²C naredbe i zaustavlja komunikaciju

izvršitelj („Slave“) - uređaj koji sluša na sabirnici i kojemu se upravljač obraća

Na sabirnici je dozvoljeno imati više izvršitelja, a i više upravljača. U slučaju kada je na sabirnici više upravljača postupkom arbitriranja se sprječava kolizija između njih tj. Samo se jednom upravljaču dodjeljuje pravo na upravljanje sabirnicom.

2.1.2. Signali na sabirnici

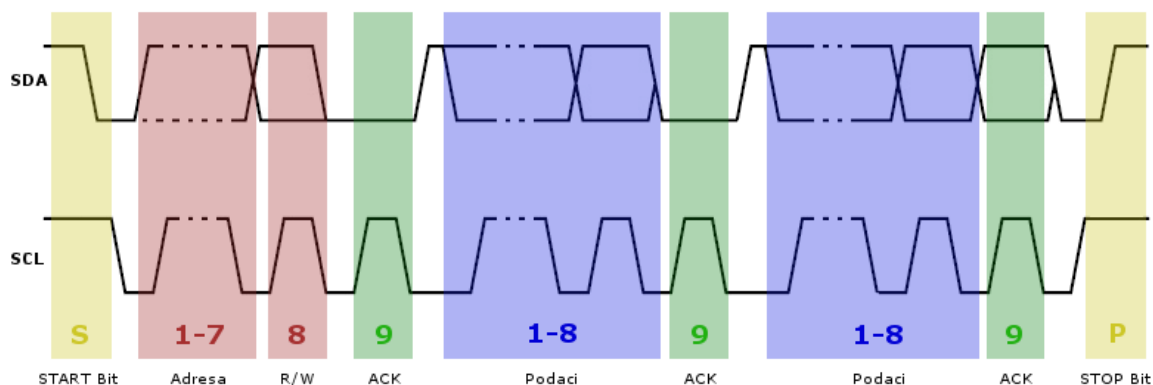
Oba signala (SDA i SCL) su dvosmjerna. Obje žice su spojene sa „pull-up“ otpornicima na napajanje sabirnice, što znači da kada je sabirnica slobodna (nitko ne komunicira na sabirnici), oba signala su na pozitivnoj naponskoj razini (naponska razina napajanja). Svi uređaji moraju imati „open-collector“ ili „open-drain“ pinove. Uzemljenje („pull down“) predstavlja aktivaciju linije (logički 1). Broj uređaja na jednoj sabirnici je skoro neograničen, jedini uvjet je da kapacitivnost cijele sabirnice ne prelazi 400 pF. Zbog toga što naponska razina logičke 1 ovisi o naponu napajanja, ne postoji standardni napon za sabirnicu, što omogućava veću fleksibilnost u raznolikosti uređaja na sabirnici.



Slika 5. Open-drain konfiguracija SDA pina kod I²C protokola

2.1.3. Serijski prijenos podataka

U svakom pulsu takta prenosi se jedan bit podatka. SDA linija se može mijenjati samo kada je SCL linija u niskoj naponskoj razini, kada je SCL linija u visokoj naponskoj razini, podaci trebaju biti stabilni.



Slika 6. Ciklus prijenosa podataka prema I²C protokolu

2.1.4. START i STOP stanje

Svaka I²C naredba koju šalje upravljač mora započeti sa START stanjem, a završiti sa STOP stanjem. Za oba navedena stanja, SCL linija mora biti u visokoj naponskoj razini. Prijelaz iz visokog u nisko naponsko stanje SDA linije signalizira START stanje, a prijelaz iz niskog u visoko naponsko stanje SDA linije signalizira START stanje. Poseban slučaj je ponovljeni START, gdje se ponovno signalizira START stanje, bez slanja STOP stanja. Nakon slanja START signala, sabirnica se tretira kao zauzeta i ne može ju koristiti drugi upravljač, dok god se ne pošalje STOP signal. Nakon START signala, upravljač može generirati ponovljeni START signal, koji je istovjetan normalno START signalu i obično je popraćen adresom I²C izvršitelja.

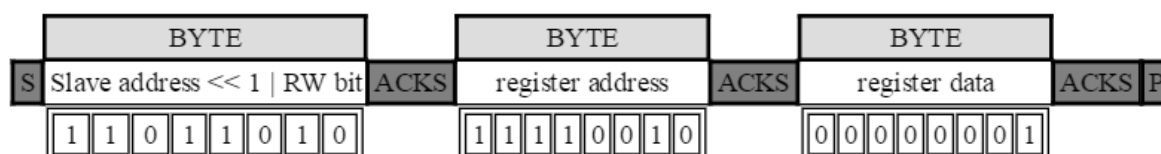
Podaci na I²C sabirnici se prenose u paketima veličina 8 bitova (1 bajt). Ne postoji ograničenje na broj poslanih bajtova, ali je uvjet da je nakon svakog poslanog bajta potrebno primiti potvrdni bit („Acknowledge“). Upravljač mora generirati takt za sve podatkovne bitove, uključujući i potvrdni bit. Ako izvršitelj ne potvrdi prijenos podataka, to znači da više nema podataka na sabirnici ili da uređaj još nije spreman za prijenos podataka. Upravljač mora generirati STOP signal ili ponovljeni START.

2.1.5. Komunikacija s izvršiteljima sa 7-bitnim adresama

Svaki izvršitelj na sabirnici mora imati jedinstvenu 7-bitnu adresu. Uređaji dolaze tvornički programirani s adresom ili sa par unaprijed programiranih adresa koje se mogu odabrati uzemljenjem određenih pinova. Komunikacija START signalom, nakon kojeg slijedi slanje 7-bitne adrese izvršitelja i bitom za određivanje hoće li se podaci čitati ili pisati. Ako je taj bit 0, upravljač šalje podatke, a ako je bit 1, upravljač čita podatke s izvršitelja. Nakon što je poslana adresa izvršitelja i bit koji označava čitanje ili pisanje, upravitelj može nastaviti s čitanjem s izvršitelja ili pisanjem na isti. Komunikacija prestaje slanjem STOP signala, koji istovremeno signalizira da je sabirnica slobodna.

Tablica 1. Popis posebnih adresa definiranih I²C protokolom

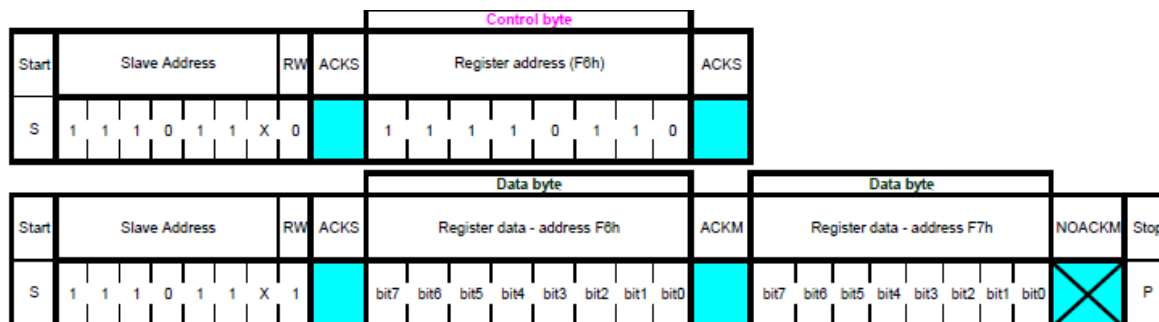
Adresa izvršitelja	R/W bit	Opis
0000 000	0	Adresa za opći poziv
0000 000	1	START bajt
0000 001	X	CBUS adresa
0000 010	X	Rezervirano za različite formate sabirnice
0000 011	X	Rezervirano za buduću upotrebu
1111 1XX	X	Rezervirano za buduću upotrebu
1111 0XX	X	10-bitno adresiranje izvršitelja



Slika 7. Primjer odabira izvršitelja i registra na izvršitelju te pisanje u registar.

U ovom primjeru, adresa izvršitelja je 0x76, koja se mora pomaknuti lijevo za jedan bit kako bi se napravilo mjesta za RW bit. RW bit je postavljen kao 0, što znači da upravitelj želi pisati podatke u izvršitelja. Nakon potvrde od izvršitelja, upravljač šalje adresu registra u koji želi pisati. U ovom slučaju je to 0xF2. Ponovnom potvrdom od izvršitelja, upravljač šalje podatke koji se upisuju u odabrani registar, u ovom slučaju 0x1. Izvršitelj ponovno potvrđuje primitak podataka i upravitelj generira STOP signal, kojim se završava komunikacija i oslobađa sabirnica.

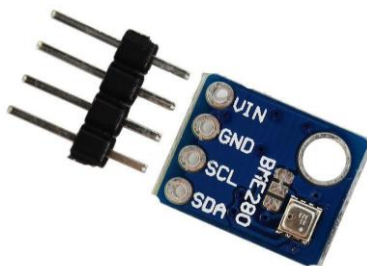
U slučaju kada upravljač želi očitati više bajtova s izvršitelja, memorijska adresa na izvršitelju se automatski uvećava te je zbog toga moguće u jednoj komunikaciji očitati više od jednog bajta.



Slika 8. Primjer ištavanja više bajtova s početnim registrom na izvršitelju

2.2. BME280 Senzor

BME280 je nova generacija senzora izrađenih u Boschu. Sam modul je kombinacija senzora tlaka, vlage i temperature na jednom čipu. Karakterizira ga vrlo visoka točnost, odaziv i mala potrošnja u odnosu na cijenu. Samim time se nameće kao idealan izbor na našem sustavu.



Slika 9. BME280 Senzor

Tablica 2. Karakteristike BME280 senzora

Dimenzije	2,5 mm x 2,5mm x 0,93 mm
Komunikacijsko sučelje	I ² C komunikacija (do 3.4 MHz) SPI komunikacija (do 10 MHz)
Napajanje	V _{DD} glavno napajanje: 1,71 V – 3,6 V V _{DDIO} napajanje sučelja: 1,2 V – 3,6 V
Potrošnja	Minimalno 3,6 μA , maksimalno 27 mA
Područje rada	Temperature od -40 °C do +85 °C Vlažnost od 0% do 100% Tlak od 300 hPa do 1100 hPa

Primjena senzora:

- kontrola ventilacije i klima uređaja
- očitavanja i mjerenja na meteorološkim stanicama
- u raznim medicinskim uređajima

Sam senzorski modul je malih dimenzija i pogodan je za ugradnju u uređaje kao što su mobiteli, GPS uređaji ili satovi. Podržava komunikaciju preko I²C i SPI sučelja.

Na raspolaganju su tri stanja rada senzora:

- „sleep mode“
- „normal mode“
- „forced mode“

2.2.1. „Sleep mode“

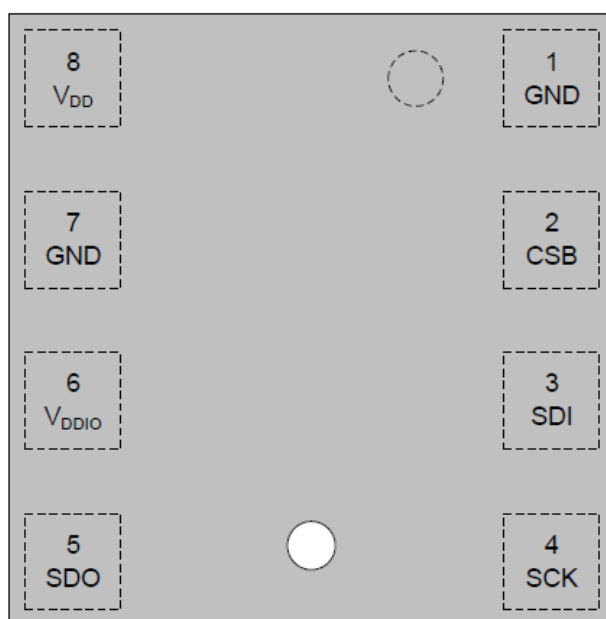
„Sleep mode“ je početno stanje senzora nakon njegovog pokretanja. U tom modu se ne vrše nikakva mjerenja i potrošnja energije je najmanja. Svi registri su dostupni i nema nikakvih ograničenja na samom sučelju.

2.2.2. „Forced mode“

U tome modu se vrši samo jedno mjerenje, nakon čega se senzor vraća u sleep mode. Rezultati mjerenja se dobivaju iz registra. Ovaj mod je preporučljiv gdje nisu česti intervali mjerenja kako bi se uštedila energija (baterija).

2.2.3. „Normal mode“

U ovom modu se konstantno mijenjaju ciklusi mjerenja i ciklusi neaktivnosti. Vremenski interval ciklusa se može mijenjati između 0,5 i 1000 ms.



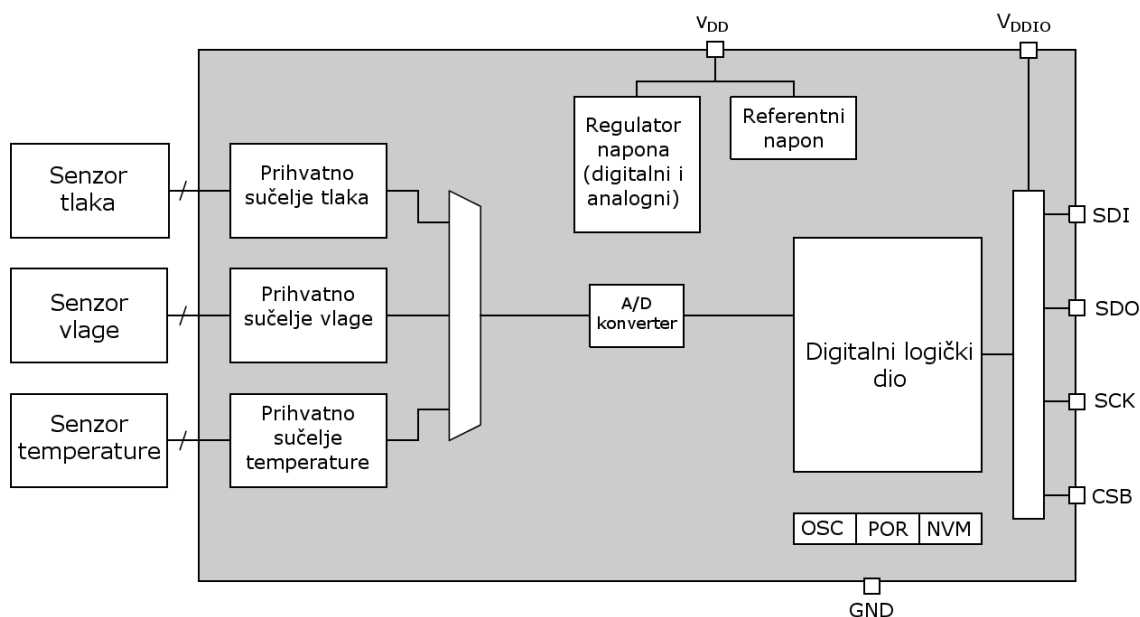
Slika 10. Raspored pinova BME280

Tablica 3. Oznake pinova BME280

Pin	Oznaka	Naziv
1	GND	GND
2	CSB	Chip Select
3	SDI	Serial data input
4	SCK	Serial clock input
5	SDO	Serial data output
6	V _{DDIO}	Digital interface supply
7	GND	GND
8	V _{DD}	Analog interface supply

Pinovi 1 i 7 služe sa spajanje na masu, dok su 6 i 8 napajanje modula V_{DD} i V_{DDIO}. V_{DD} je glavno napajanje za sve unutarnje analogne i digitalne funkcionalne blokove, dok je V_{DDIO} posebno napajanje koje se koristi za digitalno sučelje. V_{DD} se također koristi kao referentna naponska točka kako bi mjerenje bilo točno u slučaju naponskih oscilacija na ulazu. Pinovi 3 i 4 služe sa komunikaciju preko I²C sučelja, gdje je pin 3 za slanje podataka, a

pin 4 za takt. Pinove 2 i 5 ne koristimo u I²C komunikaciji, nego je uz pinove 3 i 4 služe za komunikaciju preko SPI sučelja.



Slika 11. Blok dijagram BME280

Iz navedenog blok dijagrama možemo detaljnije sagledati strukturu samog senzora. Gledano s lijeve strane imamo zasebne osjetne elemente za temperaturu, tlak i vlažnost. Svaki taj element je spojen na vlastito analogno sučelje koje prihvaća podatke i šalje ih dalje kroz A/D pretvornik gdje se pretvaraju u digitalni signal.

Senzor ima ugrađen POR („Power On Reset“) generator koji resetira logički dio i registre nakon što VDD i VDDIO dosegnu svoju najmanju razinu. U „non-volatile“ memoriju (NVM) su tvornički isprogramirani kalibracijski koeficijenti koji se koriste za precizni izračun temperature, tlaka i vlage.

Na sljedećoj slici je dan skraćeni pregled memorije senzora. Kalibracijski koeficijenti se nalazi na memorijskim lokacijama od 0x88 do 0xA1 i od 0xE1 do 0xF0. Svi registri su duljine jednog bajta (8 bitova).

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state	
hum_lsb	0xFE	hum_lsb<7:0>								0x00	
hum_msb	0xFD	hum_msb<7:0>								0x80	
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00	
temp_msb	0xFA	temp_msb<7:0>								0x80	
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0	0x00
press_lsb	0xF8	press_lsb<7:0>								0x00	
press_msb	0xF7	press_msb<7:0>								0x80	
config	0xF5	t_sb[2:0]			filter[2:0]			spi3w_en[0]		0x00	
ctrl_meas	0xF4	osrs_t[2:0]			osrs_p[2:0]			mode[1:0]		0x00	
status	0xF3	measuring[0]				im_update[0]				0x00	
ctrl_hum	0xF2					osrs_h[2:0]				0x00	
calib26..calib41	0xE1...0xF0	calibration data								individual	
reset	0xE0	reset[7:0]								0x00	
id	0xD0	chip_id[7:0]								0x60	
calib00..calib25	0x88...0xA1	calibration data								individual	

Registers:	Reserved registers	Calibration data	Control registers	Data registers	Status registers	Chip ID	Reset
Type:	do not change	read only	read / write	read only	read only	read only	write only

Slika 12. Mapa memorije senzora BME280

Tablica 4. Memorijske lokacije i tipovi podataka potrebnih kalibracijskih koeficijenata

Adresa registra	Sadržaj registra	Tip podatka
0x88 / 0x89	dig_T1 [7:0] / [15:8]	unsigned short
0x8A / 0x8B	dig_T2 [7:0] / [15:8]	signed short
0x8C / 0x8D	dig_T3 [7:0] / [15:8]	signed short
0x8E / 0x8F	dig_P1 [7:0] / [15:8]	unsigned short
0x90 / 0x91	dig_P2 [7:0] / [15:8]	signed short
0x92 / 0x93	dig_P3 [7:0] / [15:8]	signed short
0x94 / 0x95	dig_P4 [7:0] / [15:8]	signed short
0x96 / 0x97	dig_P5 [7:0] / [15:8]	signed short
0x98 / 0x99	dig_P6 [7:0] / [15:8]	signed short
0x9A / 0x9B	dig_P7 [7:0] / [15:8]	signed short
0x9C / 0x9D	dig_P8 [7:0] / [15:8]	signed short
0x9E / 0x9F	dig_P9 [7:0] / [15:8]	signed short
0xA1	dig_H1 [7:0]	unsigned char
0xE1 / 0xE2	dig_H2 [7:0] / [15:8]	unsigned short
0xE3	dig_H3 [7:0]	unsigned short

0xE4 / 0xE5 [3:0]	dig_H4 [11:4] / [3:0]	signed short
0xE5[7:4] / 0xE6	dig_H5 [3:0] / [11:4]	signed short
0xE7	dig_H6	signed char

Cijela memorija senzora, od memorijske lokacije 0x88 do 0xFE se iščitava u jednom komunikacijskom ciklusu¹ te se sprema u strukturu na sljedećoj slici. Jedan od razloga je čisto tehnički: kod stalnog očitavanja temperature, vlage i tlaka, može doći do miješanja podataka između mjerenja jer su podaci raspoređeni kroz par registara u memoriji. Drugi razlog je čisto radi lakšeg pridruživanja vrijednosti varijablama koje se koriste kasnije. Struktura je pakirana, kako bi se očitavanje iz memorije moglo točno memorijski samo staviti u strukturu sa „memcpy“. Varijable p1, p2 i p3 su varijable koje se koriste za popunjavanje mjesta nedefiniranih registara, kako bi se sve memorijski lokacije mogle kopirati 1:1 direktno u strukturu.

Nakon kopiranja podataka u strukturu, varijablama se lako može pristupiti pristupanjem članovima strukture.

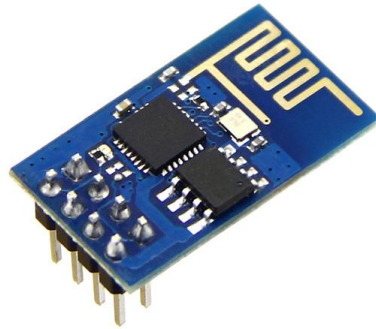
¹ Komunikacijskim ciklusom se smatra sva komunikacija između početnog START stanja i završnog STOP stanja

```
typedef struct __attribute__((packed)) {  
  
    //constants from registers  
    uint16_t dig_T1;  
    int16_t dig_T2;  
    int16_t dig_T3;  
    //constants from registers  
    uint16_t dig_P1;  
    int16_t dig_P2;  
    int16_t dig_P3;  
    int16_t dig_P4;  
    int16_t dig_P5;  
    int16_t dig_P6;  
    int16_t dig_P7;  
    int16_t dig_P8;  
    int16_t dig_P9;  
    //constants from registers  
    uint8_t dig_H1;  
    uint8_t p1[45];  
    uint8_t chip_id;  
    uint8_t p2[16];  
    //uint8_t unknown[62]; //padding bits from 0xA2 to 0xE0  
    uint8_t reset;  
    int16_t dig_H2;  
    uint8_t dig_H3;  
    int16_t dig_H4;  
    int16_t dig_H5;  
    int8_t dig_H6; //0xE7  
    uint8_t p3[15];  
    uint8_t press[3];  
    uint8_t temp[3];  
    int16_t adc_H;  
    //T readout from sensor  
    int32_t adc_T, adc_P, t_fine;  
    //return variables  
    uint32_t H,P;  
    float T;  
}calibration_t;
```

Slika 13. Struktura koja se koristi za iščitavanje memorije iz senzora

2.3. ESP8266 mikrokontroler

ESP8266 je mikrokontroler s bežičnim modulom kojeg karakterizira izrazito niska cijena. Mikrokontroler se može koristiti zasebno ili kao bežični modul za komunikaciju na mikrokontrolerima koji nemaju integrirani modul. Podržava komunikaciju preko serijskog sučelja.



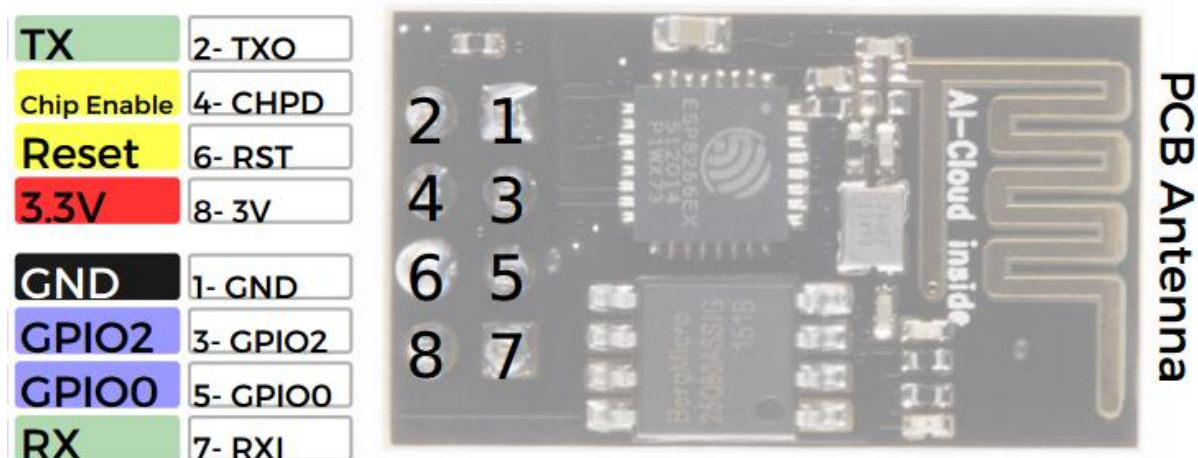
Slika 14. ESP8266 mikrokontroler

Karakteristike ESP8266 modula:

- 32-bit RISC CPU 80 MHz
- 64 kB RAM-a
- podrška za komunikaciju preko serijskog, SPI ili I²C sučelja
- 10-bit A/D pretvornik
- 512 kB flash memorije
- podržava 802.11 b/g/n protokol
- podržava P2P, AP
- integrirana podrška za TCP/IP protokol

Primjena:

- automatizacija procesa u industriji
- prijenosna elektronika (satovi, mobiteli)
- bežična komunikacija sa sensorima
- bežični modul u mrežnim kamerama



Slika 15. Raspored pinova ESP8266

Tablica 5. Oznake pinova ESP8266

Pin	Oznaka	Naziv
1	GND	Masa
2	TX	Tx pin serijskog sučelja
3	GPIO2	Opći ulazno-izlazni pin
4	CH_PD	„Chip power down“
5	GPIO0	Opći ulazno-izlazni pin
6	RST	Reset pin
7	RX	Rx pin serijskog sučelja
8	VCC	Napajanje

Kako bi ušteda energije bila maksimalna modul može raditi u sljedećim stanjima:

„OFF“ – CH_PD pin je u niskoj razini napona, svi registri su prazni, RTC ne radi

„Deep Sleep“ – RTC je aktivan i drži spremljene osnovne podatke o mreži, svi ostali dijelovi čipa su neaktivni

„Sleep“ – RTC je uključen, oscilator je neaktivan, bilo kakav vanjski utjecaj će uključiti čip

„Wake“ – tzv. Buđenje čipa, oscilator i PLL se aktiviraju iz „sleep“ stanja

„ON“ – svi dijelovi čipa su aktivni, radi na visokom taktu, spreman je za rad

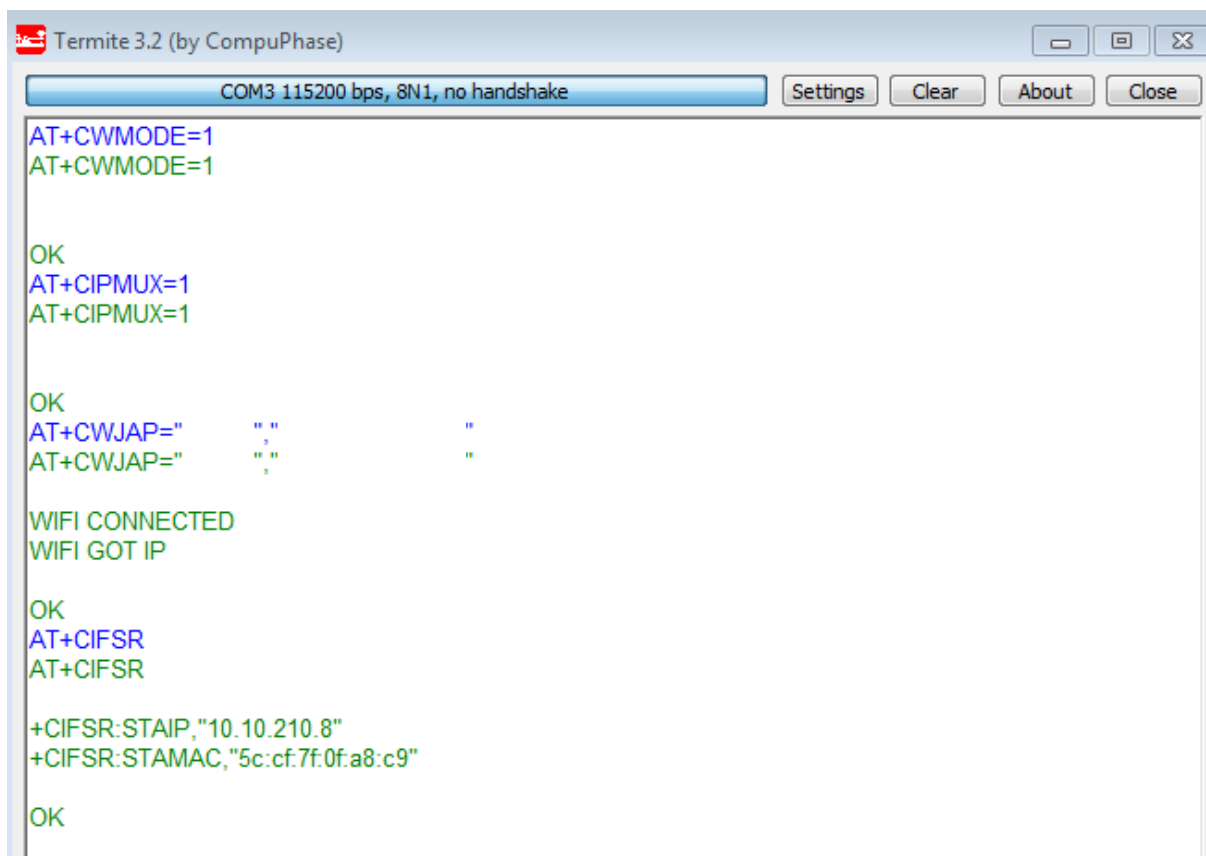
Svaki modul dolazi sa programiranim setom AT naredbi sa kojima ga možemo odmah spojiti na mrežu. Nekoliko primjera takvih naredbi je u tablici.

Tablica 6. Primjer AT naredbi na ESP8266

Funkcija	Naredba	Odgovor
Spremnost za rad	AT	OK
Restart	AT+RST	OK (System ready, Vendor: www.ai-thinker.com)
Prikaz dostupnih bežičnih mreža	AT+CWLAP	AT+CWLAP +CWLAP:(4,"RocheFortSurLa c", - 38,"70:62:b8:6f:6d:58",1) +CWLAP:(4,"LiliPad2.4", - 83,"f8:7b:8c:1e:7c:6d",1) OK
Spajanje na bežičnu mrežu	AT+CWJAP="SSID","Password"	Query AT+CWJAP?+CWJAP:"Roc hefortSurLac" OK

Odspajanje s wireless mreže	AT+CWQAP	OK
Očitanje IP adrese	AT+CIFSR	AT+CIFSR 192.168.1.5 OK
WiFi način rada	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
Postavljanje TCP/UDP konekcije	Za CIPMUX=0 AT+CIPSTART = <type>,<addr>,<port> Za CIPMUX=1 AT+CIPSTART= <id><type>,<addr>,<port>	Id=0-4 Type=TCP/UDP Addr=IP adresa Port=port
Slanje podataka	Za CIPMUX=0 AT+CIPSEND=<length> Za CIPMUX=1 AT+CIPSEND=<id>,<length>	
Zatvaranje TCP/UDP konekcije	AT+CIPCLOSE=<id> Ili AT+CIPCLOSE	

Kako bi ispitali funkcionalnost ESP8266 mikrokontrolera obaviti ćemo testno spajanje na bežičnu mrežu. Mikrokontroler smo spojili na računalo uz pomoć „USB to Serial“ adaptera i komunicirali s njim preko terminala. Dijagram spoja je prikazan na slici .



```
Termit 3.2 (by CompuPhase)
COM3 115200 bps, 8N1, no handshake
Settings Clear About Close

AT+CWMODE=1
AT+CWMODE=1

OK
AT+CIPMUX=1
AT+CIPMUX=1

OK
AT+CWJAP=" " " "
AT+CWJAP=" " " "

WIFI CONNECTED
WIFI GOT IP

OK
AT+CIFSR
AT+CIFSR

+CIFSR:STAIP,"10.10.210.8"
+CIFSR:STAMAC,"5c:cf:7f:a8:c9"

OK
```

Slika 16. Primjer spajanja na bežičnu mrežu i provjera dobivene IP adrese

Naredba `AT+CWMODE=1` stavlja ESP8266 u „Station“ mode, gdje se ponaša kao klijent koji se spaja na dostupnu bežičnu mrežu. `AT+CIPMUX=1` naredba omogućuje više istovremenih TCP ili UDP konekcija. Nakon toga, naredbi `AT+CWJAP` dajemo argumente za spajanje na bežičnu mrežu: SSID i lozinku. ESP8266 javlja je li se uspješno spojio i je li mu dodijeljena IP adresa. Naredbom `AT+CIFSR` možemo dobiti podatke o dobivenoj IP adresi i MAC adresu samog ESP-a.

Naredbom AT+CIPSTART otvaramo TCP/UDP konekciju sa sljedećim parametrima: tip konekcije je TCP, dana IP adresa je adresa servera s kojim se ostvaruje konekcija i na kraju je dan i port servera na koji se spajamo. AT+CIPSEND označava koliko ćemo bajtova odjednom primiti od strane servera. Podaci će stizati sve dok ih server šalje, ali u komadima veličine 47 bajtova. Nakon toga, ESP izvještava o uspješnosti ostvarivanja konekcije. Ako je konekcija ostvarena, sljedeći korak je poslati GET zahtjev serveru. Nakon zahtjeva, ESP ispisuje primljene podatke od servera. U ovom slučaju, HTML izvorni kod zatražene stranice.

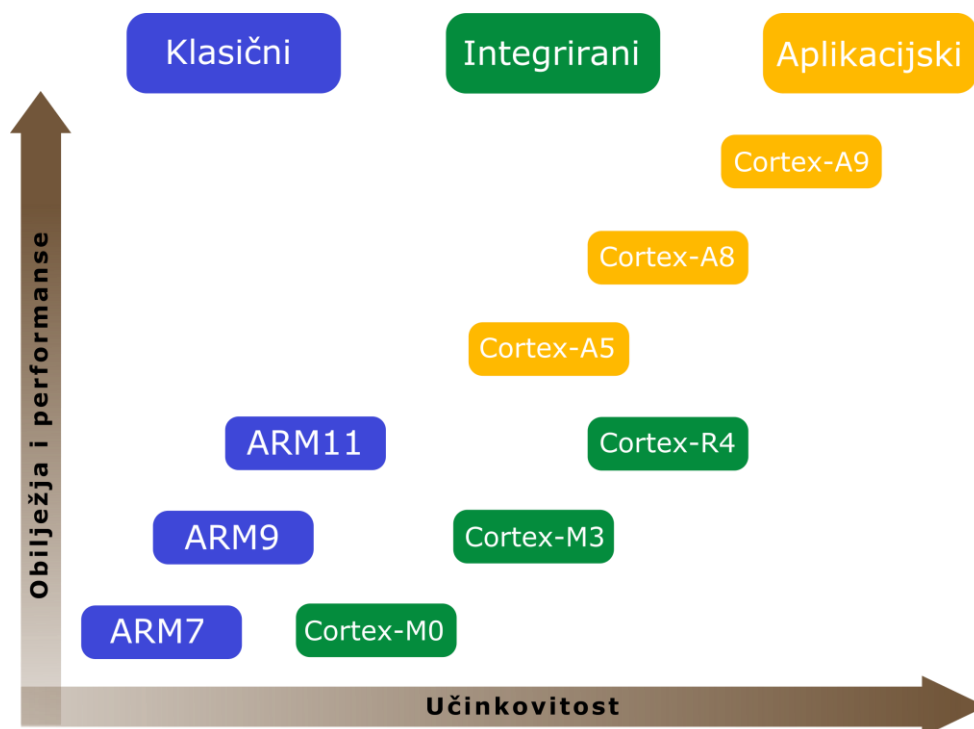
2.4. STM32F103 mikrokontroler

STM32F103 je mikrokontroler baziran na ARM Cortex-M3 procesoru frekvencije 72 MHz. Primjena ovog mikrokontrolera je vrlo široka. Koristi se u raznim prijenosnim uređajima (GPS, računalna periferija), printerima, skenerima, alarmnim sustavima te u industriji kao PLC. Ovaj mikrokontroler će služiti kao posrednik pri slanju podataka sa senzora i prihvatanju naredbi sa bazne stanice.

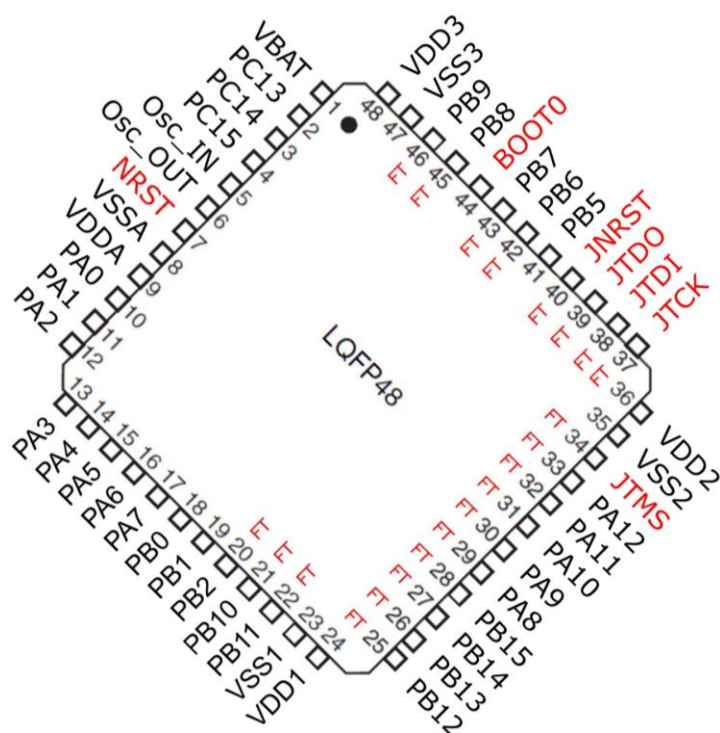
Prvi ARM procesori su se pojavili 80-tih godina prošlog stoljeća, a predstavljali su jednostavniju i jeftiniju alternativu tadašnjim 8-bitnim procesorima koji su imali kompleksniji set naredbi (CISC). Nakon izlaska prvih ARM procesora tržište je prihvatilo novu i jeftiniju alternativu te danas ta arhitektura zauzima ukupno 70% tržišta 32-bitnih procesora.

Prednosti ARM arhitekture:

- jednostavniji set naredbi
- manji broj tranzistora potreban za izradu
- modularna izvedba
- mala potrošnja energije



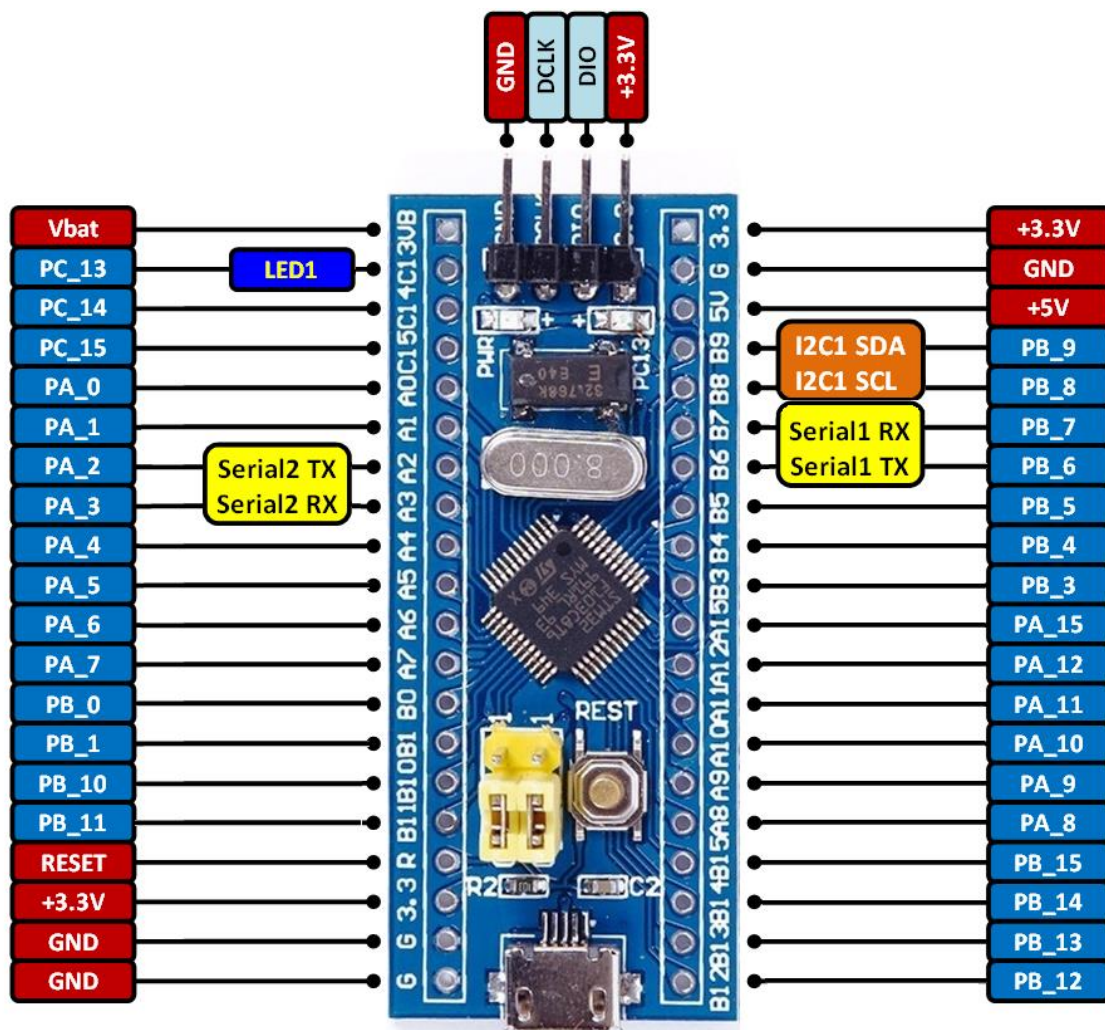
Slika 18. Podjela ARM procesora



Slika 19. Raspored pinova na 48-pinskoj verziji mikrokontrolera

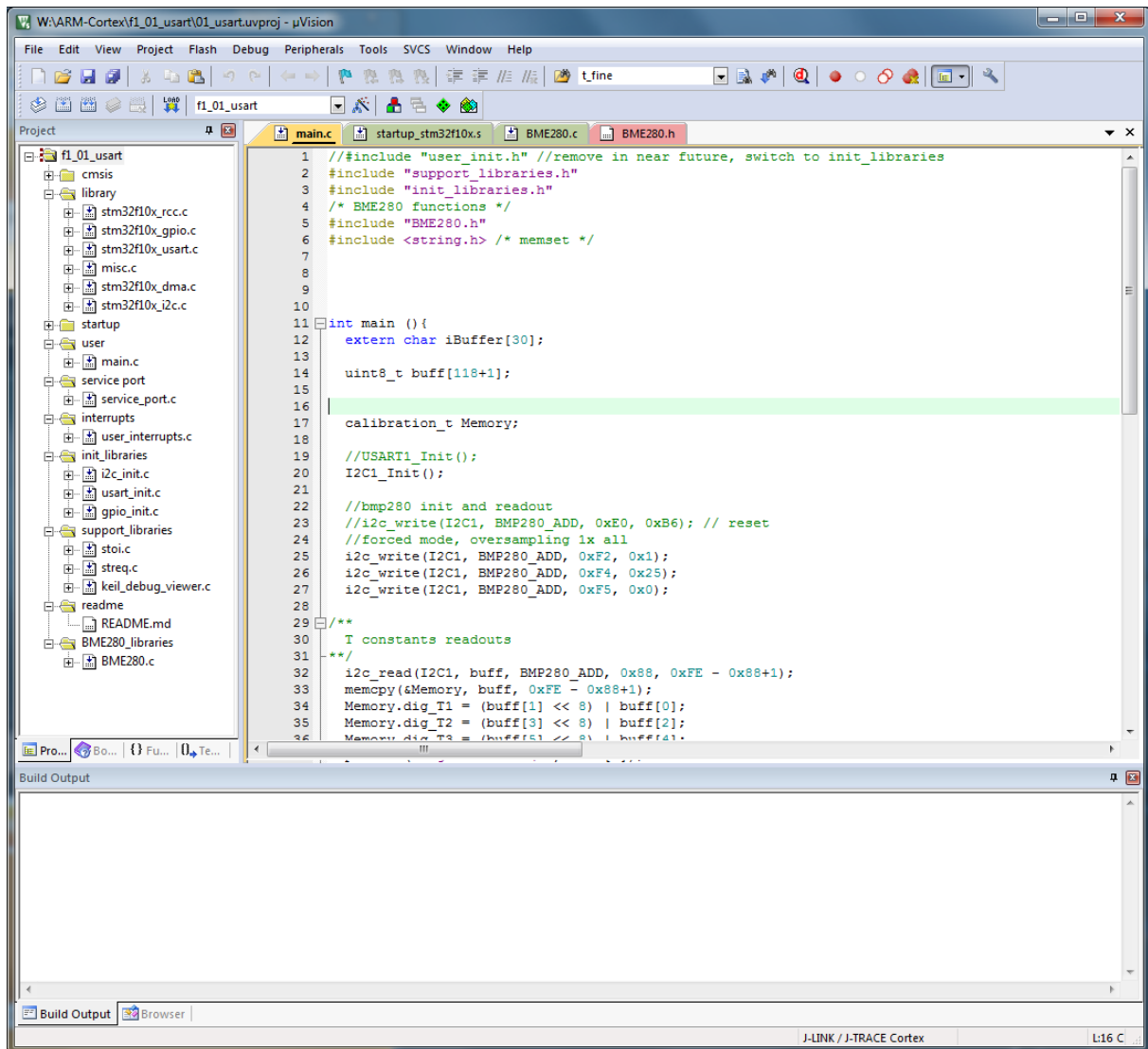
Na gornjoj slici je prikazan raspored pinova na 48-pinskoj verziji STM32F103C8 mikrokontrolera u LQFP kućištu („Low profile Quad Flat Package“). Sam mikrokontroler STM32F103 dolazi i u 36-pinskom kućištu.

Na sljedećoj slici je prikazana korištena razvojna pločica te su označeni pinovi koji su korišteni za senzorsku jedinicu. Serijski port pod brojem 1 se koristi za servisni terminal, koji je aktivan samo ako je netko spojen na njega. I²C pod brojem 1 se koristi za kontrolu sabirnice, u ovom slučaju senzora BME280. Serijski port pod brojem 2 se koristi za komunikaciju s ESP8266 modulom.



Slika 20. Korišteni pinovi na STM32F103C8 mikrokontroleru

Za razvoj samo programskoj dijela koristi se KEIL IDE integrirano razvojno okruženje. Na sljedećoj slici je prikazan izgled radnog okruženja KEIL IDE za navedeni projekt senzorske stanice. Na lijevoj strani su datoteke korištene u razvoju projekta, na desnoj strani je prikazan izvorni kod i kartice s otvorenim datotekama. Na vrhu se nalaze alatne trake pomoću koji se može odabrati mikrokontroler, na koji način se programira i „debugiranje“ koda koji se direktno izvršava na mikrokontroleru. U paketu sa razvojnim okruženjem dolazi i C prevodilac optimiziran za integrirana razvojna rješenja.



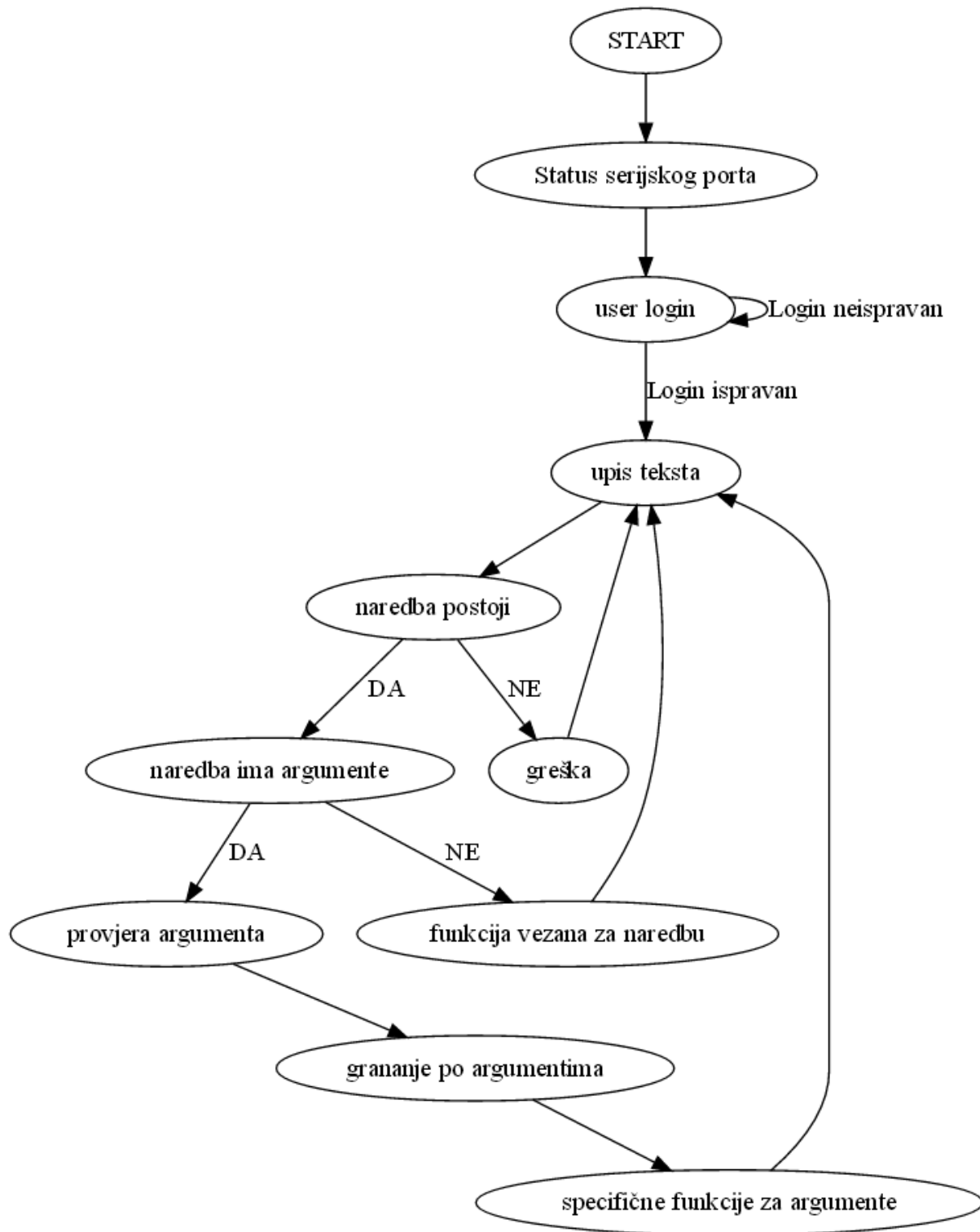
Slika 21. Glavni prozor KEIL integriranog okruženja



Slika 22. J-Link JTAG/SWD programator i debugger

Za programiranje samog mikrokontrolera koristi se J-Link programator i debugger tvrtke Segger koji služi za programiranje i „debugiranje“ programa uživo. JTAG podržava dva protokola: JTAG i SWD. U našem slučaju, koristi se SWD protokol („Single Wire Debugging“) zbog toga što zauzima manje pinova i na samoj razvojnoj pločici su već pripremljeni pinovi za taj protokol.

2.5. Servisni terminal



Slika 23. Dijagram toka rada servisnog terminala

Servisni terminal služi za dijagnostiku same senzorske jedinice i dostupan je samo tehničaru koji se mora fizički spojiti na senzorsku jedinicu i ulogirati se s lozinkom, kako bi mogao pristupiti dijagnostičkom dijelu senzorske jedinice. Nakon uspješnog ulogiravanja, na raspolaganju je broj naredbi koje prijavljuju stanje senzorske jedinice i njezinih perifernih dijelova. Izvještaji se prikazuju u prozoru terminala i služe dijagnostici kvara. Također, servisni terminal može izvijestiti o stanju baterije i njezinoj povijesti upotrebe.

3. POSTAVLJANJE LAMP SERVERA

LAMP je osnovni paket programa potrebnih za normalan rad web servera, a sastoji se od Linux distribucije (u našem slučaju Ubuntu 16.04), Apache HTTP servera, MySQL baze podataka i PHP kao programskog jezika. Ovakva kombinacija programa je postala popularna zbog otvorenog koda i u potpunosti su besplatni za korištenje. Ovaj programski paket je korišten na glavnoj baznoj stanici za prihvatanje podataka sa senzora i slanje naredbi. Slijedi kratak opis svakog od tih modula.

3.1. Linux OS

Linux je operativni sustav baziran na UNIX-u, a zamišljen je kao operativni sustav koji je besplatan za korištenje i u potpunosti otvorenog koda. Linux je osmislio 1991. Linus Torvalds za 32-bitna računala, a od tada sustav je prebačen na mnoge arhitekture i postao je najrasprostranjeniji sustav na serverima, „mainframe“ računalima i superračunalima, dok u isto vrijeme drži samo 3,2% tržišta osobnih računala zbog manjka softverske podrške koju su korisnici navikli na Windows operativnim sustavima. Tek sa probojem Android sustava koji je baziran na Linuxu vidimo porast broja korisnika Linux operativnog sustava te ujedno porast korisničke podrške. U budućnosti Linux bi mogao preuzeti većinu korisnika zbog ubrzanog napretka. Tipično sam Linux dolazi u distribucijama kako bi što više olakšalo korisnicima prijelaz i instalaciju.

Neke od distribucija Linuxa :

- RedHat – komercijalna distribucija, jedna od najstarijih distribucija
- Fedora – temelji se na RedHat distribuciji, služi kao osnova komercijalne distribucije
- Debian – distribucija sastavljena u potpunosti s „open source“ softwareom
- Ubuntu – baziran na debianu, za kućnu uporabu, jedna od najpopularnijih distribucija
- Gentoo – distribucija za najzahtjevnije Linux korisnike, instalacija zahtjeva prevođenje sustava iz izvornog koda
- Slackware – distribucija za Linux korisnike koji žele stabilan i pouzdan sustav
- Kali – baziran na Debianu, sadrži paket programa za etičko hakiranje i testiranje sigurnosti

Na našoj baznoj stanici koristili smo Ubuntu 16.04 distribuciju te smo na nju dodali web server i bazu podataka.

3.2. Apache server

Apache HTTP server je najrasprostranjeniji web server na svijetu, pretpostavka je da se nalazi na 46% svih web stranica na internetu. Nastao je 1995. godine, besplatan je za korištenje i otvorenog je koda. Dostupne su verzije za Linux i Windows operativni sustav. Postoje slične inačice web servera kao što su nginx, IIS i GWS.

3.2.1. Instalacija i konfiguracija na Ubuntu OS

Kako je Apache već dostupan u Ubuntu repozitoriju potrebno je otvoriti terminal i utipkati slijedeću naredbu za instalaciju:

```
sudo apt-get install apache2
```

Nakon instalacije server konfiguriramo preko konfiguracijskih datoteka koje su u principu obični tekst koji izmjenjujemo po volji. Lokacija tih konfiguracijskih datoteka je `/etc/apache2`.

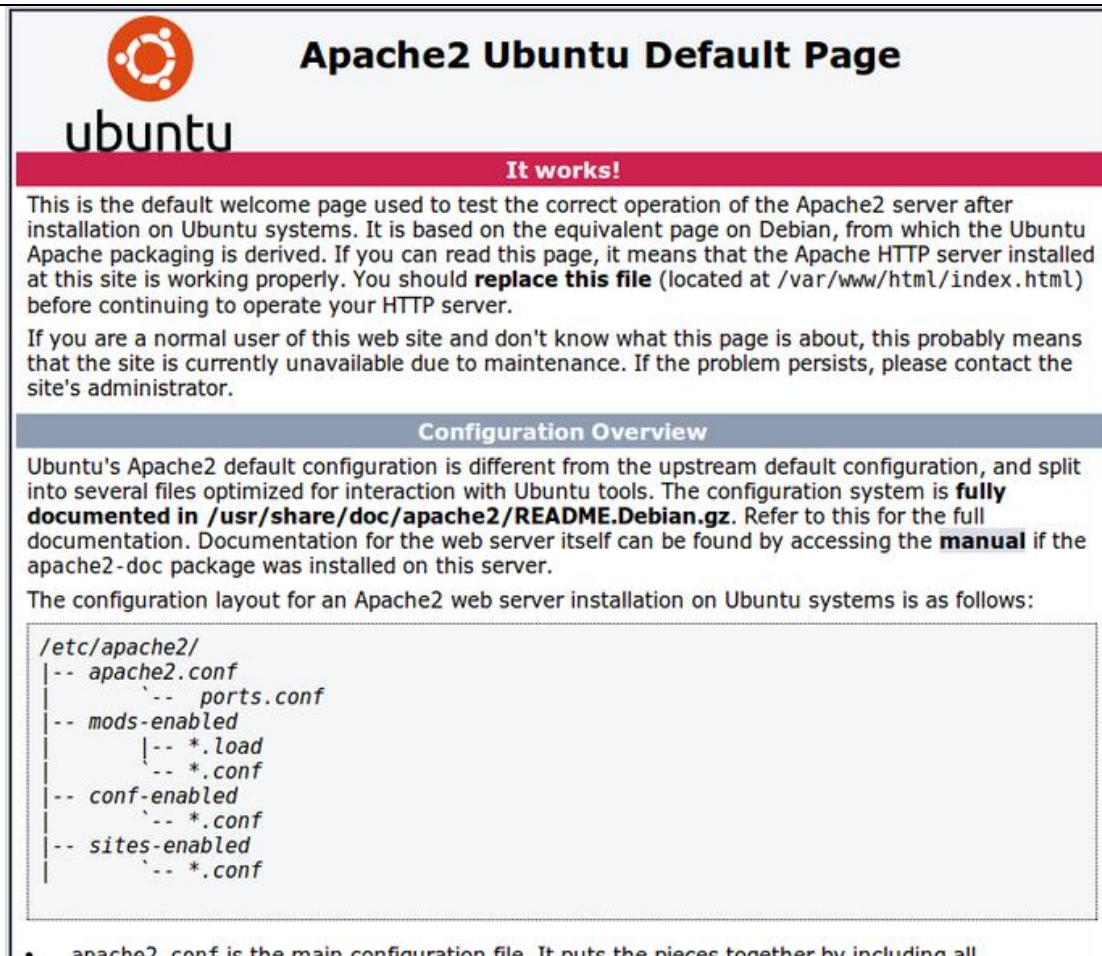
apache2.conf – glavna konfiguracijska datoteka, sva potrebna podešenja se rade u njemu

ports.conf – služi za podešenje portova na kojima server sluša

conf.d/ - mapa u kojoj namještamo specifične dijelove servera, npr. SSL konfiguraciju i sigurnosne postavke

httpd.conf – starija inačica glavne konfiguracijske datoteke, služi zbog povećanja kompatibilnosti

Ukoliko je sve iskonfigurirano kako treba, unosom adrese servera u web preglednik bi trebalo prikazati testnu stranicu prikazanu na slici.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all

Slika 24. Testna stranica apache servera

Nakon postavljanja web servera, zbog sigurnosti je potrebno iskonfigurirati firewall.

3.3. PHP i MySQL

PHP je programski jezik koji se izvodi na serverskoj strani, a služi za kreiranje interaktivnih web stranica. PHP je kratica od imena „Hypertext Preprocessor“. Vrlo je popularan zbog toga jer se može direktno upisivati kraj HTML koda, besplatan je i otvorenog je koda te ima podršku za rad s raznim bazama podataka kao što su MySQL te je ujedno u toj kombinaciji najrasprostranjeniji način izrade interaktivnih web stranica. Struktura i sintaksa samog koda je vrlo slična C programskom jeziku ili JavaScriptu.

MySQL je upravitelj bazama podataka koji se izvodi na serverskoj strani i pogodan je u izradi bilo manjih ili većih aplikacija, glavna karakteristika je mogućnost izvršavanja na velikom broju platformi te je također besplatan za upotrebu. Dizajniranje MySQL baze podataka smo odradili uz pomoć phpMyAdmin web sučelja prikazanog na slici.

```
<?php
$var1 = $_GET['temp'];
echo $var1 . "<br>";
$var2 = $_GET['node'];
echo $var2 . "<br>";
$var3 = $_GET['tlak'];
echo $var3 . "<br>";
$var4 = $_GET['vlag'];
echo $var4 . "<br>";

$servername = "localhost";
$username = "root";
$password = "XXXXXXXXXX";
$dbname = "data_table";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO main_table (node_humidity, node_name, node_pressure, node_temperature)
VALUES ('$var4', '$var2', '$var3', '$var1')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

?>
```

Slika 25. PHP stranica za prihvatanje podataka

Zadatak prihvatne stranice je da preko URL poveznice prihvaća poslano podatke i zapisuje ih u bazu podataka. Za unos su potrebna četiri podatka, a to su: temperatura, identifikacijski broj senzorske jedinice, tlak i vlaga. Peti parametar se automatski generira u bazi, a to je vremenska oznaka kada su podaci primljeni. Time se izbjegava potreba za vođenjem brige o vremenu na samoj senzorskoj jedinici te se to prepušta serveru, što omogućuje lakšu vremensku preglednost podataka, jer je server jedini zadužen za postavljanje vremena.


```

<?php
$servername = "localhost";
$username = "root";
$password = '██████████';
$dbname = "data_table";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT timestamp, node_name, node_temperature, node_pressure, node_humidity FROM main_table";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "<b>Vrijeme: </b>" . $row["timestamp"]. " | <b>Node: </b>" . $row["node_name"]. " | <b>Temperatura: </b>" . $row["node_temperature"]. " | <b>Tlak: </b>" . $row["node_pressure"]. " | <b>Vlaznost: </b>" . $row["node_humidity"] . "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

Slika 26. PHP stranica za dohvat podataka iz baze

Stranica za dohvat podataka dohvaća podatke iz baze i ispisuje ih. Koristi se samo za vizualnu provjeru upisanih vrijednosti.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'data_table'. The selected table is 'main_table'. The interface displays a table with the following data:

timestamp	node_name	node_temperature	node_pressure	node_humidity
2016-08-16 12:55:30	14	24	1014	66
2016-08-16 13:51:10	17	25	1013	75
2016-08-16 15:22:47	2	1	1011	21
2016-08-16 15:23:48	17	28	1011	65
2016-08-16 15:24:02	18	28	1011	65
2016-08-16 15:24:12	18	28	1011	55
2016-08-16 16:03:09	4	24	1008	98
2016-08-16 16:03:14	s4	24	1008	98
2016-08-16 16:04:58	senzor2	10	1018	55
2016-08-18 14:09:07	senzor2	10	1018	55

Slika 27. Direktno isčitavanje baze preko „phpMyAdmin“ sučelja

4. ZAKLJUČAK

Kao što osobna računala imaju svoj povijesni izvor u „mainframe“ računalima namijenjenima industriji i u istraživačke svrhe, tako pametni domovi svoj postanak duguju industrijskoj automatizaciji. Ova dva slučaja su slična i po još nečemu, industrijska automatizacija i kontrola proizvodno-tehnoloških procesa je skupa, ogromna infrastruktura i namijenjena isključivo industriji, a takvu ulogu su prvotno imala i „mainframe“ računala. Smanjenjem cijena, minijaturizacijom i širom dostupnošću sve većeg broja hardverskih komponenti koje zadovoljavaju i industriju i ljude koji se time bave iz hobija, došlo je do razvoja kućne automatizacije. Sustav je preslika industrijskih procesa, gdje se sam sustav sastoji od osjetilnih dijelova (senzori vlage, temperature, tlaka, protoka, svjetline, itd.), ali u puno manjem mjerilu: krojeno prema potrebama kućanstava. Svjedočimo ogromnom rastu popularnosti IoT („Internet of Things“) područja, gdje se umrežavaju razne senzorske i upravljačke jedinice kako bi se automatizacija uvela i u naše domove. Lakša kontrola i pregled potrošnje svakako pomaže u efikasnijem upravljanju vlastitim domom, gdje je najjednostavniji primjer kontrola grijanja i hlađenja. Poput industrijskih procesa, senzori i upravljačke jedinice komuniciraju sa centralnim serverom, koji služi kao HMI („human machine interface“) sučelje između cijelog sustava i samog korisnika. Zbog toga što su oba sustava tako bliski rođaci, pate i od nekih zajedničkih boljki. Najvažnija od njih je sama sigurnost prijenosa podataka i upravljanje. Industrijska automatizacija se često vodi principom „*Security through obscurity*“, gdje su implementacije često nestandardizirane i ovise o proizvođaču te se vežu na opremu koja je dostupna isključivo industrijskim korisnicima. Kod primjera kućne automatizacije dolazimo do istog problema oko sigurnosti, ali zbog drugih razloga. Mikrokontroleri i senzorsko-upravljačke jedinice često ni nemaju dovoljnu snagu za implementaciju nekih od sigurnosnih protokola pa je komunikacijski kanal između korisnika i samog sustava često izložen napadima. U posljednje vrijeme je vidljiv porast malicioznog preuzimanja IoT uređaja kako bi mogli sudjelovati u DDoS napadima. Samo polje kućne automatizacije je relativno mlado, ali vjerujem da će kao i sve industrije dostići svoju zrelost i poraditi na stvarima koje su manjkave.

LITERATURA

- [1] DAVOR BONAČI: I²C Sabirnica, Zagreb 2007.
- [2] I²C bus: <https://en.wikipedia.org/wiki/I2C>
- [3] TEXAS INSTRUMENTS, J. VALDEZ, J. BECKER: Understanding the I²C bus, 2015.
- [4] <http://www.realestatetodayblr.com/smart-home-technology-2016/>
- [5] BOSCH SENSORTEC datasheet: BME280 Combined humidity and pressure sensor, 2015.
- [6] ESP8266 datasheet: <http://www.esp8266.com/>
- [7] ESP8266 configuration: <https://nurdspace.nl/ESP8266>
- [8] ST ELECTRONICS datasheet : STM32F103x8 Medium-density performance line ARM, 2015.
- [9] Keil Embedded Development Tools for ARM: <https://www.keil.com/>
- [10] LAMP configuration: <https://www.digitalocean.com>