

# Pametna košnica

---

**Salopek, Stjepan**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:514755>

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

*Download date / Datum preuzimanja:* **2024-12-31**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**  
**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I**  
**INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**PAMETNA KOŠNICA**

**Diplomski rad**

**Stjepan Salopek**

**Osijek, 2020.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 28.09.2020.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za diplomski ispit**

<b>Ime i prezime studenta:</b>	Stjepan Salopek
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Računarstvo
<b>Mat. br. studenta, godina upisa:</b>	D-1011R, 23.09.2019.
<b>OIB studenta:</b>	83190660157
<b>Mentor:</b>	izv. prof. dr.sc. Josip Job
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	Izv.prof.dr.sc. Ratko Grbić
<b>Član Povjerenstva 1:</b>	izv. prof. dr.sc. Josip Job
<b>Član Povjerenstva 2:</b>	Dr. sc. Krešimir Romić
<b>Naslov diplomskog rada:</b>	Pametna košnica
<b>Znanstvena grana rada:</b>	<b>Procesno računarstvo (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	Potrebno je istražiti mogućnosti primjene tehnologija korišćenih u području interneta objekata te predložiti dizajn sustava za nadziranje stanja pčelinjih košnica. Tema rezervirana za: Stjepan Salopek
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene mentora:</b>	28.09.2020.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 30.09.2020.

Ime i prezime studenta:

Stjepan Salopek

Studij:

Diplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

D-1011R, 23.09.2019.

Turnitin podudaranje [%]:

8

Ovom izjavom izjavljujem da je rad pod nazivom: **Pametna košnica**

izrađen pod vodstvom mentora izv. prof. dr.sc. Josip Job

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

## SADRŽAJ

1. UVOD .....	1
1.1. Zadatak diplomskog rada.....	2
2. PREGLED PODRUČJA TEME .....	3
2.1. Apiloger.....	4
2.2. SMSVAGA.....	4
2.3. XLOG bee .....	5
2.4. OpenHiveScale .....	6
2.5. Bee hive monitoring .....	7
3. TEHNIČKI PREDUVIJETI I KORIŠTENE TEHNOLOGIJE .....	8
3.1. Arduino platforma .....	8
3.1.1. Mikroupravljač ATmega328.....	9
3.1.2. Arduino UNO R3.....	12
3.1.3. Arduino IDE .....	14
3.2 GSM/GPRS tehnologija .....	16
3.3. SIM800L modul .....	19
3.3.1. AT naredbe .....	22
3.4. Senzor mase.....	23
3.5. HX711 .....	27
3.6. DHT11 .....	28
3.7. LCD I2C .....	29
3.8. Li-ion baterija .....	31
3.8.1. Punjenje li-ion baterije.....	33
3.9. PCF85063A RTC .....	36
3.10. MySQL baza podataka .....	38
3.11. PHP.....	39
3.12. phpMyAdmin .....	39
4. PAMETNA KOŠNICA.....	40
4.1. Arduino programski primjer za mjerenje mase .....	46
4.2. Arduino programski primjer za mjerenje temperature i vlage.....	47
4.3. Arduino programski primjer za ispis na LCD ekran .....	48
4.4. Arduino programski primjer za postavljanje i ispis vremena.....	49
4.5. SQL programsko rješenje .....	50
4.6. PHP programsko rješenje .....	50
4.7. Arduino programsko rješenje .....	53
5. ISPITIVANJE MOGUĆNOSTI SUSTAVA .....	59
6. ZAKLJUČAK .....	63
LITERATURA.....	64
PRILOZI.....	69

## 1. UVOD

U današnje vrijeme razvoj i usluga IoT (*engl. Internet of Things*) uređaja raste u velikoj mjeri. Svakodnevno se mogu pronaći informacije o novim idejama koje izlaze na tržište i na neki način olakšavaju ljudski rad, bilo to financijski ili fizički. Tako je tehnologija pronašla svoje mjesto u raznim granama industrije i znanosti, od poljoprivrede pa sve do medicine. Tijekom višegodišnjeg iskustva u radu s pčelama su uočeni nedostaci koji bi se upravo mogli riješiti uvođenjem tehnologije i tako bi se uvelike smanjili financijski troškovi i povećala proizvodnja meda. Korištenjem pametne digitalne vage za košnice se drastično smanjuje broj posjeta košnicama te se šteti na vremenu i novcu. Osim toga se korištenjem odabranih senzora povećava produktivnost unutar košnice što rezultira povećanom prinosu meda zahvaljujući daljinskom nadzoru gdje pčelar pravovremenu bude obaviješten o trenutnoj situaciji unutar košnice. Na osnovu rezultata koje pčelar dobiva on može reagirati na vrijeme te premjestiti košnice na drugu lokaciju.

Nakon uvodnog poglavlja, u drugom poglavlju je napravljeno istraživanje na tržištu, te se opisuju karakteristike i mogućnosti sličnih proizvoda kao ovaj projekt. U trećem poglavlju su opisane korištene tehnologije, glavna komponenta projekta, a to je Arduino pločica sa ATmega328 mikroupravljačem. Nadalje je objašnjen princip rada GSM/GPRS tehnologije koja se koristi za odašiljanje mjerenih podataka i SMS poruka. Nakon objašnjenog rada ove tehnologije je tehnički opisan SIM800L GSM/GPRS modul i AT kontrolne naredbe. Zatim slijede opisi ostalih korištenih komponenti (senzor mase, HX711, DHT11, LCD, PCF85063A, litij ionska baterija). Osim elektroničkih komponenti opisane su MySQL i PHP tehnologije te *phpmyadmin* alat za upravljanje nad bazom podataka. Četvrto poglavlje opisuje način korištenja spomenutih elektroničkih komponenti i povezivanje s Arduino pločicom, te su prikazani Arduino programski primjeri za implementaciju istih. Osim toga je prikazano PHP rješenje za povezivanje i unos informacija u bazu podataka. Peto poglavlje detaljno opisuje Arduino programsko rješenje projekta. U šestom poglavlju se opisuje na koji način je izvršeno ispitivanje mogućnosti određenih modula.

## **1.1. Zadatak diplomskog rada**

Zadatak diplomskog rada je kreirati *Arduino* platformu za praćenje mase košnice koja u različitim vremenskim intervalima korisniku šalje SMS poruku o trenutnom stanju košnice. Osim SMS obavijesti, dnevni podaci se putem GPRS (*engl. General Packet Radio Service*) komunikacijske usluge pohranjuju u bazu podataka nakon čega se isti prikazuju na web stranici gdje korisnik ima uvid o svim prethodnim podacima. Također je ugrađen senzor vlage i temperature na osnovu kojih korisnik ima uvid o vremenskim uvjetima koji se nalaze oko košnice te na osnovu tih parametara je moguće predvidjeti jesu li lokalni vremenski uvjeti bili zadovoljeni za rad pčela.

## 2. PREGLED PODRUČJA TEME

U današnje vrijeme na tržištu postoje razne verzije pčelarskih vaga te je zbog napretka tehnologije danas gotovo neizostavni dio pčelarske opreme. Tijekom zadnjih godina kada se mogao uočiti umanjeni doprinos meda na državnoj razini, upravo je nabava ovakvog projekta pomogla malim pčelarima da održe balans s korporacijskim konkurentima.

Iako se proizvodnja i razvoj mjernih uređaja za košnice izvodi već 20 godina, tek se u proteklih 10 godina započelo s ozbiljnim razvojem pčelarskih vaga.

Osnovne značajke kao što je mjerenje mase, temperature i vlage te slanje istih vrijednosti putem SMS poruke ili u bazu podataka dijele sve verzije pčelarskih vaga, no tijekom zadnjih 10 godina, svaki proizvođač je uvidio nove funkcionalnosti koje je moguće nadograditi i tako poboljšati proizvodnju meda. Dodavanje novih komponenti kao što su senzor za kišu, vjetar ili senzor za temperaturu legla su omogućile još bolji daljinski uvid u trenutno stanje okoline u kojoj se košnica nalazi. Magnetni kontakt za alarm, GPS i akcelerometar su omogućili zaštitu košnica od krađe.

Osim hardverskih proširenja, neki proizvođači su klijentima omogućili telekomunikacijske usluge izvan države. Tako korisnik bez problema može putovati Europom i dalje dobivati SMS uvid košnice po istoj cijeni preplate. Upravo mnogi pružaju godišnju preplatu svojim klijentima i tako se pčelari rješavaju briga kao što je trenutno stanje računa na SIM kartici ili vrijeme isteka preplate.

Proširenje opcija i sigurnost proizvoda nije jedina bitna stavka profesionalnih pčelarskih vaga. Proizvođači pružaju višegodišnje garancije, zamjenu dijelova i povjerljivu tehničku podršku. Trud i rad je mnogim proizvođačima donio razne certifikate i nagrade u državi i u Europi što je sigurno privuklo nove klijente i utjecalo na pozitivan marketing. Visoka pouzdanost profesionalnih pčelarskih vaga je utjecalo na cijenu te se tako na tržištu mogu pronaći pčelarske vage od minimalno 3000 kuna, a s opcijom nadogradnje, cijena se diže do 7000 kuna.

Za razliku od postojećih rješenja, ova verzija pametne košnice ne nudi razne dodatne mogućnosti u pogledu dodatnih informacija, sigurnosti i mobilnosti. Usprkos tome, osiguravaju se minimalne značajke koje sve digitalne pčelarske vage imaju zajedničko, a to je mjerenje mase, temperature i vlage te slanje tih informacije putem SMS poruke i/ili u bazu podataka.



## 2.1. Apiloger

*Apiloger* [1] pčelarska vaga prije svega privlači dizajnom i definitivno je lakša i praktičnija za rukovanje. Pruža osnovne tri usluge: mjerenje mase, temperature i vlage. Mjereni podaci se šalju putem SMS poruke.



**Sl. 2.1.** *Apiloger* pčelarska vaga<sup>1</sup>

## 2.2. SMSVAGA

*SMSVAGA* [2] pčelarska vaga nudi više opcija kao što su izvještaj mjerenih vrijednosti putem bluetooth tehnologije, SMS porukom ili besplatnim pozivom. Moguća je nadoplatiti osnovni model i tako nadograditi pčelarsku vagu sa sustavom zaštite od krađe, preopterećenja ili iznenadni gubitak težine.

---

<sup>1</sup> <https://www.pcelarska-oprema.hr/proizvodi/pcelarska-digitalna-vaga/>



Sl. 2.2. SMSVAGA<sup>2</sup>

### 2.3. XLOG bee

*XLOG bee* [3] pčelarska vaga nudi kvalitetnu inox izradu, zaštitu kablova od glodavaca, zaštitu od krađe, mjerenje tlaka zraka, količine padalina te jačine vjetra. Također jamče svojim kupcima trajanje baterije do 10 godina. Moguća je nadoplata osnovnog modela te se tako proširuju funkcionalnosti ove pčelarske vage.



Sl. 2.3 XLOG bee<sup>3</sup>

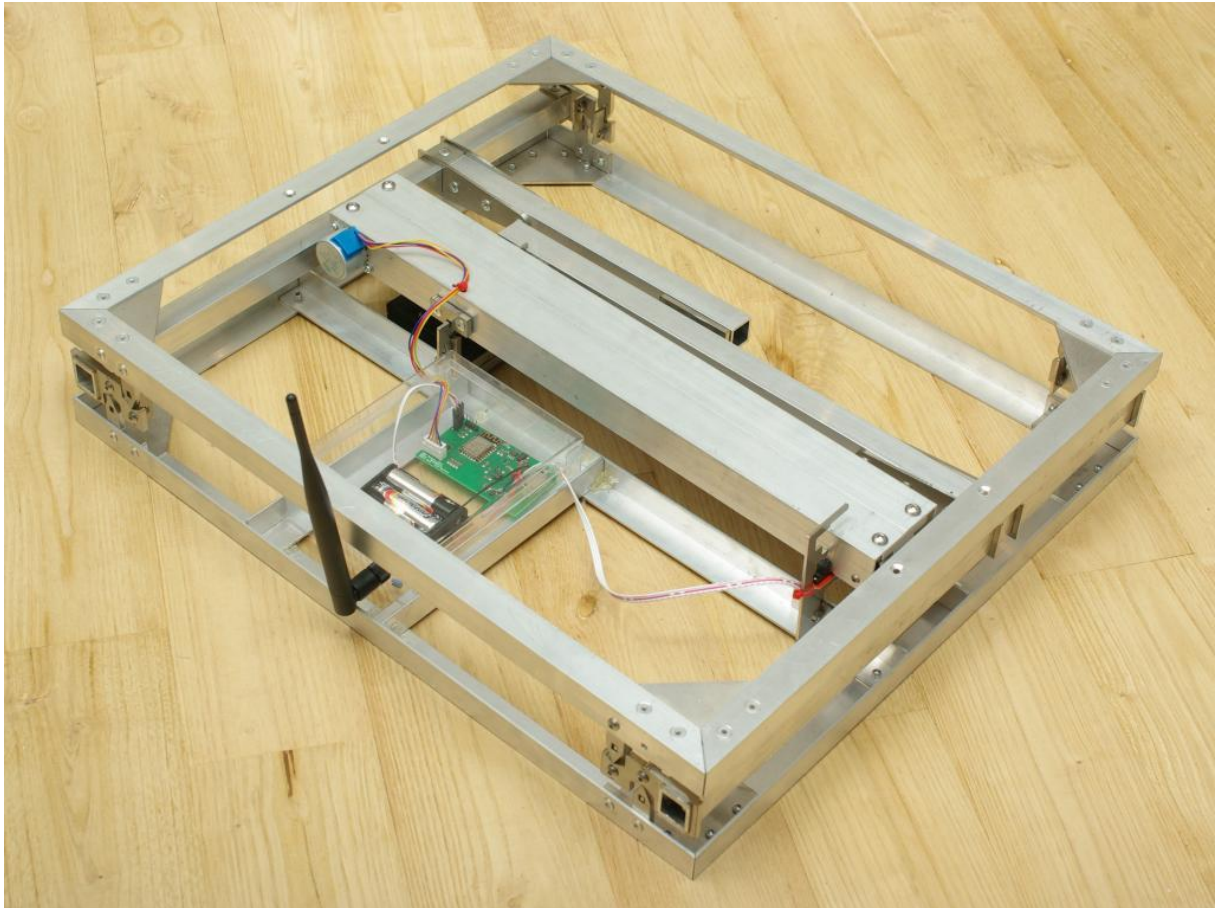
---

<sup>2</sup> <http://smsvaga.com/index.php/sr/prodavnica/sms-vaga-pocinkovana>

<sup>3</sup> <http://smsvaga.com/index.php/sr/prodavnica/sms-vaga-pocinkovana>

## 2.4. OpenHiveScale

*OpenHiveScale* [4] se razlikuje od ostalih proizvođača u smislu što oni kupcima osiguravaju sve dijelove i upute za samostalnu izradu DIY (*engl. Do it yourself*) pčelarske vage. Osim toga se klijentima nudi usluga godišnja preplata za usluge komunikacija. Osim standardnih GSM komunikacijskih opcija, nudi se povezivanje putem WiFi, Sigfox ili Lora mreže.



Sl. 2.4. *Open hive scale*<sup>4</sup>

---

<sup>4</sup> <http://www.openhivescale.org/>

## 2.5. Bee hive monitoring

*Bee hive monitoring* [5] omogućuje klijentima povezivanje više pčelarskih vaga u mrežu te putem mobilne aplikacije pružaju uvid u stanje svake košnice ili uvid kao u grupu košnica. Osim osnovnih funkcionalnosti, dodatno se mjeri aktivnost pčela, frekvencija i amplituda pčelinje kolonije te se tako može procijeniti zdravlje pčela. Sve informacije prikazane su kao linijski grafovi, a uvođenjem raznih filtera se može specificirati točan trenutak ili podatak koji se traži.



Sl. 2.5. *Bee hive monitoring*<sup>5</sup>

---

<sup>5</sup> <https://beehivemonitoring.com/en/home/78-scientific-hive-scale-3s-8588007887029.html>

### 3. TEHNIČKI PREDUVIJETI I KORIŠTENE TEHNOLOGIJE

Glavni dio sustava je Arduino UNO R3 platforma s ATmega328 mikroupravljačem koji svojim dizajnom pruža realiziranje problema. Korištene ulazne periferne komponente su senzori mase, temperature, vlage i sat stvarnog vremena. Arduino mikroupravljač omogućuje obradu i manipulaciju nad ulaznim podacima te ih takve može proslijediti izlaznim perifernim komponentama. U ovom slučaju su to LCD 16x2 ekran za ispis informacija i SIM800L GSM/GPRS modul za odašiljanje SMS poruka i pohranu informacija u bazu podataka. Za napajanje Arduino UNO R3 mikroupravljača se koristi 9V baterija, dok se za napajanje SIM800L modula koristi punjiva 2100mAh li-ionska baterija. Za pohranu informacija je korištena MySQL baza podataka te se s administracijskim alatom *phpmyadmin* i PHP skriptnim jezikom na taj način omogućuje pohrana i prikaz podataka unutar HTML-a.

#### 3.1. Arduino platforma

*Arduino* je platforma otvorenog koda čiji je glavni cilj kreiranje projekata koji su zasnovani na programabilnim elektroničkim komponentama. Zbog mogućnosti definiranja i čitanja ulaza i izlaza, omogućeno je na osnovu određenog događaja pokrenuti drugi kao što je na primjer paljenje LED diode, ako temperatura prostorije pređe određenu vrijednost. Spomenuti događaj će se moći izvršiti zbog toga što se ATmega328 mikro upravljaču na pločici šalje niz instrukcija. Arduino je do sada iskorišten za mnoge razne IoT (*engl. Internet of Things*) projekte i aplikacije zahvaljujući sljedećim prednostima [6]:

- **Financijski dostupan** – relativno jeftin naspram ostalih mikroupravljačkih platformi te je za početni modul potrebno isplatiti oko 300 kn
- **Raznorodnost** – Arduino IDE (*engl. Integrated Development Environment*) okruženje koje je zasnovano na C/C++ programskom jeziku je dostupno na Windows, Mac i Linux operacijskim sustavima. Većina ostalih mikroupravljačkih modula je ograničeno na Windows operacijski sustav
- **Nadogradivi otvoreni kod** – uzevši u obzir da su s izlaskom na tržište objavljeni alati otvorenog koda, omogućeno je proširenje stvaranjem C++ biblioteka. Tako danas postoje mnoge verzije biblioteka za razno sklopovlje
- **Nadogradivo sklopovlje** – osim otvorenog koda za stvaranje programa, Arduino pločica je s izlaskom na tržište dozvolila mnogim dizajnerima da kreiraju svoju verziju mikroupravljačke pločice

Arduino pločice se zasnivaju na Atmelovim 8-bitnim AVR mikroupravljačima kao što je ATmega328 te ovisno o verziji pločice se razlikuju i ostale značajke kao što je količina flash memorije ili broj ulazno/izlaznih pinova. Detaljnije o ATmega328 mikroupravljaču je opisano u potpoglavlju 3.1.1.

U svrhe diplomskog rada je korištena Arduino UNO R3 pločica koja je prikazana na slici 3.1



**Sl. 3.1.** Arduino Uno R3 mikroupravljačka pločica<sup>6</sup>

### 3.1.1. Mikroupravljač ATmega328

ATmega328 je mikroupravljač zasnovan na 8-bitnoj RISC (*engl. Reduced Instruction Set Computer*) AVR jezgri. Detaljne karakteristike [7] ATmega328 mikroupravljaču su prikazane u tablici 3.1.

**Tab. 3.1.** Osnovne karakteristike mikroupravljača ATmega328

Flash memorija	20 MIPS na 20 MHz
SRAM	2 KB
EEPROM	1 KB
Maksimalna radna frekvencija	20 MHz
Maksimalni broj ulazno/izlaznih jedinica	23
Broj instrukcija	131
Broj registara opće namjene	32
Broj pulсно širinskih moduliranih kanala	6
Napajanje	1.1V – 5.5V
A/D pretvornik	6 kanala po 10 bita

<sup>6</sup> <https://xplorcircuits.com/product/arduino-uno/>

Osim navedenih karakteristika, mikroupravljač ima tri brojača:

- Jedan 16 bitni timer/counter
- Dva 8 bitna timer/counter
- Jedan brojač s nezavisnim signalnom takta

Tri modula za komunikaciju:

- Programabilni serijski USART
- Master/Slave SPI komunikacija
- Two wire serijska komunikacija

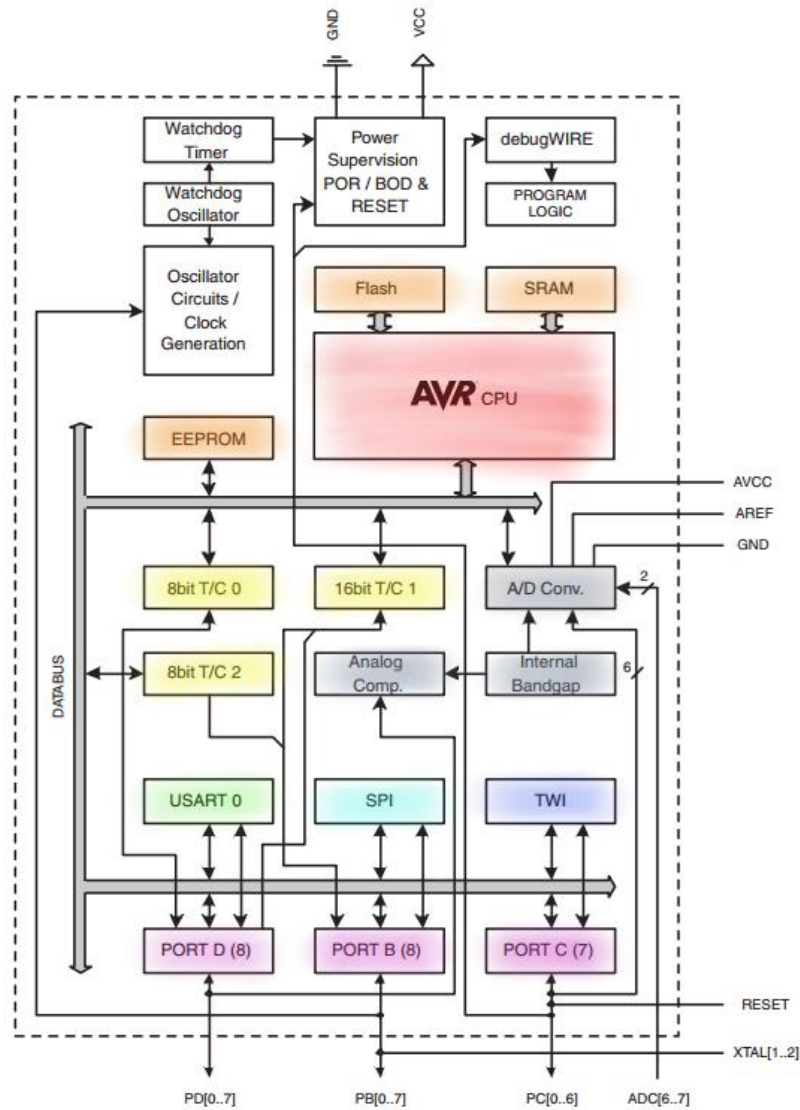
Za različite naredbe se koriste navedene tri memorije:

- **SRAM memorija** (*engl. Static Random Access Memory*) – služi kao radna memorija i tijekom izvođenja programa se u nju spremaju podaci
- **EEPROM memorija** (*engl. Electrically Erasable Programmable Read-Only Memory*) – služi za pohranu nepromjenjivih podataka kao što su različite konstante ili konfiguracijski parametri. Moguće je izvršiti najmanje 100 000 ciklusa pisanja i brisanja.
- **Flash memorija** – služi za pohranu programskih instrukcija te može izdržati najmanje 10 000 ciklusa pisanja i brisanja. Osim toga, sadržaj memorije se ne briše nakon što mikroupravljač ostane bez napajanja

Kao što je objašnjeno u [8] AVR jezgra koristi niz od 32 registra opće namjene koji su direktno povezani na aritmetičko logičku jedinicu (ALU) što omogućuje da se u jednoj instrukciji koja se izvodi u jednom ciklusu pristupi dva neovisnih registara. Upravo ovakva arhitektura rezultira efikasnijim programskim kodom dok se omogućuje do deset puta brži protok nego kod običnih CISC (*engl. Complex Instruction Set Computer*) mikroupravljača.

Mikroupravljač također sadrži generator takta koji osim definiranja brzine izvršavanja programskog koda određuje i rad ostalih periferija kao što su spomenuti brojači ili komunikacijski moduli (SPI, UART). Osim ugrađenog generatora takta za izvor je moguće iskoristiti još dva rješenja, a to su vanjski kristal i generirani takt signala koji se na mikroupravljač dovodi preko odgovarajućeg pina. Također je izvedeno rješenje za prekide (*engl. interrupt*) koji mogu biti unutrašnji i vanjski. Spomenuti brojači (*engl. timers*) služe za mjerenje vremena, brojanje te generiranje posebnih signala te se za slučaj nepredviđenih okolnosti (program zapeo u beskonačnoj petlji) koristi jedan od tih brojača pod nazivom

*watchdog*. Njegova uloga je automatsko resetiranje mikroupravljača. Sve navedene karakteristike kao blokovski prikaz se mogu vidjeti na slici 3.2.



Sl. 3.2. Blokovski prikaz ATmega328 mikroupravljača<sup>7</sup>

<sup>7</sup> <https://learn.adafruit.com/mcus-how-do-they-work?view=all>



### 3.1.2. Arduino UNO R3

Arduino UNO R3 je najčešća verzija Arduino pločice te je upravo ona korištena za izradu ovog projekta. Kao i ostale Arduino verzije, bazira se na ATmega328 mikroupravljaču. Popis karakteristika [9] je prikazan u tablici 3.2.

**Tab. 3.2.** Karakteristike Arduino UNO R3 pločice

Mikroupravljač	ATmega328
Radni napon	5V
Ulazni napon (preporučeni)	7-12V
Ulazni napon (granični)	6-20V
Broj digitalnih ulazno/izlaznih pinova	14 (6 nudi PWM izlaz)
Analogni ulazni pinovi	6
Struja na ulazno/izlaznom pinu	40 mA
Struja na 3.3V pinu	50 mA
Flash memorija	32 KB
SRAM	2 KB
EEPROM	1 KB
Radni takt	16 MHz

Arduino UNO R3 pločica se sastoji od sedam glavnih komponenti te prema [10] njihove uloge su sljedeće:

- **LED** – ugrađena LED dioda koja je povezana na digitalni pin 13. Za vrijednost '1' LED dioda je upaljena. Za vrijednost '0' LED dioda je ugašena. Vrijednosti '1' i '0' se odnosi na prisustvo napona +5V ili 0V na digitalnom pinu 13.
- **VIN** – ulazni napon za pločicu koji se dovodi vanjskim izvorom napajanja. Pin se može koristiti za napajanje ili se može pristupiti izvoru napajanja kada se koristi USB 5V napajanje ili neki drugi 5V pretvarač.
- **5V** – pin isporučuje napon 5V koji je dostupan putem DC napajanja (7-20V), USB konektorom (5V) ili VIN pinom na pločici (7-20V)
- **3.3V** – pin isporučuje napon 3.3V
- **GND** – pinovi za uzemljenje
- **RESET** – pin se koristi za mogućnost dodavanja gumba koji će resetirati pločicu

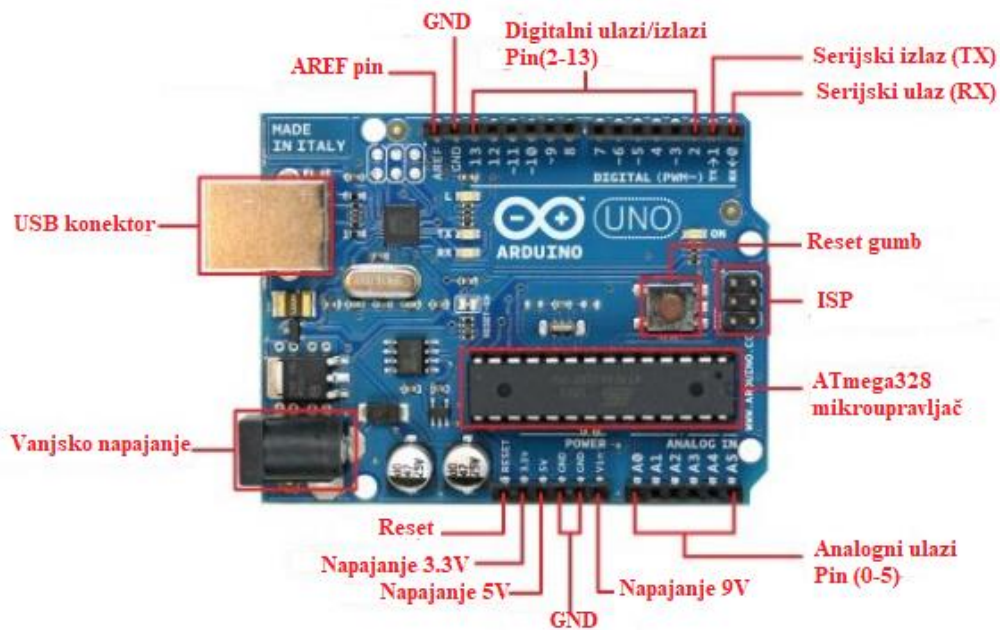
- **IOREF** – pin pruža referentnu vrijednost napona s kojom mikroupravljač radi. S pomoću ovog pina je moguće odabrati odgovarajući izvor napajanja ili omogućiti pločici rad korištenjem 5V ili 3.3V.

Osim glavnih komponenti Arduino UNO R3 pločica sadrži 14 digitalnih i 6 analognih pinova koji se mogu koristiti kao izlaz ili ulaz. Svi rade na 5V i mogu isporučiti ili primiti preporučenih 20mA struje te pinovi imaju ugrađeni *pull-up* otpornik od 20 do 50 kΩ. Iako je preporučena struja 20mA, do 40mA pinovi će relativno izdržati rad, ali se ne preporučuje takav rad jer je vrlo vjerojatna trajna šteta na mikroupravljaču.

Neki od navedenih pinova također imaju dodatnu funkciju:

- **SERIAL/UART** – digitalni pin D0 (RX) i digitalni pin D1 (TX) se koriste za primanje (RX) i odašiljanje (TX) TTL serijskih podataka
- **Vanjski prekidači** – digitalni pinovi D2 i D3 se mogu iskoristiti kao okidači (*engl. trigger*) za prekide (*engl. interrupt*)
- **PWM (*engl. Pulse Width Modulation*)** – digitalni pinovi D3, D5, D6, D9, D10 i D11 mogu isporučiti 8-bitni pulsno širinski modulirani signal korištenjem *analogWrite()* funkcije
- **SPI (*engl. Serial Peripheral Interface*)** – digitalni pinovi D10 (SS), D11 (MOSI), D12 (MISO) i D13 (SCK) podržavaju SPI komunikaciju korištenjem SPI biblioteke
- **TWI (*engl. Two Wire Interface*)** – analogni pinovi A4 (SDA) i A5 (SCL) podržavaju TWI komunikaciju korištenjem *Wire* biblioteke
- **AREF (*engl. Analog Reference*)** – referentni napon za analogne ulaze

Popis svih navedenih komponenti Arduino UNO R3 pločice se može vidjeti na slici 3.3



Sl. 3.3. Dijelovi Arduino UNO R3 pločice

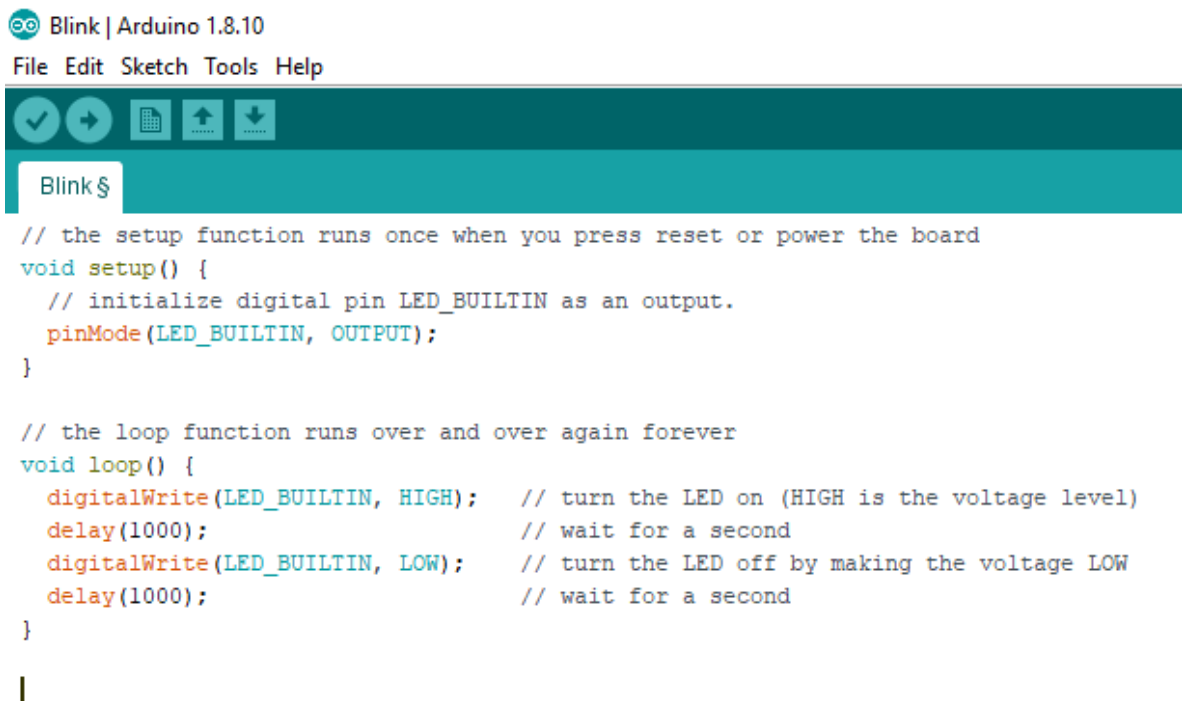
### 3.1.3. Arduino IDE

Arduino IDE (*engl. Integrated Development Environment*) je razvojno okruženje za pisanje programa za Arduino pločice te se zasniva na C/C++ programskom jeziku. Aplikaciju je moguće koristiti na Windows, Mac i Linux operacijskim sustavima te se koristi za pisanje i prebacivanje programa putem USB kabla na Arduino kompatibilnu pločicu, odnosno istovremeno omogućuje kompajliranje programa i izravno zapisivanje strojnog koda u programsku memoriju ATmega328 mikroupravljača. Također je omogućeno dodavanje vanjskih biblioteka od drugih korisnika. Bitno je napomenuti da svaki programski kod mora sadržavati dvije glavne funkcije kao što je prikazano na slici 3.4. Funkcija *setup()* se poziva svaki put prilikom inicijalizacije te funkcija *loop()* koja na neki način predstavlja beskonačnu petlju, odnosno unutar ove funkcije je izvršava glavni dio svakog programa. Osim dvije glavne funkcije korisnik može kreirati svoje vlastite funkcije koje kasnije može pozivati kada god je to potrebno.

Također je potrebno objasniti na koji način se definiraju vrijednost ulazno/izlaznih pinova. [11] Svaki pin može imati dvije naponske razine (5V ili 0V) koje odgovaraju logičkoj '1' ili logičkoj '0'. Osim naponskih razina, svakom pinu ATmega328 mikroupravljača su pridružena 3 8-bitna registra:

- DDR – registar smjera
- PORT – pin izlazni registar
- PIN – pin ulazni registar

Na osnovu ta tri registra se definira stanje (ulaz/izlaz) svakog pina. Naime, u DDR registru se za svaki pin definira njegova uloga, odnosno koristi li se on kao ulaz ili kao izlaz. Ovisno o stanju DDR registra se aktiviraju veze na PORT ili PIN registru što znači, ako se neki pin definira kao izlaz, tada se naponska razina na tom pinu definira pisanjem logičke '0' ili '1' u registru PORT. Isto vrijedi ako se neki pin definira kao ulaz, samo će se tada u registru PIN zapisati logička vrijednost '0' ili '1'. Ova radnja ima gotovu funkciju unutar Arduino IDE razvojnog okruženja. Funkcija *pinMode()* kojom se definira stanje pina (aktiviranje DDR registra) i dvije funkcije za pisanje *digitalWrite()* i čitanje *digitalRead()* stanja pina ATmega328 mikroupravljača. Primjer korištenja navedenih funkcija je prikazan na slici 3.4.



```
Blink | Arduino 1.8.10
File Edit Sketch Tools Help

Blink $
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
|
```

Sl. 3.4. Arduino IDE i primjer programa 'Blink'

### 3.2 GSM/GPRS tehnologija

GPRS (*engl. General Packet Radio Service*) je paketno orijentirana bežična usluga [13] osnovana 2000. godine na Europskom telekomunikacijskom institutu za standarde (ETSI). GPRS je usluga na GSM (*engl. Global System for Mobile Communications*) mreži, odnosno globalnom sustavu mobilne komunikacije za drugu (2G) i treću (3G) generaciju mobilnih komunikacijskih mreža. GPRS usluga podrazumijeva promjenjivu propusnost i kašnjenje koje ovisi o količini drugih korisnika koji u tom trenutku zajednički dijele uslugu. Pošto je ova vrsta mreže ograničena brojem korisnika, upotreba iznad ograničenja podataka se može u nekoj mjeri riješiti na tri načina kao što su naplata po količini skinutih podataka, ograničenje brzine skidanja ili trenutačno onemogućavanje pristupa mreži. 2G mreža zajedno s GPRS-om se još naziva 2.5G te se time misli na tehnologiju između druge (2G) i treće (3G) generacije mobilnih mreža. Brzina prijenosa podataka u 2.5G mreži je promjenjiva u rasponu od 56 do 144 kbit po sekundi.

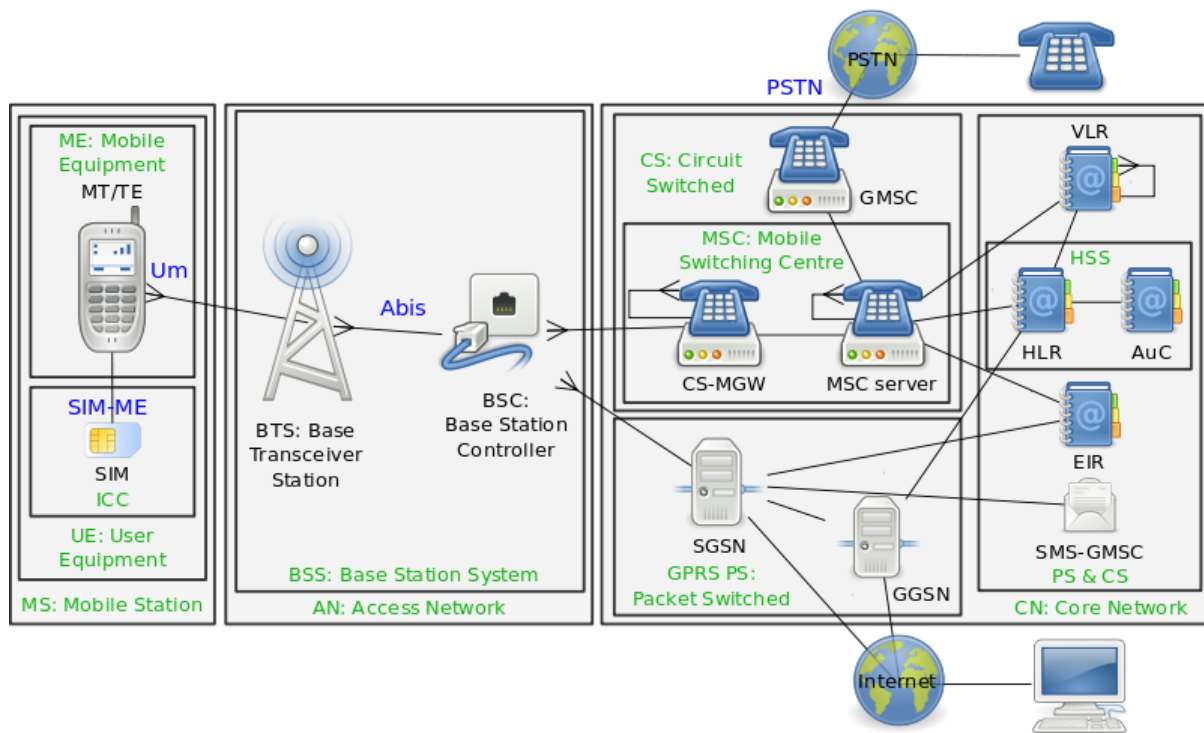
GPRS proširuje GSM mrežu i time stvara sljedeće mogućnosti:

- **Slanje i primanje SMS/MMS poruka** – brzina slanja SMS poruke GPRS-om je oko 30 SMS poruka po minuti što je puno brže naspram slanja SMS poruke korištenjem GSM mreže čija je brzina slanja oko 6-10 SMS poruka po minuti
- **Pristup Internet mreži** – uklanja proces *dial-up* i tako omogućuje pristup internetu jednim klikom
- **PoC (*engl. Push to Talk over Cellular*)** – usluga koja korisniku mobilnog uređaja omogućuje izravnu konekciju prema drugom korisniku korištenjem mobilne mreže
- **Wireless Village** – niz specifikacija za mobilno 'instant dopisivanje' (*engl. instant messaging*) i prikaz dostupnost korisnika (*engl. presence services*). Danas se mobilni uređaji dizajniraju s IMPS uslugom
- **Internet aplikacije za pametne uređaje** – korištenjem WAP (*engl. Wireless Application Protocol*) protokola se omogućuje korištenje WAP preglednika za pristup Internet mreži
- **P2P (*engl. Point to Point*)** – usluga koja omogućuje konekciju između dvije krajnje komunikacijske točke ili čvora kao što je telefonska linija
- **P2MP (*engl. Point to Multipoint*)** – usluga koja omogućuje konekciju 'jedan prema više'

Kako bi se GPRS konekcija uspostavila potrebno je definirati referencu na odgovarajući APN (*engl. Access Point Name*) koji nakon uspostavljanja konekcije definira traženu uslugu bilo da

se radi o SMS poruci ili Internet komunikaciji. Ovisno čija se usluga mobilnog operatera koristi, potrebno je definirati njihov APN. U ovom projektu se koriste T-Mobile usluge te je njihov APN `internet.ht.hr`

GSM mreža [11] se može podijeliti na tri glavna dijela (mobilna stanica, podsustav bazne stanice i mrežni podsustav) koji kao cjelina omogućuju komunikaciju kao što je prikazano na slici 3.5.



Sl. 3.5. Struktura GSM/GPRS mreže<sup>8</sup>

Mobilna stanica (*engl. Mobile Station*) se sastoji od samog GSM uređaja kao što je mobilni telefon ili u ovom slučaju SIM80L modul i SIM (*engl. Subscriber Identity Module*) kartice s pomoću koje se osigurava mogućnost preplate na mobilnu mrežu. Na SIM kartici je pohranjen IMSI (*engl. International Mobile Subscriber Identity*) unikatni broj što čini svaku SIM karticu posebnom i služi za identifikaciju na lokalnoj mobilnoj mreži.

Sustav mobilne stanice se zatim bežično povezuje i razmjenjuje podatke s najbližom baznom stanicom BTS (*engl. Base Transceiver Station*) koristeći Um 'zračno' sučelje koje zapravo opisuje komunikacijsku vezu između dvije stanice u mobilnim mrežama. Zatim se putem Abis sučelja bazna stanica povezuje s upravljačkim djelom bazne stanice BSC (*engl. Base Station Controller*). BSC predstavlja mrežni element koji upravlja i nadgleda broj baznih stanica te nudi sučelje prema mrežnom podsustavu gdje je glavna komponenta MSC (*engl. Mobile*

<sup>8</sup> <https://en.wikipedia.org/wiki/GSM>

*Switching Cener*). MSC se može opisati kao komunikacijski čvor koji vrši potrebne zahtjeve koristeći ostale elemente [14] mrežnog podsustava kao što su:

- **AuC** (*engl. Authentication Center*) – funkcija za autentifikaciju SIM kartica koje se povezuju na GSM mrežu. Ovaj postupak se uobičajeno obavlja onog trenutka kada je GSM uređaj upaljen.
  - **HLR** (*engl. Home Location Register*) – onog trenutka kada je autentifikacija uspješno obavljena HLR može upravljati sa SIM karticom. HLR zapravo predstavlja bazu podataka koja posjeduje podatke o GSM uređaju te se dodatno stvara enkripcijski ključ koji se naknadno koristi za kriptiranje bežične komunikacije između GSM uređaja i mrežnog podsustava
- **VLR** (*engl. Visitor Location Register*) – baza podataka koja pohranjuje podatke primljene od mobilne stanice ili HLR-a. Svaka bazna stanica BTS se služi sa samo jednim VLR-om što znači da u jednom trenutku unutar VLR-a mora biti točno jedan pretplatnik.
- **EIR** (*engl. Equipment Identity Register*) – predstavlja sigurnosnu uslugu koja je često integrirana s HLR-om. EIR sadrži popis svih GSM uređaja (identificirani po IMEI-u) koji su zabranjeni na mreži ili se prate. Ova usluga omogućuje praćenje ukradenih GSM uređaja.

Ovisno zahtjeva li korisnik uslugu spajanja na Internet ili primanje/slanje SMS poruka i poziva se koristi različita metoda. Metoda preklapanja krugova (*engl. Circuit switching*) omogućuje primanje/slanje poruka i poziva tako što se između dva mrežna čvora uspostavlja komunikacijski kanal (krug). Komunikacijski krug osigurava potpunu propusnost kanala te veza ostaje uspostavljena sve dok traje komunikacijska sesija.

Nadogradnjom GSM mreže s GPRS tehnologijom se uvodi druga metoda, preklapanje paketa (*engl. Packet switching*) koja grupira podatke koji se odašilju kroz mrežu u pakete koji se sastoje od zaglavlja (*engl. header*) i tijela (*engl. payload*). Zaduženo mrežno sklopovlje koristi podatke iz zaglavlja kako bi usmjerilo pakete prema odredištu gdje se podaci iz tijela izdvajaju od strane određene aplikacije.

### 3.3. SIM800L modul

SIM800L modul predstavlja GSM modem koji se može iskoristiti za razne IoT projekte jer omogućuje uspostavljanje i primanje poziva, poruka te povezivanje na Internet koristeći GPRS mrežu.



Sl. 3.6. SIM800L modul<sup>9</sup>

Modul se primarno sastoji od GSM čipa [15] čiji je radni napon između 3.4 - 4.4V te se zbog toga preporučuje koristiti posebno napajanje kao što je li-ion baterija koja je detaljnije opisana u potpoglavlju 3.8. Radni napon nije glavna stavka zbog koje se preporučuje korištenje vlastitog napajanja, a ne napajanje s Arduino pločice. Ovisno o stadiju rada SIM800L modula, maksimalna moguća struja iznosi oko 2A tijekom prijenosa podataka. U tablici 3.3 se nalazi popis potrošnje struje u različitim stadijima rada.

Tab. 3.3. Potrošnja u različitim stadijima rada

Stadij rada	Potrošnja struje
Ugašen	60 uA
<i>Sleep</i>	1 mA
U stanju pripravnosti ( <i>stand by</i> )	18 mA
Poziv	131 mA – 216 mA
GPRS	453 mA
<i>Transmission burst</i>	2 A

<sup>9</sup> <https://makepro.co/product/sim800l-gsm-gprs-module/>



SIM800L modul zahtjeva vanjsku antenu za spajanje na mrežu te dolazi sa U.FL konektorom na koji je moguće spojiti 3dBi antenu. Korištenjem ove antene se omogućuje spajanje na mrežu dok je SIM800L modul s ostalim sklopovljem zatvoren u kućištu sve dok je antena vani.



Sl. 3.7. SIM800L modul sa spojenom 3dBi antenom<sup>10</sup>

Modul se također sastoji od LED diode koja služi kao pokazatelj, tj. ovisno o dužini stanke između upaljene i ugašene (*blink*) se ukazuje na stanje modula:

- **Blink svake sekunde** – modul je upaljen ali nije uspostavio vezu prema mobilnoj mreži
- **Blink svake dvije sekunde** – aktivna GPRS veza
- **Blink svake tri sekunde** – modul je uspostavio vezu s mobilnom mrežom i spreman je za primanje i slanje poziva i SMS poruka

<sup>10</sup> <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>

Prema [16], SIM800L modul se sastoji od 12 pinova kao što je prikazano na slici 3.8, a njihova uloga je opisana u tablici 3.4.



Sl. 3.8. Raspored pinova na SIM800L modulu<sup>11</sup>

Tab. 3.4. Opis pinova SIM800L modula

NET	Pin za spajanje antene na modul
VCC	Napajanje modula između 3.4 – 4.4 V
RST	<i>Hard reset</i>
RXD	<i>Receiver</i> pin za serijsku komunikaciju
TXD	<i>Transmitter</i> pin za serijsku komunikaciju
GND	Uzemljenje
RING	<i>Ring</i> indikator
DTR	Pin za aktiviranje/deaktiviranje <i>sleep</i> moda HIGH = <i>sleep</i> , LOW = <i>wake up</i>
MIC±	Ulaz za mikrofon
SPK±	Izlaz za zvučnik

<sup>11</sup> <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>

### 3.3.1. AT naredbe

Za upravljanje uređaja koji koriste GSM mrežu potrebno je koristiti AT (*engl. Attention*) naredbe putem serijske komunikacije. Prema sintaksi [17], AT naredbe se mogu podijeliti u tri kategorije:

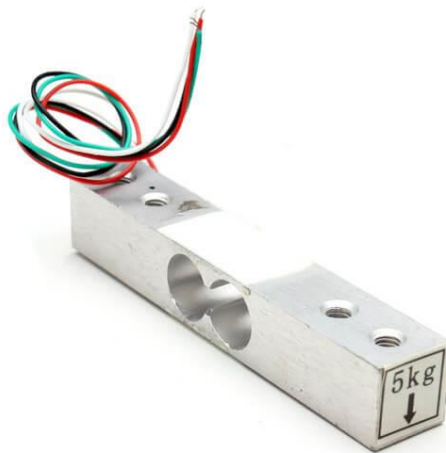
- **Osnovne** – AT naredbe koje imaju format **AT<x><n>**, gdje je '<x>' naredba a '<n>' argument
- **'S' parametar** – AT naredbe koje imaju format **ATS<n>=<m>**, gdje '<n>' označava indeks 'S' registra, a '<m>' predstavlja vrijednost koja joj se dodjeljuje.
- **Proširene** – AT naredbe koje mogu raditi u nekoliko režima rada

**Tab. 3.5.** Popis najčešće korištenih AT naredbi, značenje i rezultat

AT naredba	Značenje AT naredbe	Rezultat	Značenje rezultata
AT	Provjera je li uređaj spojen na mrežu	OK	Uređaj je spojen na mrežu
AT+CSQ	Kvaliteta signala. Vrijednosti između 0-31.	+CSQ: 24,0	Prvi broj označava kvalitetu u [dB]
AT+CCID	Vraća SIM broj i provjerava valjanost SIM kartice	"broj kartice" OK	Vraća broj kartice (19 znamenki)
AT+CREG?	Provjera je li GSM uređaj registriran u mreži.	+CREG: 0,1	Drugi broj 1 ili 5. 1 = 'kućna' mreža 5 = <i>roaming</i>
AT+CBS	Stanje li-ion baterije	+CBS: 0,93,3877	Drugi broj označava postotak napunjenosti. Treći broj predstavlja vrijednost u mV
AT+CMGF=1	Odabiranje SMS formata	OK	Odabran je SMS format
AT+CMGS=+385...	Slanje SMS poruke na odabrani broj	"SMS poslana poruka"	SMS poruka koja može biti definirani <i>string</i> ili mjerena vrijednost

### 3.4. Senzor mase

Senzor mase (*engl. load cell*) [18] spada u grupu uređaja koji se zovu transduktori čija je uloga pretvaranje signala iz jednog oblika energije u signal drugog oblika energije. U ovom slučaju pretvara sile kao što su napetost, pritisak ili naprezanje u električni signal koji se može izmjeriti i prilagođavati potrebama. Kako se sila primijenjena na senzoru mase povećava tako se proporcionalno mijenja i električni signal. Danas na tržištu postoje mnoge vrste ovisno o vrsti sile koja se primjenjuje, pa tako postoji hidraulički i pneumatički senzor mase, ali za potrebe ovog projekta je korišten senzor koji koristi tenzometar (*engl. strain gauge*).

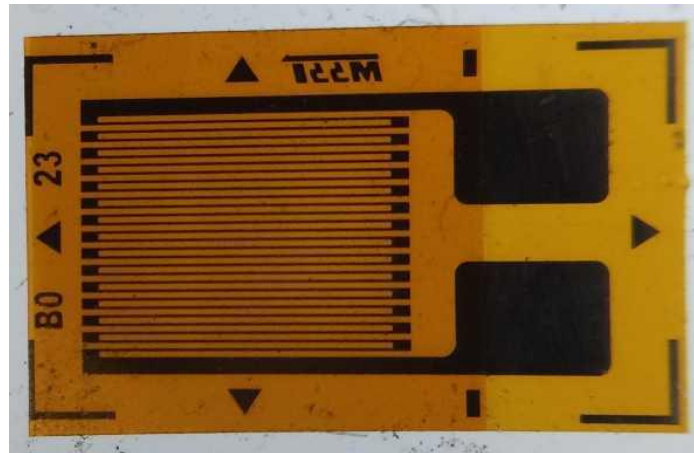


Sl. 3.9. Senzor mase<sup>12</sup>

Tenzometar [19] je uređaj koji mjeri naprezanje nekog objekta te se danas najčešće sastoji od izolacijske fleksibilne podloge koja podržava uzorak na metalnoj foliji kao što je prikazano na slici 3.10. Kako se sila primjenjuje na objekt tako se on deformira, što deformira i metalnu foliju, a rezultat je mijenjanje električnog otpora. Deformacije na metalnoj foliji mogu biti manje od jednog milimetra pa sve to nekoliko centimetara i to je dovoljno da se izmjeri minimalna i maksimalna vrijednost naprezanja.

---

<sup>12</sup> <https://www.jsumo.com/load-cell-bar-0-5kg>



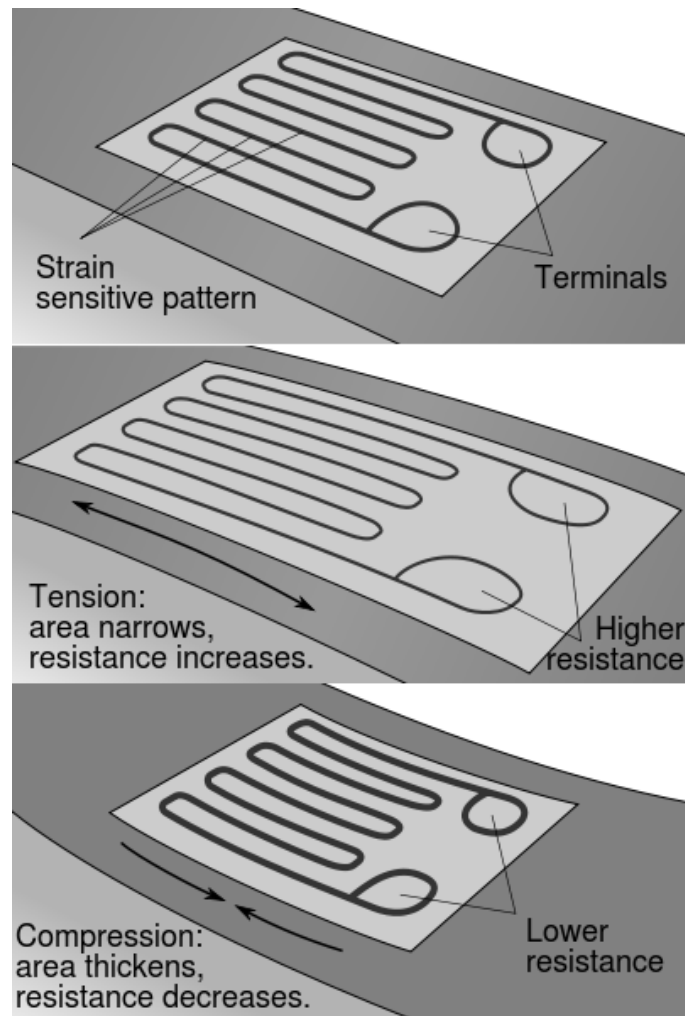
**Sl. 3.10.** Tenzometar u obliku metalne folije<sup>13</sup>

Tenzometar koristi dva osnovna principa, a to su električna provodljivost i njegova ovisnost o geometriji električnog vodiča. To znači da kada se električni vodiči koji se nalaze u metalnoj foliji, prošire do svoje granice rastezljivosti, oni postaju uži i duži što će na krajevima (terminalima) rezultirati s većim električnim otporom. Vrijedi i obrnuto kada se vodiči skupe, oni postaju kraći, što smanjuje električni otpor na krajevima kao što je prikazano na slici 3.11.

Uobičajeni tenzometri se sastoje [20] od dugih i tankih vodljivih linija postavljeni u *zig-zag* uzorak paralelnih linija. Postavljanje vodljivih linija u paralelni uzorak ne znači da će se povećati osjetljivost jer je omjer promjene u otporu za određeno naprezanje isti kao i da se postavi samo jedan uzorak. Paralelne linije se koriste samo zbog toga što bi jedan uzorak morao biti ekstremno tanak i pri tome bi bio podložan pregrijavanju što bi uzrokovalo promjenom otpora kako se vodljive linije šire, a samim time bi bilo teže izmjeriti točnu promjenu otpora te je moguće pucanje tenzometra.

---

<sup>13</sup> [https://en.wikipedia.org/wiki/Strain\\_gauge](https://en.wikipedia.org/wiki/Strain_gauge)

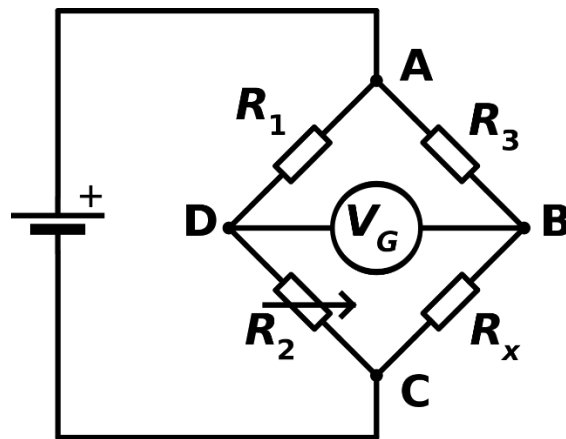


Sl. 3.11. Prikaz rada tenzometra prilikom deformacije<sup>14</sup>

S obzirom na to da je promjenjivi otpor koji se mjeri jako mali, teško je s visokom preciznošću izmjeriti veličine no to se može riješiti tako da se poveća broj primjenjivih tenzometra (otpornika) koji povećavaju male promjene u veličine koje je lakše izmjeriti. Niz od četiri otpornika postavljeni u formaciju se zove *Wheatstone bridge*.

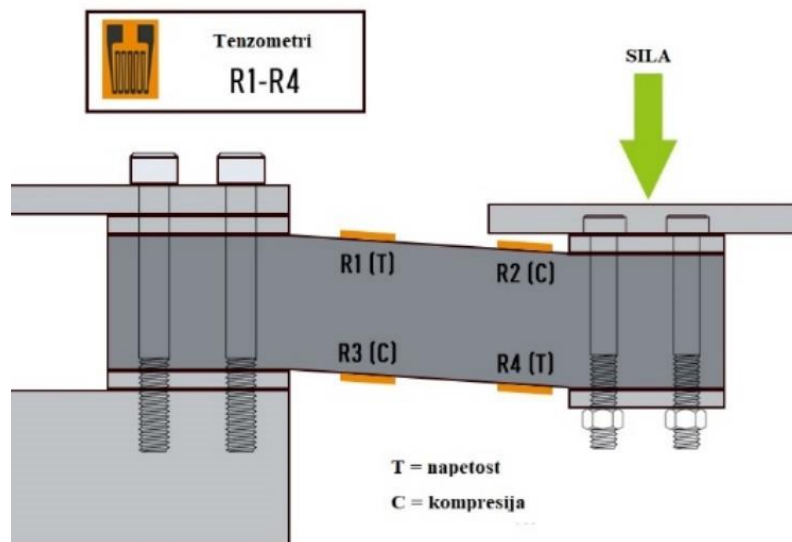
*Wheatstone bridge* [21] je konfiguracija koja se sastoji od četiri otpornika s poznatim pobuđenim naponom kao što je prikazano na slici 3.12. U slučaju kada se koristi senzor mase otpornike zamjenjuje tenzometar.

<sup>14</sup> [https://en.wikipedia.org/wiki/Strain\\_gauge](https://en.wikipedia.org/wiki/Strain_gauge)



Sl. 3.12. *Wheatstone bridge* konfiguracija<sup>15</sup>

Pobuđeni napon  $V_{EX}$  je poznata konstanta dok je napon izlaza  $V_O$  promjenjiv ovisno o napetosti što znači da ako su sva četiri otpornika u ravnoteži  $\frac{R_1}{R_2} = \frac{R_4}{R_3}$  izlazi napon  $V_O$  je nula. U slučaju da se otpor u samo jednom otporniku promijeni tada će rezultirati promjenom i na izlaznom naponu. Promjene u izlaznom naponu  $V_O$  se mogu jednostavno izmjeriti korištenjem Ohmovog zakona te kada se sva četiri otpornika u *Wheatstone bridge* krugu primjene, izlazni napon  $V_O$  je jednak jednadžbi:  $V_O = \left( \frac{R_3}{R_3+R_4} - \frac{R_2}{R_1+R_2} \right) * V_{EX}$



Sl. 3.13. Vizualizacija promjene otpora primjenom vanjske sile<sup>16</sup>

<sup>15</sup> [https://en.wikipedia.org/wiki/Wheatstone\\_bridge](https://en.wikipedia.org/wiki/Wheatstone_bridge)

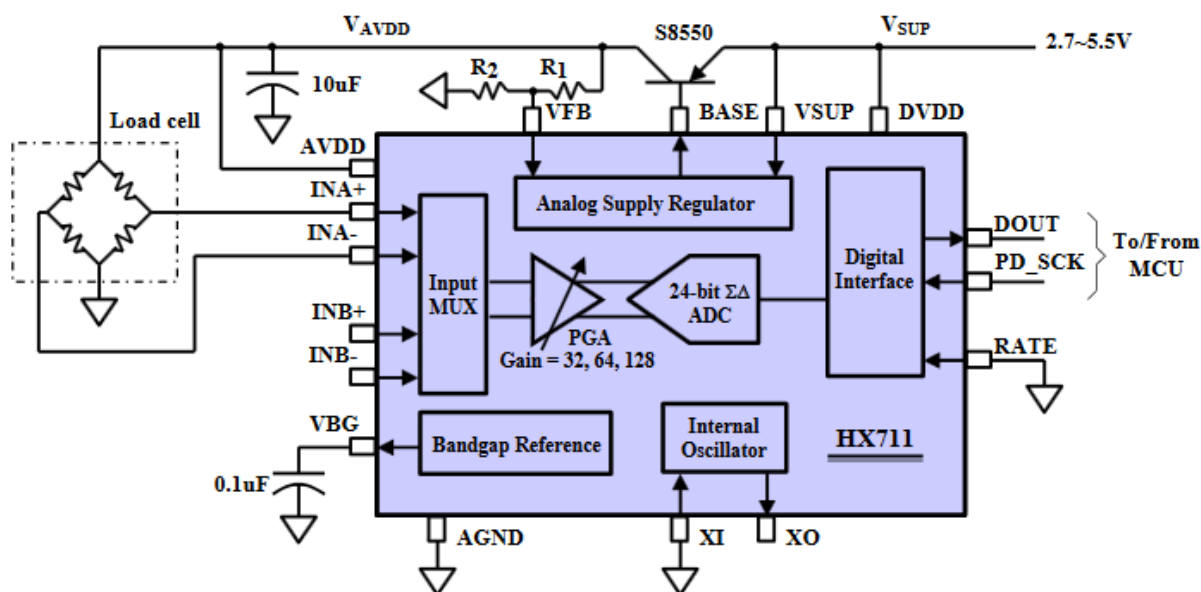
<sup>16</sup> [https://en.wikipedia.org/wiki/Load\\_cell](https://en.wikipedia.org/wiki/Load_cell)

Površina metalne folija ovisno o veličini uređaja nad kojim se vrši vanjska sila je između 2-13 mm<sup>2</sup> te može koristiti 2.7-5V kao ulazni napon što je savršeno za ovaj projekt jer se može iskoristiti 5V pin na Arduino pločici. Napon koji se očitava, odnosno izlazni napon je reda milivolti [mV].

### 3.5. HX711

Iako se uvođenjem *Wheatstone bridge*-a povećavaju promjene otpora one su i dalje male, pa su promjene izlaznog napona male što nije prilagodljivo za precizno mjerenje. Upravo se zbog toga koristi 24-bitni analogno digitalni pretvarač HX711 koji pojačava signal 128 puta. Spajanjem pojačala na mikroupravljač, odnosno Arduino pločicu, omogućuje jednostavnije čitanje promjena izlaznog napona sa senzora mase nakon potrebne kalibracije.

Kao što je prikazano na slici 3.14, HX711 pretvarač se sastoji od multipleksera na ulazu koji odabire kanal A ili B za rad. Kako je objašnjeno prema [22], u većini slučajeva se odabire rad na kanalu A jer je on namijenjen za pojačanje signala 128 ili 64 puta dok kanal B ima fiksno pojačanje od 32 puta.



Sl. 3.14. Blok dijagram HX711 pretvarača<sup>17</sup>

<sup>17</sup> <https://components101.com/ics/hx711-24-bit-analog-digital-converter-adc>

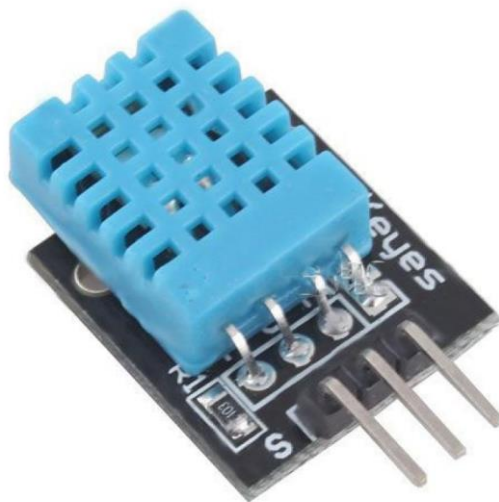


Senzor mase najčešće dolazi u izvedbi s četiri izlaza (žice), tj. ako je izrađena izvedba *Wheatstone bridge* tada će svakako postojati četiri žice za izlaz. U slučajevima kada senzor dolazi s tri žice, znači da je potrebno samostalno izvesti *Wheatstone bridge* uvođenjem tri dodatna senzora.

### 3.6. DHT11

DHT11 [23] senzor temperature i vlage je jednostavan senzor za mjerenje temperature i relativne vlage na lokalnom području. Senzor se sastoji od dva dijela: kapacitivni senzor vlage i termistor koji je zapravo otpornik koji mijenja otpor s temperaturom. U suštini, svi otpornici su termistori jer se otpor svakog otpornika mijenja s temperaturom, ali te promjene su iznimno male i u velikoj većini zanemarive. Termistori su s toga izgrađeni tako da se njihov otpor drastično mijenja s promjenom temperature pa je tako mogući omjer promjene oko  $100\Omega/^\circ\text{C}$ . Također se senzor sastoji od čipa koji obavlja analogno digitalnu pretvorbu te taj digitalni signal isporučuje kroz jedan od pinova koji je jednostavno očitati korištenjem mikroupravljača.

Na tržištu postoje dvije serije [24] DHT senzora, DHT11 i DHT22 čije su bitnije razlike samo u rasponu mjerenja temperature i relativne vlage. Tako je DHT11 pogodan za mjerenje relativne vlage u rasponu od 20-80% s preciznošću od 5% i za mjerenje temperature od  $0-50^\circ\text{C}$  s preciznošću  $\pm 2^\circ\text{C}$ . DHT22 ima mogućnost mjerenja relativne vlage od 0-100% s preciznošću 2-5% te omogućuje znatno veći raspon temperature od  $-40$  do  $80^\circ\text{C}$  s preciznošću od  $\pm 0.5^\circ\text{C}$ .



Sl. 3.15. DHT11 senzor temperature i vlage<sup>18</sup>

<sup>18</sup> <https://components101.com/dht11-temperature-sensor>

Mjerenje relativne vlage je jednako omjeru količine vodene pare u zraku i vode u zraku u točki zasićenja. U trenutku zasićenja, vodena para se počinje kondenzirati i akumulirati na površinama i tako stvara rosu. Točka zasićenja se mijenja s temperaturom zraka što znači da hladni zrak može sadržavati manje vodene pare prije nego dosegne točku zasićenja dok topli zrak može sadržavati više vodene pare prije nego dosegne točku zasićenja. Relativna vlaga se mjeri u postocima i može se izračunati korištenjem sljedeće formule:

$$RH = \frac{\rho_w}{\rho_s} * 100\% \quad (3-1)$$

gdje je RH relativna vlaga,  $\rho_w$  gustoća vodene pare,  $\rho_s$  gustoća vodene pare u točki zasićenja.

### 3.7. LCD I2C

LCD 16x2 (*engl. Liquid Crystal Display*) ekran omogućuje ispisivanje informacije u raznim oblicima do 16 znakova u 2 reda. LCD ekran dolazi s promjenjivim pozadinskim osvjetljenjem koje je moguće definirati korištenjem 10k $\Omega$  potenciometra. LCD ekran je temeljen na tehnologiji tekućih kristala te se sastoji od određenog broja piksela koji su u redu poredani, a iza njih se nalazi pozadinsko osvjetljenje koje je najčešće LED dioda. Radni napon je 5V što omogućuje izravno povezivanje na Arduino pločicu te je potrošnja ne računajući pozadinsko osvjetljenje 1mA.

Osim prikaza alfanumeričkih znakova, moguć je i prikaz svih ASCII znakova jer se svaki znak prikazuje unutar granica 5x8 piksela. Kao što je prikazano na slici 3.16, LCD ekran se sastoji od 16 pinova [25] čije je objašnjenje opisano u tablici 3.6.



Sl. 3.16. Raspored pinova na LCD ekranu<sup>19</sup>

<sup>19</sup> <https://components101.com/16x2-lcd-pinout-datasheet>

**Tab. 3.6.** Popis i objašnjenje pinova LCD ekrana

Broj pina	Naziv pina	Opis
1	Vss (GND)	Pin za uzemljenje
2	Vdd (+5V)	Napajanje LCD-a (4.7V – 5.3V)
3	VE (Contrast V)	Određuje razinu kontrasta
4	Register Select	Prebacivanje između registara command/dana na mikroupravljaču
5	Read/Write	Za čitanje/pisanje podataka
6	Enable	Za potvrdu podataka se izmjenjuje između 0 i 1. Spojen na mikroupravljač
7	Data Pin 0	8-bitna podatkovna linija
8	Data Pin 1	8-bitna podatkovna linija
9	Data Pin 2	8-bitna podatkovna linija
10	Data Pin 3	8-bitna podatkovna linija
11	Data Pin 4	8-bitna podatkovna linija
12	Data Pin 5	8-bitna podatkovna linija
13	Data Pin 6	8-bitna podatkovna linija
14	Dana Pin 7	8-bitna podatkovna linija
15	LED Positive	Anoda LED diode
16	LED Negative	Katoda LED diode

Iz slike i tablice se može zaključiti da je za uspješni rad LCD ekrana potrebno spojiti 8 ulazno/izlaznih jedinica na 8 digitalnih pinova mikroupravljača samo za prikaz podataka, ne računajući pinove za napajanje ili pozadinsko osvjetljenje. U situacijama gdje se koristi samo LCD bez ostalih sklopovlja, ovo ne predstavlja problem, ali u većini slučajeva broj digitalnih ulaza/izlaza biva vrlo brzo ograničen. Upravo se zbog toga na LCD ekran često zalemi I2C (*engl. Inter-Integrated Circuit*) pretvarač koji omogućuje broj pinova sa 16 na 4 (dva za komunikaciju i dva za napajanje) kao što je prikazano na slici 3.17.

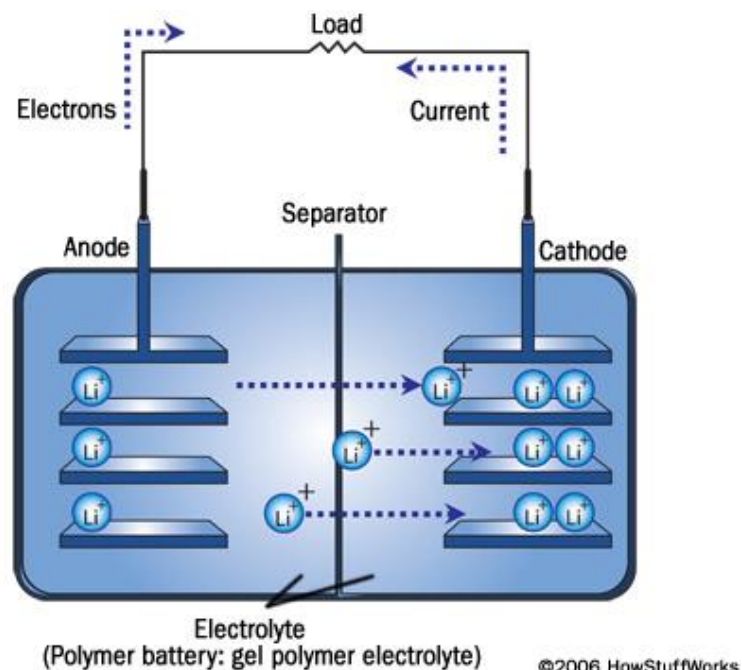


Sl. 3.17. I2C pretvarač<sup>20</sup>

Pinovi za komunikaciju GND i VCC se spajaju na GND i 5V pinove na Arduino UNO pločici dok se komunikacijski pinovi spajaju na analogne ulaze. Pin za slanje serijskih podataka (SDA) s I2C pretvarača se spaja na A4 analogni ulaz dok se serijski *clock* (SCL) spaja na A5 analogni ulaz.

### 3.8. Li-ion baterija

Li-ion odnosno litij-ion baterije [26] su punjive baterije koje koriste litij-ionsku tehnologiju u smislu da se prilikom pražnjenja, ioni litija pomiču od negativne elektrode do pozitivne elektrode kroz elektrolit te obrnuto prilikom punjenja. Elektrode i pregrada koja ih odvaja su uronjene u gel koji je zapravo elektrolit i on omogućuje kretanje iona. Proces punjenja i pražnjenja li-ion baterije je prikazan na slici 3.18.



Sl. 3.18. Proces punjenja i pražnjenja li-ion baterije<sup>21</sup>

<sup>20</sup> <https://www.indiamart.com/proddetail/i2c-lcd-adapter-module-12990521762.html>

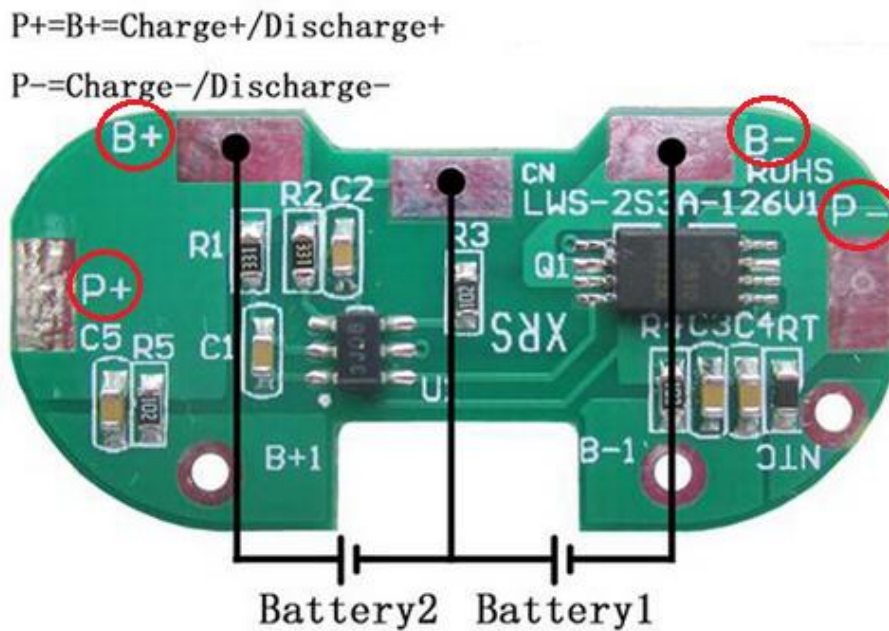
<sup>21</sup> <https://electronics.howstuffworks.com/everyday-tech/lithium-ion-battery.htm>

Zbog svojih malih dimenzija i mase, najčešće se koristi za napajanje mobilnih uređaja iako su dostupne različite verzije koje mogu napajati brojne elektroničke sklopove te su danas najbolje punjive baterije dostupne na tržištu. Osim toga, zahvaljujući visokim kapacitetima, danas se li-ion baterije također koriste za električna vozila te potrebe vojske i svemirskih projekata.

Za razliku od NiMH (*engl. Nickel-metal hydride*) punjivih baterija, li-ionske imaju značajne prednosti [27] kao što su:

- **Veći nazivni napon** – nazivni napon je tri puta veći (3.7V)
- **Veća gustoća energije** – skoro dvostruko veća gustoća
- **Minimalni memorijski učinak** – punjenje i pražnjenje je moguće u bilo kojem stanju napunjenosti bez smanjivanja ukupnog kapaciteta baterije što rezultira potrebom za održavanjem
- **Dugotrajnost** – 500 do 1000 potpunih ciklusa punjenja (od 0% do 100%)
- **Razne dimenzije** – dostupne izvedbe u raznim dimenzijama. Povećavanjem dimenzija se povećava i ukupni kapacitet baterije

Osim očiglednih prednosti, veliki nedostatak ovih baterija je pretjerano punjenje (*engl. overcharge*) i pražnjenje (*engl. overdischarge*). Ovisno koliko je baterija napunjena se mijenja napon od maksimalnih 4.2V kada je potpuno puna do minimalno 2.5V. Pretjerano pražnjenje baterije ugrožava ukupni kapacitet jer napon ne bi trebao biti manji od 3V. Kako bi se riješio taj problem, mnogi proizvođači kreiraju elektroničko rješenje u obliku PCM-a (*engl. Protection Circuit Management*) čija je uloga prekidanje strujnog kruga pri prevelikom odnosno premalom naponu na bateriji.



Sl. 3.19. PCM modul<sup>22</sup>

### 3.8.1. Punjenje li-ion baterije

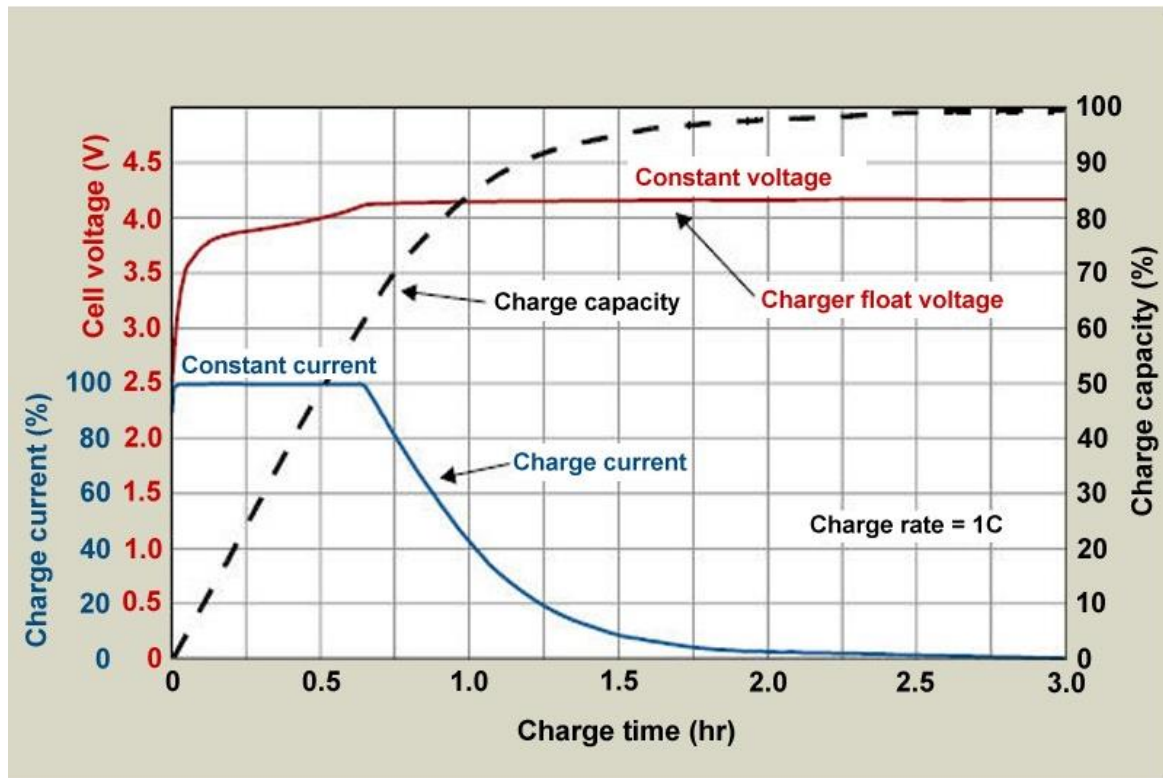
Zbog svojeg dizajna, li-ion baterije zahtijevaju posebne punjače koji će zadovoljiti različite režime rada same baterije te se samim time štiti ukupni kapacitet baterije. Postoje tri faze [28] punjenja, a to su predpunjenje, konstantna struja i konstantni napon.

- **Predpunjenje** (*engl. preconditioning*) – ova faza se događa samo kada je napon baterije ispod 3V, znači kada je baterija potpuno prazna. Pošto je baterija potpuno prazna to znači da ne može pohraniti energiju iako je spojena na punjač. Upravo zato je potrebno predpunjenje što znači da se baterija najprije mora dovesti do 3V. Predpunjenje se neće izvršiti, ako je napon baterije 3V, odnosno tada se ova faza preskače.
- **Konstantna struja** (*engl. constant current*) – u ovoj fazi se baterija puni konstantnom strujom u onom trenutku kada je napon baterije između 3V i 4.2V. Tvorničke preporuke su da bi brzina punjenja (*engl. charge rate*) trebala biti 0.5C (0.8C ili manje) gdje C predstavlja kapacitet baterije. Što znači da će se baterija od 1000mAh puniti dva sata, ako se koristi 0.5C (500mA/h). Također je omogućena opcija brzog punjenja (*engl. quick charge*) gdje se koristi 1.0C i 2.0C. Brzo punjenje će svakako skratiti trajanje ove

<sup>22</sup> [https://www.kisiifmm.com/index.php?main\\_page=product\\_info&products\\_id=217431](https://www.kisiifmm.com/index.php?main_page=product_info&products_id=217431)

faze, ali će isto tako produžiti trajanje sljedeće faze (faza konstantnog napona) što će na kraju rezultirati gotovo istim vremenom potpune napunjenosti kao da se koristi preporučeno 0.5C punjenje.

- **Konstantni napon** (*engl. constant voltage*) – posljednja faza punjenja se događa kada je baterija došla do određenog napona (4.2V), odnosno ovdje se baterija nadopunjuje do maksimalnog kapaciteta te se u tom trenutku punjenje prekida



Sl. 3.20. Dijagram punjenja li-ion baterije<sup>23</sup>

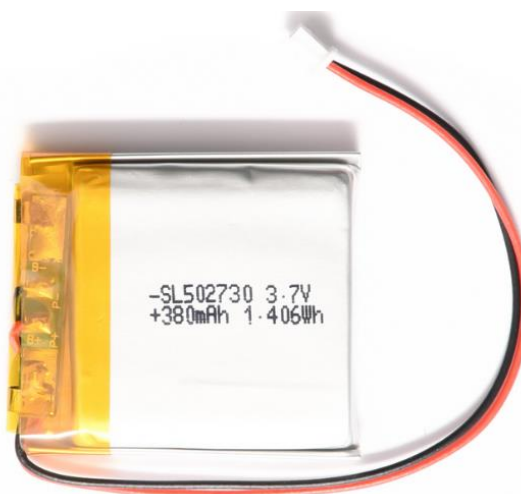
Navedene faze punjenja je moguće zadovoljiti korištenjem posebnih punjača koji koriste integrirani CC/CV (*engl. constant current / constant voltage*) način punjenja kao što je primjer na slici 3.21.

<sup>23</sup> [https://batteryuniversity.com/learn/article/charging\\_lithium\\_ion\\_batteries](https://batteryuniversity.com/learn/article/charging_lithium_ion_batteries)



Sl. 3.21. Punjač za li-ion baterije<sup>24</sup>

Ulazni napon punjača je 4.5V do 5.5V te je napajanje putem USB kabela pristupačno, ali osim toga su dostupni pinovi IN+ i IN- za druge oblike povezivanja napajanja. Maksimalna izlazna struja iznosi 1A. Ova verzija punjača također dolazi s JST HX konektorom koji je idealan za veliku većinu li-ion baterija kao što je primjer baterije na slici 3.22. Također je omogućena zaštita od kratkog spoja, prevelikog i preniskog napona.



Sl. 3.22. Li-ion baterija s JST HX konektorom<sup>25</sup>

<sup>24</sup> <https://e-radionica.com/hr/punjac-za-litijsku-bateriju-e-radionica.html>

<sup>25</sup> <https://e-radionica.com/hr/li-ion-baterija-380mah-3-7v.html>

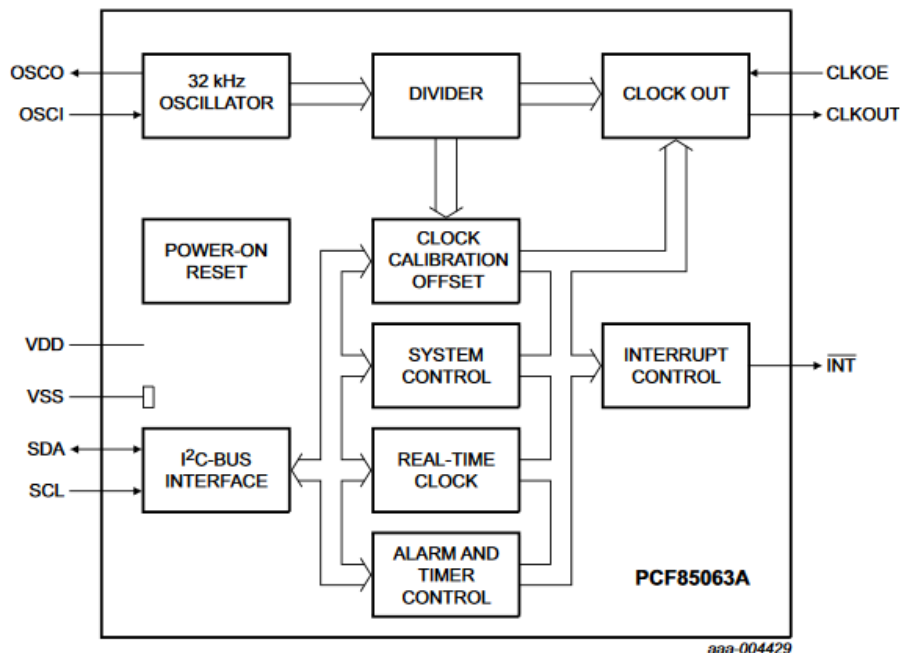


### 3.9. PCF85063A RTC

PCF85063A je CMOS (*engl. Complementary Metal Oxide Semiconductor*) RTC (*engl. Real Time Clock*) računalni sat za praćenje vremena i datuma. Podaci i adrese koje se koriste za komunikaciju s mikroupravljačem se prenose serijskim putem preko I2C sabirnice. Također je omogućeno korištenje CR2032 litijske baterije kojom se omogućuje da podešene postavke vremena i datuma ostanu spremljene. Osnovne značajke [29] PCF85063A RTC-a su sljedeće:

- Praćenje godine, mjeseca, dana, vikenda, sati, minuta i sekundi
  - Frekvencija 32.768 kHz (kvarcni kristal)
  - Radni napon: 0.9V – 5.5V
  - 400kHz dvolinijska I2C sabirnica
  - Funkcija alarma i štoperice
  - *Interrupt* pin

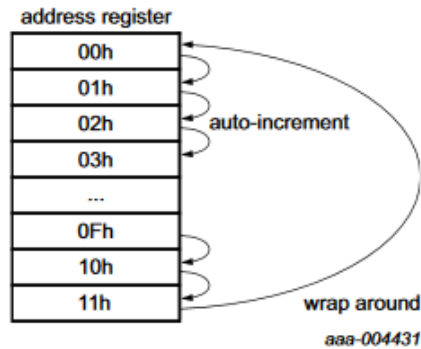
Kao što se može primijetiti na slici 3.23, pomoću I2C sabirnice kao i kod ostalih I2C uređaja se omogućuje sučelje sa SCL ulazom (*engl. Serial Clock Line*) i SDA ulazom/izlazom (*engl. Serial Data Line*) pinovima. Od ostalih ponuđenih pinova se ističe INT (*engl. Interrupt*) izlaz pin koji u ovom slučaju omogućuje buđenje mikroupravljača iz *sleep* stanja.



SI. 3.23. Blok dijagram PCF85063A sklopa<sup>26</sup>

<sup>26</sup> <https://www.nxp.com/docs/en/data-sheet/PCF85063A.pdf>

PCF85063A sadrži 18 8-bitnih registra koji automatski povećavaju svoju adresu registra pomoću 32.768 kHz oscilatora te se upravo tako omogućuje uloga sata stvarnog vremena. Ugrađeni adresni registri će se automatski povećavati nakon svakog čitanja ili upisivanja bajta podataka do registra s adresom 11h. Nakon toga će se automatsko povećavanje prebaciti na registar s adresom 00h i tako u krug. Ovaj proces je prikazan na slici 3.24.



Sl. 3.24. Upravljanje adresnim registrima<sup>27</sup>

Uloge pojedinih registara su opisani u tablici 3.7. Registri zaduženi za pohranu sekundi, minuta, sati, dana, vikenda, mjeseci, godina te odgovarajući registri za alarm su kodirani korištenjem BCD (*engl. Binary Coded Decimal*) formata. Onog trenutka kada se jedan od RTC registara zapiše ili pročita, sadržaj svih brojača vremena se zaustavi te se tako sprječava neispravno pisanje ili čitanje sata i kalendara.

<sup>27</sup> <https://www.nxp.com/docs/en/data-sheet/PCF85063A.pdf>

**Tab. 3.7.** Pregled registara i njihova uloga

ADRESA REGISTRA	IME REGISTRA (ULOGA)
00h	Kontrolni/status registar
01h	Kontrolni/status registar
02h	Offset registar (omogućuje podešavanje sata)
03h	RAM bajt
04h	Sekunda
05h	Minuta
06h	Sat
07h	Dan
08h	Vikend
09h	Mjesec
0Ah	Godina
0Bh	Sekunda (alarm)
0Ch	Minuta (alarm)
0Dh	Sat (alarm)
0Eh	Dan (alarm)
0Fh	Vikend (alarm)
10h	Brojač ( <i>timer</i> funkcija)
11h	Brojač ( <i>timer</i> funkcija)

### 3.10. MySQL baza podataka

MySQL predstavlja RDBMS (*engl. Relation database management system*) odnosno sustav za upravljanje nad relacijskim bazama podataka. Relacijske baze podataka uređuju podatke u jednu ili više tablica u kojima podaci mogu ili ne moraju biti u odnosu. Osnovne prednosti [30] korištenja MySQL baze podataka su sljedeće:

- Sustav baza podataka koji se koristi na webu i pokreće na serveru
- Prigodan za male i velike aplikacije
- Brz, pouzdan i jednostavan za korištenje
- Koristi SQL standard
- Omogućuje rad na različitim platformama
- Besplatan za korištenje (licenca otvorenog koda)

Kreiranje baze podataka je omogućeno uz pomoć alata *phpMyAdmin* koji je objašnjen u potpoglavlju 3.12.

### 3.11. PHP

PHP [31] je skriptni jezik i alat za kreiranje dinamičkih i interaktivnih web stranica te se savršeno uklapa u rad s MySQL bazom podataka. PHP datoteke mogu sadržavati HTML, CSS, JavaScript i PHP kod koje se izvršavaju na serveru, a rezultat se vraća kroz preglednik u obliku HTML-a. U nastavku se nalazi popis prednosti PHP jezika:

- Generiranje dinamičkih web stranica
- Kreiranje, otvaranje, čitanje, pisanje, brisanje i zatvaranje datoteka na serveru
- Slanje i primanje kolačića (*engl. cookies*)
- Kreiranje, brisanje i uređivanje podataka u bazi podataka
- Enkripcija podataka
- Dostupan na različitim platformama
- Besplatan za korištenje

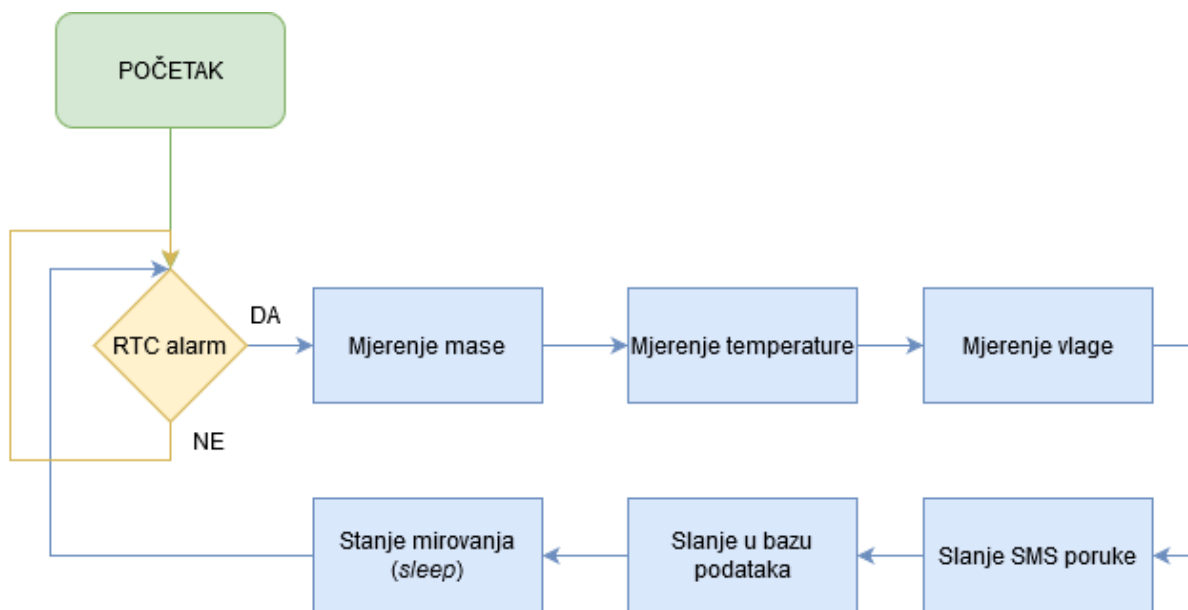
### 3.12. phpMyAdmin

phpMyAdmin [32] je besplatni administracijski alat pisan u PHP jeziku s namjenom za rukovanje nad MySQL i MariaDB bazama podataka. Osim rada s bazama podataka, vrlo bitna stavka je uloga besplatnog web hostinga. U nastavku se nalazi popis značajki *phpMyAdmin* alata:

- Omogućuje web sučelje
- Podržava MySQL i MariaDB
- Podržava pretraživanje, kreiranje, brisanje i izmjenjivanje baza podataka, tablica i polja
- Uvoz podataka iz CSV-a (*engl. Comma-separated values*) i SQL-a (*engl. Structured Query Language*)
- Izvoz podataka u CSV, SQL, XML, PDF, Word, Excel
- Administracija više poslužitelja odjednom
- Dostupan na raznim platformama
- Besplatan za korištenje

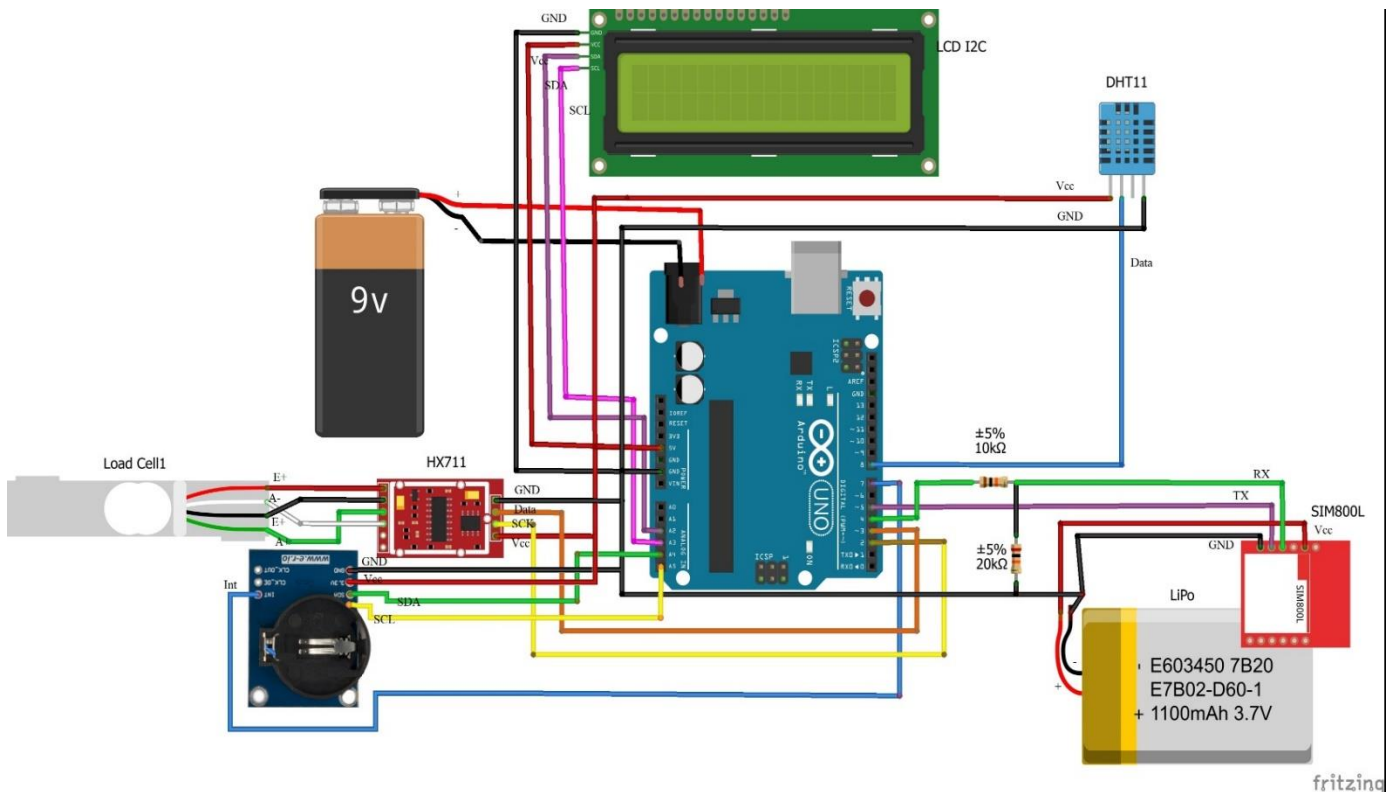
## 4. PAMETNA KOŠNICA

Blok dijagram na slici 4.1 opisuje tijek izvođenja događaja u programskom kodu, a sve ovisi o jednom uvjetu. Programski kod se nalazi u petlji, a sat stvarnog vremena koji ima svoju bateriju u određenom vremenu (jednom dnevno) šalje signal na *interrupt* pin. Odnosno, kada je zadano vrijeme alarma isto kao i stvarno vrijeme, Arduino se postavlja u aktivno stanje, mjeri se masa, temperatura i vlaga te se mjerene informacije šalju putem SMS poruke korisniku, te se također šalju u bazu podataka. Nakon izvršenog slanja, Arduino se postavlja u stanje mirovanja u kojemu je sve dok se ponovno ne pogodi zadano vrijeme alarma sa stvarnim vremenom.



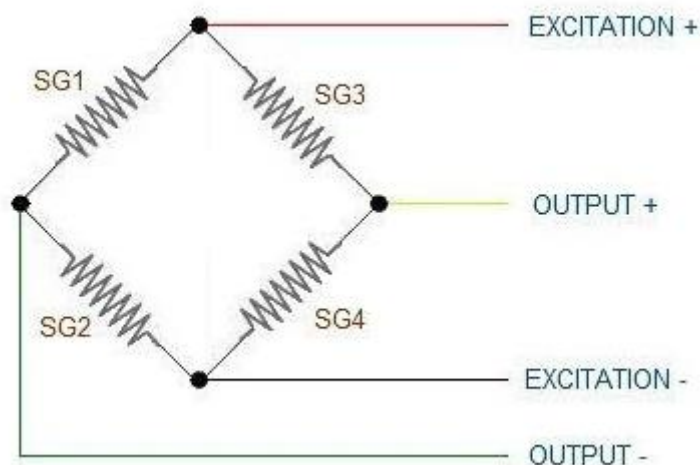
Sl. 4.1. Blok shema pametne košnice

Pomoću alata *Fritzing* koji omogućuje unos brojnih elektroničkih komponenti i modula, kreirana je shema projekta. Elektronička shema projekta, također kreirana u *Fritzing* alatu nalazi se u prilogu P.4.1.



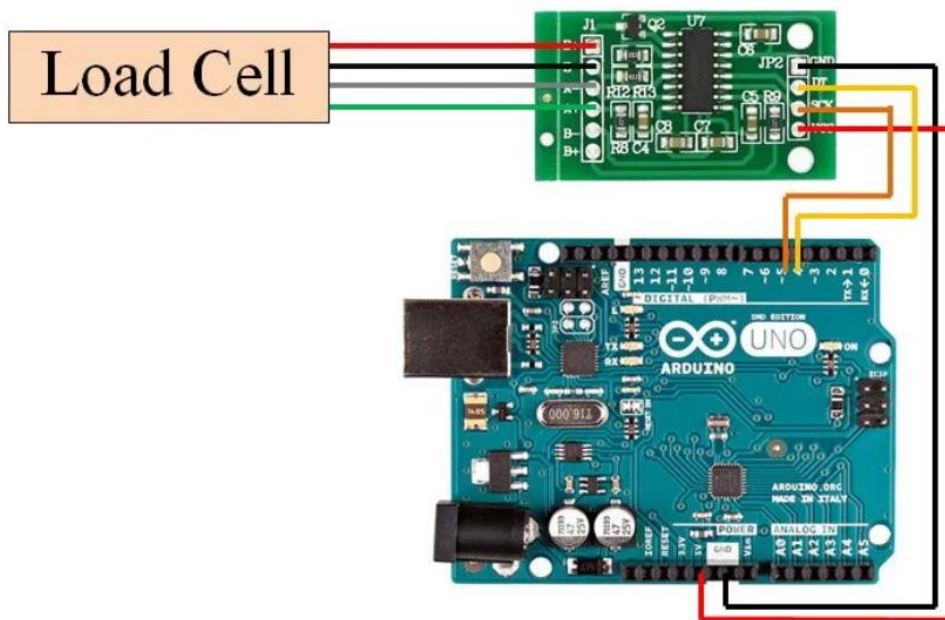
Sl. 4.2. Shema pametne košnice

Senzor mase koji se koristi kao senzor mase ima četiri izlaza koja su označena kao *Excitation+* (E+) i *Excitation-* (E-) te se koriste za napajanje dok se druga dva izlaza *Output+* i *Output-*, često označeni kao A+ i A- (kao oznaka za korištenje kanala A) koriste za promjenu napona koji se mjeri. Upravo takvo označavanje pinova se koristi na HX711 pretvaraču, dok s druge strane se nalaze pinovi za napajanje (GND i VCC) te pin za prijenos podataka (DT) i serijski *clock* pin (SCK). Pinovi za napajanje se na Arduino UNO pločicu spajaju na GND odnosno 5V pinove dok se pinovi DT i SCK spajaju na neki od digitalnih ulaza kao što je prikazano na slici 4.3.



Sl. 4.3. Wheatstone bridge izlazi<sup>28</sup>

Za potrebe ovog projekta se koristi senzor mase nosivosti 200 kg te iako postoje drugačije izvedbe tenzometra, koristi se verzija kao na slici 3.9 jer se tako omogućuje fizički jednostavnija izrada metalne konstrukcije na kojoj stoji košnica.



Sl. 4.4. Spajanje senzora mase i HX711 pretvarača sa Arduino UNO pločicom<sup>29</sup>

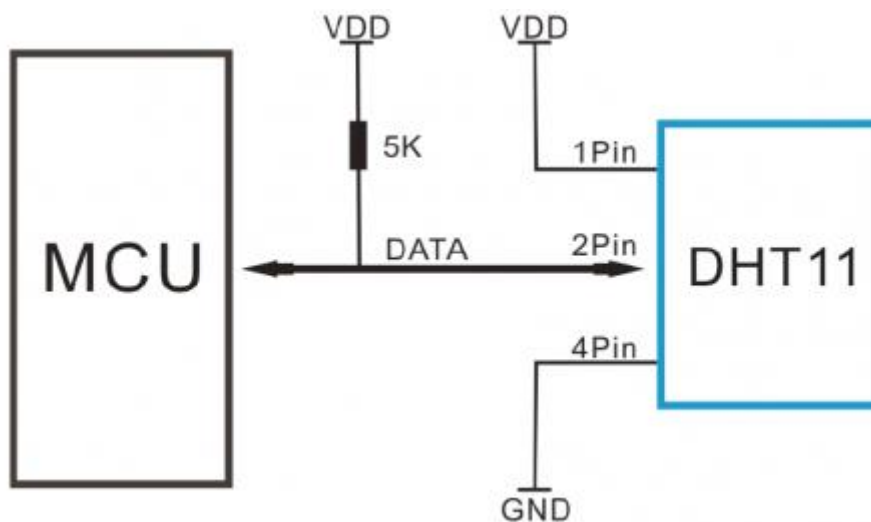
<sup>28</sup> <https://trybotics.com/project/How-to-Interface-HX711-Balance-Module-With-Load-Ce-10864>

<sup>29</sup> <https://www.instructables.com/id/How-to-Interface-HX711-Balance-Module-With-Load-Ce/>

DHT11 senzor temperature ima radni napon između 3-5.5 V što ga čini pogodnim za 5 V pin na Arduino pločici, a u posebnim uvjetima je moguće spojiti 100nF kondenzator između napajanja (VDD i GND) kako bi se omogućilo filtriranje snage. Izlaz sa senzora je potrebno spojiti na neki od digitalnih ulaza na Arduino pločici kao što je prikazano na slici 4.5.

Neke izvedbe DHT11 senzora dolaze s tri ili četiri pina. U svakom slučaju treba provjeriti raspored pinova kako se napajanje ili digitalni izlaz ne bi zamijenili i tako uništili čip u senzoru. U verziji s četiri pina potrebno je koristiti 10 k $\Omega$  *pull-up* otpornik između +5 V i pina za digitalni izlaz.

Za potrebe ovog projekta je sasvim dovoljan DHT11 jer neće biti potrebe mjeriti negativnu temperaturu s obzirom na to da će se koristiti u ljetnim mjesecima.

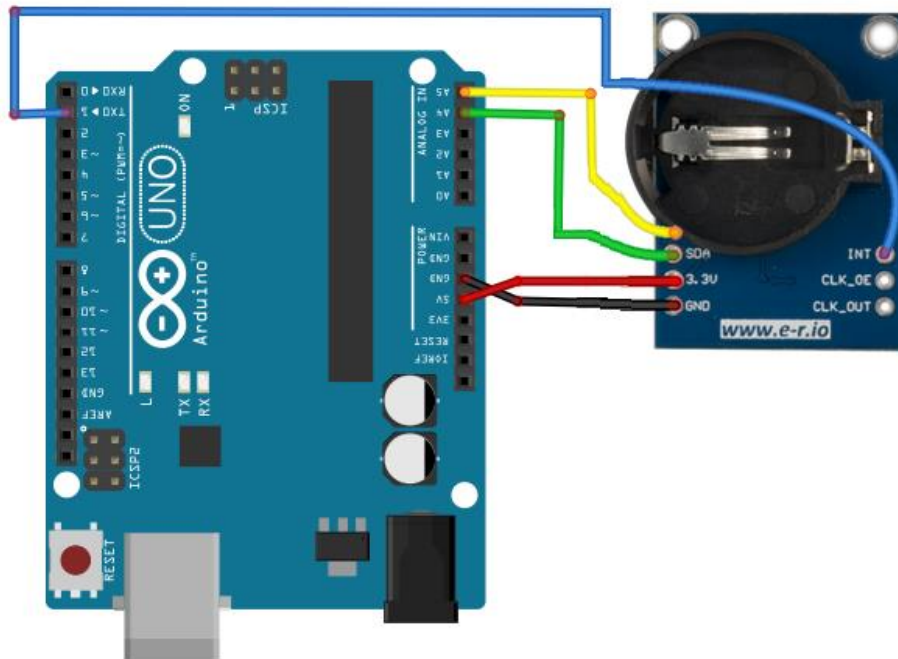


Sl. 4.5. Spajanje DHT11 senzora sa mikroupravljačem<sup>30</sup>

<sup>30</sup> <https://www.gorilladistribution.com.au/product/dht11-temperature-humidity-sensor/>



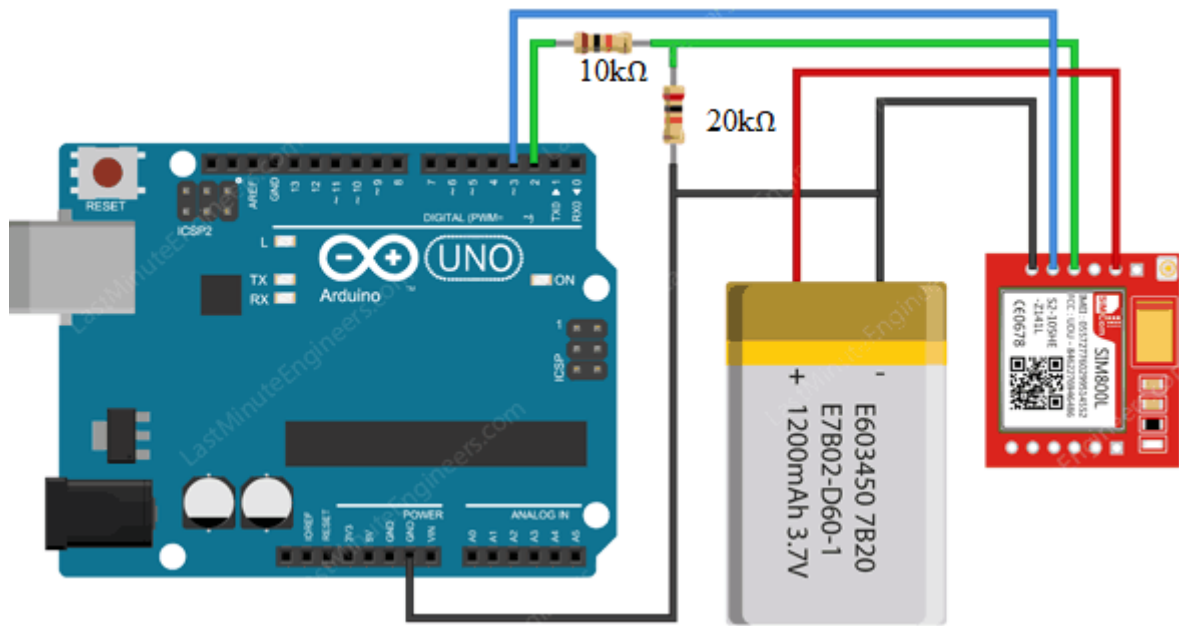
PCF85063A sat stvarnog vremena je podešen tako da jednom dnevno preko *interrupt* pina pošalje signal Arduino platformi i probudi ga iz stanja mirovanja. Kada se zadovolji uvjet da je podešeno vrijeme alarma isto kao i trenutno vrijeme, okida se impuls na *interrupt* pinu koji je spojen na jedan od digitalnih ulaza na Arduino pločici. Napajanje RTC-a je moguće do maksimalno 5.5 V što čini pin 5 V na Arduino idealnim za napajanje. Osim napajanja, RTC koristi SCL i SDA pinove za slanje serijskih podataka. Kao i LCD I2C modul, spajanje se na A4 i A5 analogni ulaz.



Sl. 4.6. Spajanje PCF85063A RTC-a s Arduino pločicom

Odašiljanje mjerenih informacija putem SMS poruke i u bazu podataka je omogućeno korištenjem SIM800L modula. Iako na tržištu postoje razne verzije GSM/GPRS modula koje koriste različite radne napone i različite generacije mobilnih mreža (3G, 4G), za potrebe ovog projekta SIM800L zbog svoje pristupačnosti i niske cijene u potpunosti zadovoljava sve zadane uvijete.

Spajanje SIM800L modula na Arduino UNO pločicu se izvodi tako da se TXD pin može direktno spojiti na jedan od PWM digitalnih pinova kao što je D3 pin. RXD pin se može direktno spojiti jer Arduino koristi 5 V za logičke razine (0 i 1) dok SIM800L modul koristi 3.3V za logičke razine te po tome RXD pin nije 5V tolerantan. To znači da *Tx* signal koji dolazi s Arduina mora biti spušten na 3.3 V kako ne bi došlo do oštećenja SIM800L modula. Rješenje je izvedeno spajanjem 10 k $\Omega$  otpornika između SIM800L RXD pina i Arduino D2 pina te dodatnim 20 k $\Omega$  otpornikom koji je spojen između SIM800L RXD pina i GND-a.



Sl. 4.7. Spajanje SIM800L modula sa Arduino UNO pločicom<sup>31</sup>

<sup>31</sup> <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>

## 4.1. Arduino programski primjer za mjerenje mase

Prije svega je potrebna biblioteka za komuniciranje s HX711 pretvaračem koji sadrži potrebne funkcije za čitanje vrijednosti. Osim toga su potrebna dva programska koda. Prije pravilnog korištenja je potrebno kalibrirati senzor mase, odnosno, potrebno je pronaći faktor kalibracije koji je kasnije moguće koristiti u ostalim programskim rješenjima.

```
#include "HX711.h" //uključivanje biblioteke
#define DOUT 2 //definiranje DT pina
#define CLK 3 //definiranje SCK pina
//faktor kalibracije koji je u početku nasumična vrijednost do onog
//trenutka kada je kalibracija uspješno izvedena. Odnosno kada load cell
//prikazuje točnu vrijednost.
float calibration_factor = 782.4;
HX711 scale;

void setup() {
  Serial.begin(9600);
  //inicijalizacija HX711 biblioteke sa pinovima DOUT i CLK
  scale.begin(DOUT, CLK);
  //postavi 'SCALE' vrijednost koja se koristi za pretvaranje 'sirovih'
  podataka u podatke koji su čitljivi.
  scale.set_scale(calibration_factor);
  //postavi offset vrijednost, odnosno 'resetiraj' na nulu
  scale.tare()
}

void loop() {
  scale.set_scale(calibration_factor);

  if (Serial.available()){ //petlja za omogućivanje pisanja u SerialMonitor
    char z = Serial.read(); //čitanje sa tipkovnice

    //ako je unesen znak '-', smanji faktor kalibracije za 5
    if (z == '-') calibration_factor -= 5;
    //ako je unesen znak '+', povećaj faktor kalibracije za 5
    if (z == '+') calibration_factor += 5;
  }
  Serial.print("Calibration factor: ");
  //ispiši trenutnu vrijednost faktora kalibracije
  Serial.print(calibration_factor);
  Serial.print("\tWeight:\t");
  //funkcija get_units() vraća omjer 'sirove' vrijednosti i faktora
  kalibracije te izvršava 10 čitanja
  //1 označava broj decimala nakon decimalne točke
  Serial.println(scale.get_units(10), 1); }
```

**Programski kod 4.1.** Kalibriranje i ispis vrijednosti senzora mase

Nakon programskog rješenja za kalibraciju senzora u glavnom programskom kodu samo je potrebno definirati varijablu koja će predstavljati faktor kalibracije, inicijalizirati skalu unutar `setup()` funkcije te unutar `loop()` funkcije pozvati `get_units()` funkciju koja vraća masu mjenog objekta.

## 4.2. Arduino programski primjer za mjerenje temperature i vlage

Prije svega je potrebna biblioteka za pristup funkcijama za komuniciranje s DHT serijom senzora relativne vlage i temperature.

```
#include "DHT.h" //uključivanje biblioteke
#define DHTPIN 2 //definiranje pina
#define DHTTYPE DHT11 //definiranje DHT serije (u ovom slučaju DHT11)

//inicijalizacija DHT senzora sa pinom i serijom senzora
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin(); //setup
}

void loop() {
  //pročitaj i pohrani relativnu vlagu
  float hum = dht.readHumidity();
  //pročitaj i pohrani temperaturu (Celsius default)
  float temp = dht.readTemperature();

  Serial.print("Vlaga: ");
  Serial.print(hum); //ispiši relativnu vlagu
  Serial.println("Temperature: ");
  Serial.print(temp); //ispiši temperaturu
  Serial.print("°C ");
}
```

**Programski kod 4.2.** Očitavanje temperature i vlage

### 4.3. Arduino programski primjer za ispis na LCD ekran

Nakon dodavanja potrebne biblioteke `LiquidCrystal_I2C` za pristup funkcijama za prikaz znakova na LCD ekran, potrebno je također uključiti `Wire` biblioteku koja omogućuje komunikaciju s I2C uređajima kao što je ovdje slučaj.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//postavljanje LCD adrese na 0x27 sa 16 znakova u 2 retka
LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
  lcd.init(); //inicijalizacija LCD
  lcd.backlight(); //paljenje pozadinskog svjetla (LED diode)
  Serial.begin(9600);
}

void loop()
{
  //brisanje LCD ekrana i postavljanje kursora u gornji lijevi kut
  lcd.clear();
  //postavljanje LCD kursora na željenu lokaciju
  //u ovom slučaju je to (0,0) ->nulti red, nulti stupac
  lcd.setCursor(0,0)
  lcd.print('Ispis na LCD');
}
```

**Programski kod 4.3.** Ispis teksta na LCD ekran

## 4.4. Arduino programski primjer za postavljanje i ispis vremena

Nakon uključivanja PCF85063A.h biblioteke za pristup odgovarajućim funkcijama, podešava se sat i datum korištenjem setTime() i setDate() funkcije. Zatim se u loop() funkciji ispisuje trenutno vrijeme svake sekunde.

```
#include <PCF85063A.h>
PCF85063A rtc;
void setup() {
  Serial.begin(115200);
  //postavljanje vremena (sati, minute, sekunde)
  rtc.setTime(6, 54, 00);
  //postavljanje datuma (dan u tjednu, dan, mjesec, godina)
  rtc.setDate(6, 6, 9, 2020);
}
void loop() {
  printCurrentTime(); delay(1000);
}
void printCurrentTime() {
  switch( rtc.getWeekday() )
  { case 0:
    Serial.print("Nedjelja, ");
    break;
  case 1:
    Serial.print("Ponedjeljak, ");
    break;
  case 2:
    Serial.print("Utorak, ");
    break;
  case 3:
    Serial.print("Srijeda, ");
    break;
  case 4:
    Serial.print("Cetvrtak, ");
    break;
  case 5:
    Serial.print("Petak , ");
    break;
  case 6:
    Serial.print("Subota , ");
    break; }
  Serial.print(rtc.getDay()); Serial.print(".");
  Serial.print(rtc.getMonth()); Serial.print(".");
  Serial.print(rtc.getYear()); Serial.print(". ");
  Serial.print(rtc.getHour()); Serial.print(":");
  Serial.print(rtc.getMinute()); Serial.print(":");
  Serial.println(rtc.getSecond()); }
```

**Programski kod 4.4.** Postavljanje i ispis vremena

## 4.5. SQL programsko rješenje

Kreirana je tablica `ebeetable` sa stupcima `id`, `weight`, `temp`, `hum`, `date`. Decimalnim vrijednostima (`weight`, `temp`, `hum`) su zapisane u obliku (6,3) i (3,1) što znači da je najviše moguće zapisati četveroznamenkasti broj s dvije decimale, odnosno dvoznamenkasti broj s jednom decimalom.

```
CREATE TABLE `ebeetable`. (  
  `id` INT NOT NULL , PRIMARY KEY AUTOINCREMENT,  
  `weight` DECIMAL(6,2) NOT NULL ,  
  `temp` DECIMAL(3,1) NOT NULL ,  
  `hum` DECIMAL(3,1) NOT NULL ,  
  `date` DATE NOT NULL);
```

### Programski kod 4.5. Kreiranje tablice

Nadalje, unutar `addData.php` skripte se koristi `INSERT INTO` naredba za dodavanje vrijednosti `weight`, `temp`, `hum` u tablicu. Ove vrijednosti se pomoću globalne varijable `$_GET` dohvaćaju iz `URL`-a koji se putem `SIM800L` modula prosljeđuju web hostu.

```
INSERT INTO ebeetable (weight, temp, hum) VALUES (  
  ". $_GET["weight"] .",  
  ". $_GET["temp"] .",  
  ". $_GET["hum"] ."');
```

### Programski kod 4.6. Unos podataka u tablicu

Za prikaz podataka na web stranicu se pomoću skripte `indeks.php` koristi jednostavna SQL naredba `SELECT`, a pomoću HTML strukture, kreira se tablica unutar koje se prikazuju podaci iz baze podataka.

```
SELECT * FROM ebeetable;
```

### Programski kod 4.7. Odabir svih stupaca iz tablice

## 4.6. PHP programsko rješenje

Za potrebe ovog projekta kreirane su dvije `php` skripte. Skripta `addData.php` koja se putem `URL`-a prosljeđuje web hostu koristeći `SIM800L` modul. Druga skripta, `index.php` se koristi za prikaz podataka iz tablice unutar web preglednika. Spomenuta prednost PHP-a da datoteka može sadržavati HTML strukturu je upravo ovdje iskorištena.

Najprije je potrebno ostvariti vezu prema bazi podataka tako da su kreirane varijable `$servername`, `$username`, `$password`, `$dbname` te su im u obliku `string`-a dodijeljeni podaci za spajanje. Nakon toga se poziva `mysqli()` funkcija koja otvara novu vezu prema MySQL serveru, a prosljeđeni parametri su spomenute `string` varijable. Funkcija

je spremljena u `$conn` varijablu te se tako omogućuje provjera uspješnog spajanja i zatvaranje veze prema MySQL serveru korištenjem `close()` funkcije.

Nakon ostvarene veze prema bazi podataka mogu se slati upiti kao što su dodavanje novih ili ispis vrijednosti iz tablice, a to je omogućeno pozivanjem `query()` funkcije izvodi upite prema bazi podataka.

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "ebee";

    // Otvaranje veze prema bazi podataka
    $conn = new mysqli($servername, $username, $password, $dbname);

    if ($conn->connect_error) // Provjera veze
    {
        echo '<p>Connection failed</p>';
        die("Connection failed: " . $conn->connect_error);
    }
    // Dodavanje novih vrijednosti u tablicu
    $sql = "INSERT INTO ebeetable (weight, temp, hum) VALUES ('
    ". $_GET["weight"] .",
    ". $_GET["temp"] .",
    ". $_GET["hum"] ."')";
    $result = $conn->query($sql);

    $conn->close();
?>
```

**Programski kod 4.8.** Skripta *addData.php* za unos podataka u bazu

Skripta *index.php* nakon spajanja s bazom podataka ispisuje vrijednosti iz baze podataka unutar web preglednika koristeći određene *tabove*. Tako se koriste `<tr></tr>` tagovi koji definiraju red i `<th></th>` tagovi koji definiraju zaglavlje HTML tablice. Kako bi se vrijednosti iz baze podataka mogle ispisati prolazi se kroz `while` petlju i koristeći `mysqli_fetch_array()` funkciju se vraća rezultat u redovima kao numerički niz, a pridruženi `$query` parametar je rezultat spajanja na bazu podataka i SQL upita *SELECT*.



```

$sql = "SELECT * FROM beetable";
$query = mysqli_query($conn, $sql);
echo "<table class = 'table'>
<tr>
    <th>Masa</th>
    <th>Temperatura</th>
    <th>Vlaga</th>
    <th>Datum</th>
</tr>";
while($row = mysqli_fetch_array($query)) {
echo ' <tr>
    <td>'. $row['weight']. '</td>
    <td>'. $row['temp']. '</td>
    <td>'. $row['hum']. '</td>
    <td>'. $row['date']. '</td>
    </tr>';
} echo "</table>";

```

**Programski kod 4.9.** Skripta *indeks.php* za ispis podataka iz baze na web stranicu

## Prikaz podataka iz baze *ebee*

Masa	Temperatura	Vlaga	Datum
151	27	85	2020-09-02
25	27	85	2020-09-02
126	26	61	2020-09-08
138	25	63	2020-09-08

**Sl. 4.8.** Ispis vrijednosti iz baze podataka na web stranicu

## 4.7. Arduino programsko rješenje

Nakon dodavanja potrebnih biblioteka za rad s elektroničkim komponentama, definirani su odgovarajući pinovi za ulaze i izlaze te objekti za komunikaciju s komponentama.

```
#include <HX711.h>
#include <DHT.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <PCF85063A.h>
#include <avr/sleep.h>
#define DHTTYPE DHT11 //DHT11 tip senzora
#define WAKEUPPIN 2
#define DOUT 4 //load cell DT pin
#define CLK 5 //load cell SCK pin
#define DHTPIN 6 //DHT pin
#define GSM_RESET 7 //reset pin
#define SIM800_TX_PIN 10 //GSM TX pin
#define SIM800_RX_PIN 11 //GSM RX pin
#define STATUS 13 //ugrađena led dioda na Arduino pločici

SoftwareSerial gsm(SIM800_TX_PIN, SIM800_RX_PIN);
DHT dht(DHTPIN, DHTTYPE);
PCF85063A rtc;
LiquidCrystal_I2C lcd(0x27, 16, 2); //LCD adresa->0x27, 16 znakova, 2 retka
HX711 scale;
```

**Programski kod 4.10.** Definiranje biblioteka, pinova i objekata

U zaglavlju su također kreirane varijable koje će se koristiti u daljnjem kodu.

```
String phoneNumber = "+38598XXXXXXX";
//string URL za unos mjerenih vrijednosti u bazu podataka
String webPage = "https://ebee.000webhostapp.com/addData.php";
float temp, hum; //varijable za pohranu temperature i vlage
float calibration_factor = 782.4; //faktor kalibracije za LoadCell
```

**Programski kod 4.11.** Definiranje potrebnih javnih varijabla

Unutar `setup()` funkcije se inicijaliziraju navedeni objekti. Dodatno, postavlja se alarm korištenjem `setAlarm()` funkcije. U slučaju da se funkciji proslijedi vrijednost 99 za određeno vrijeme, tada se ono neće zapisati u registar. Odnosno u ovom slučaju su za registar dan i vikend dodijeljene vrijednosti 99 što znači da će se alarm upaliti svaki dan u 18:00 sati. Nadalje, svaki put će se izvršiti `checkAlarm()` funkcija koja samo provjerava trenutno vrijeme alarma.

```

void setup() {
  Serial.begin(9600);
  scale.begin(DOUT, CLK);
  scale.set_scale(calibration_factor);
  scale.tare();
  dht.begin();
  lcd.init();
  lcd.backlight();
  pinMode(WAKEUPPIN, INPUT_PULLUP);
  rtc.setAlarm(00, 00, 12, 99, 99); //postavljanje alarma u 18:00h
  checkAlarm();
}

```

**Programski kod 4.12.** *setup()* funkcija

Unutar `loop()` funkcije se pozivaju kreirane metode za ispis vrijednosti na LCD zaslon i metoda za stvaranje poruke koja će se poslati u bazu podataka i u SMS poruci. Metode su objašnjene u daljnjem tekstu. Nakon 60 sekundi će se pokrenuti funkcija `sleepNow()` koja će postaviti Arduino u *sleep* režim rada.

```

void loop() {
  loadCell();
  readTemp();
  readHum();
  createMessage();
  delay(60000); // ugasi Arduino nakon jedne minute
  sleepNow();
}

```

**Programski kod 4.13.** *loop()* funkcija



**Sl. 5.1.** Ispis mjerenih vrijednosti na LCD zaslon

Najprije će se ugasiti LCD zaslon i stražnje svjetlo, a zatim se poziva `sleep_enable()` funkcija koja omogućuje *sleep* bit u *mcucr* kontrolnom registru. Zatim se uvodi ISR (*engl. Interrupt Service Routine*) odnosno upravitelj interruptima koji se može opisati kao događaj sa svojim setom instrukcija. Onoga trenutka kada se vanjski interrupt dogodi, procesor će prvo izvršiti kod koji je dostupan u ISR-u. Od parametara se prosljeđuje interrupt, funkcija `wakeUp()` i stanje *LOW*. Zatim se postavlja *sleep* način rada korištenjem `set_sleep_mode` funkcije te se gasi ugrađena led dioda (pin 13). Na kraju se poziva funkcija `sleep_cpu()` kojom se dovršava slanje Arduino mikroupravljača u *sleep* način rada.

```
void sleepNow() {
    lcd.noBacklight(); //Turn off backlight
    lcd.noDisplay(); //Turn off the display
    sleep_enable(); //Enable sleep bit in MCUCR
    attachInterrupt(0,wakeUp, LOW); //use int 0(pin 2)
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); //sleep mode
    sleep_cpu();
}
```

**Programski kod 4.14.** Funkcija za postavljanje Arduina u stanje mirovanja

Metode `loadCell()`, `readTemp()` i `readHum()` su prethodno objašnjene u odgovarajućim potpoglavljima 4.3.2 i 4.4.1 te one isključivo služe za očitavanje vrijednosti s odgovarajućeg senzora i ispis na LCD zaslon.

Prije nego što se objasni `createMessage()` metoda, potrebno je objasniti metodu `prepareData()` koja jednostavno očitava vrijednosti sa senzora te ih pakira za prosljeđivanje.

```
float prepareData(){
    float weight = 0;
    float temp = 0;
    float hum = 0;

    float weightData = scale.get_units();
    float tempData = dht.readTemperature();
    float humData = dht.readHumidity();

    weight += weightData;
    temp += tempData;
    hum += humData;
    delay(2000);
    return weight, temp, hum;}
}
```

**Programski kod 4.15.** Čitanje vrijednosti sa senzora i priprema za slanje

Podaci u takvom obliku se mogu proslijediti odgovarajućoj varijabli unutar `createMessage()` metode. Također je kreirano polje `buf[]` tipa `char` unutar kojega se sprema poruka koja se šalje. Kako bi se omogućilo pretvaranje `float` vrijednosti u `char`, potrebno je pozvati `dtostrf()` funkciju koja će odraditi upravo taj posao. Ova funkcija prima četiri parametra, a to su `float` vrijednost koja se pretvara, duljina `string`-a koji će se kreirati, broj decimala nakon decimalne točke i polje unutar kojega se sprema pretvoreni rezultat. Nakon što su mjerene vrijednosti spremne za slanje pozivaju se kreirane metode koje obavljaju određene AT naredbe.

```
void createMessage() {
    char buf[20];
    float data = prepareData();
    dtostrf(data, 3, 1, buf);

    String message = String("temp=");
    message += buf;

    gsm_fullPower();
    message.toCharArray(buf, 20);
    internet(buf);
    resetGsm();
    gsm_lowPower();
}
```

**Programski kod 4.16.** Glavna funkcija za odašiljanje podataka

Najprije se poziva `gsm_fullPower()` funkcija koja obavlja jednu AT naredbu `AT+CFUN=1,1` te se tako SIM800L modul postavlja u stanje potpune funkcionalnosti te u ovom trenutku modul najviše troši snage jer se ostvaruje veza s mobilnom stanicom.

```
void gsm_fullPower() {
    gsm.println("AT+CFUN=1,1");
}
```

**Programski kod 4.17.** Funkcija za postavljanje SIM800L modula u stanje pune funkcionalnosti

Nakon toga se poziva `Internet()` funkcija kojoj je proslijeđen parametar `buf` unutar kojega se nalazi poruka s vrijednostima sa senzora. No, prije samog uspostavljanja HTTPS servisa, poziva se dodatna funkcija `connectToNet()` unutar koje se najprije naredbom

*AT+CGATT=1* omogućuje GPRS spajanje uređaja na *packet domain* servis. Nakon toga se konfigurira nositelj i postavlja se dodijeljeni APN koji je u ovom slučaju *internet.ht.hr*. Nakon svega se naredbom *AT+SAPBR=1,1* može uspješno otvoriti GPRS kontekst.

```
Void connectToNet() {
    gsm.println(„AT+CGATT=1“);
    delay(1000);
    gsm.println(„AT+SAPBR=3,1,“CONTYPE“,“GPRS““);
    delay(1000);
    gsm.println(„AT+SAPBR=3,1,“APN“,“Internet.ht.hr““);
    delay(1000);
    gsm.println(„AT+SAPBR=1,1“);
    delay(3000);
}
```

#### Programski kod 4.18. Funkcija za postavljanje APN-a i otvaranje GPRS veze

Tek nakon što je otvorena GPRS veza se može inicijalizirati HTTP servis korištenjem *AT+HTTPIPINIT* naredbe. Ovo je prva naredba koja se mora poslati prije pokretanja HTTP usluge. Nakon toga se mogu postaviti parametri za HTTP poziv naredbom *AT+HTTTPARA* te se ovdje prosljeđuju definirani *string*-ovi (*webPage* i *message*) u zaglavlju koda.

Nakon prosljeđenih parametara, pokreće se *GET* sesija i zaustavlja se HTTP servis naredbom *AT+HTTPTERM*.

```
void internet(char *message) {
    connectToNet();
    gsm.println("AT+HTTPIPINIT");
    delay(1000);
    gsm.println("AT+HTTTPARA=“CID“,1");
    delay(1000);
    gsm.print("AT+HTTTPARA=“URL“,““);
    gsm.print(webPage);
    gsm.print(message);
    delay(1000);
    gsm.println("AT+HTTTPACTION=2");
    delay(3000);
    gsm.println("AT+HTTPTERM");
    disconnectFromNet();
    delay(10000);
}
```

#### Programski kod 4.19. Funkcija za slanje mjerenih podataka u bazu

Isto kako se prije inicijalizacije HTTP servisa morala otvoriti GPRS veza, tako se ista mora zatvoriti nakon obavljenog slanja. Za to se poziva funkcija `disconnectFromNet()` unutar koje se samo poziva naredba `AT+SAPBR=0,1` za zatvaranje GPRS konteksta.

```
void disconnectFromNet() {  
    gsm.println("AT+SAPBR=0,1");  
    delay(1000);  
}
```

**Programski kod 4.20.** Funkcija za zatvaranje GPRS veze

Nakon što se obavilo slanje podataka, poziva se `resetGsm()` funkcija čija je uloga samo da resetira SIM800L modul u slučaju da se konekcija iz nekog razloga nije prethodno zatvorila, te se tako sprječava neželjeno korištenje dodatnih podataka.

```
void resetGsm()  
{  
    digitalWrite(STATUS, 1);  
    digitalWrite(GSM_RESET, 0);  
    delay(100);  
    digitalWrite(GSM_RESET, 1);  
    digitalWrite(STATUS, 0);  
    delay(5000);  
}
```

**Programski kod 4.21.** Funkcija za resetiranje SIM800L modula

Nakon što se obavio restart modula poziva se zadnja funkcija `gsm_lowPower()` koja samo šalje naredbu `AT+CFUN=0,1` te se SIM800L modul postavlja u stanje minimalne funkcionalnosti gdje se najmanje troši snage.

```
Void gsm_fullPower() {  
    gsm.println("AT+CFUN=0,1"); }  
}
```

**Programski kod 4.22.** Funkcija za postavljanje SIM800L modula u stanje minimalne funkcionalnosti

## 5. ISPITIVANJE MOGUĆNOSTI SUSTAVA

Prije pisanja programskog rješenja bilo je potrebno upoznati se sa SIM800L modulom i potrebnim AT naredbama kako bi se ispitale mogućnosti modula. Tako su se naučile potrebne naredbe i njihova uloga. Ispitivanje se ostvarilo korištenjem jednostavne Arduino skripte koja omogućuje upisivanje AT naredbi unutar *SerialMonitor*-a te se tako dobiva povratna informacija za svaku naredbu.

Nakon što je definirana potrebna biblioteka i definirani su RX i TX pinovi, započinje se serijska komunikacija između Arduina i SIM800l modula, a u *loop()* funkciji se provjerava unos u *SerialMonitor*.

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); //RX TX

void setup(){
  //Započni serijsku komunikaciju između Arduina i Arduino IDE
  Serial.begin(9600);

  //Započni serijsku komunikaciju između Arduina i SIM800l
  mySerial.begin(9600);
}

void loop() {
  //Komunikacija između Software Serial i Serial porta
  if (mySerial.available())
    Serial.write(mySerial.read());

  //Komunikacija između Serial i Software Serial porta
  if (Serial.available()){
    while(Serial.available()){
      mySerial.write(Serial.read());
    }
  }
  mySerial.println();
}
```

**Programski kod 5.1.** Testiranje AT naredbi korištenjem *Serial Monitor*-a

Osim ispitivanja SIM800L modula, bilo je potrebno proučiti na koji je način najbolje definirati faktor kalibracije za senzor mase te kolika su odstupanja od stvarne vrijednosti. Senzor nad kojim se obavljalo ispitivanje je nosivosti 2 kg, dok će završna verzija koristiti senzor nosivosti 200 kg.

Prije nego što se kreirao Arduino programski primjer za traženje faktora kalibracija (Programski kod 4.1) najprije se kreirala skripta koja pozivom funkcije *get\_units()* vraća neobrađene vrijednosti koje nisu zanimljive i u takvom obliku se ne mogu koristiti. Vrijednosti se ispisuju



u *SerialMonitor* nakon što se postavi predmet (testni predmet mase 30 grama) na senzor mase i u tom slučaju se dobiva vrijednost -23472 s odstupanjem  $\pm 20$ . Kada se ta vrijednost podijeli s masom predmeta dobiva se točan faktor kalibracije. S obzirom na početna odstupanja neobrađene vrijednosti, stvarni rezultat mase odstupa samo  $\pm 2$  grama, što je za potrebe ovog projekta zadovoljavajuće. U situaciji kada se koristi neki drugi senzor mase mora se ponovno definirati odgovarajući kalibracijski faktor, u suprotnom, jednom kada se definira, nije potrebna ponovna kalibracija.

```
#include "HX711.h"
#define DOUT 3
#define CLK 2

HX711 scale(DOUT, CLK);

void setup() {
  Serial.begin(9600);
  scale.set_scale();
  scale.tare();

  Serial.println("Postavi predmet na vagu");
  while (!Serial.available()){};

  float value = scale.get_units(10);
  Serial.print("Vrijednost: ");
  Serial.print(value);
}

void loop() {}
```

**Programski kod 5.2.** Traženje neobrađenih mjerenih vrijednosti za faktor kalibracije

Napajanje sustava je omogućeno korištenjem dvije baterije. Li-ionska baterija za napajanje SIM800L modula i alkalna baterija za napajanje ostatka sustava. Kapacitet 9 V alkalne baterije je 550 mAh i prema tablici Tab. 5.1. treba napajati 6 elektroničkih komponenti čija ukupna potrošnja struje iznosi 148.1 mA. Prema formuli (5-1) to znači da će baterija trajati malo više od 3 sata. Treba imati na umu da će sustav raditi samo jednu minutu dnevno, a ostatak vremena se komponente nalaze u stanju mirovanja (*sleep*) gdje troše oko 138 mA. Najveći potrošač je LED dioda koja osvjetljuje pozadinu LCD zaslona. Pomoću funkcije *noBacklight()* to svjetlo se može ugasiti i tako uvelike smanjiti potrošnju struje. Uzevši to u obzir, baterija će trajati oko 15 sati.

$$\text{Trajanje baterije [h]} = \frac{\text{Kapacitet baterije [mAh]}}{\text{Potrošnja struje [mA]}} \quad (5-1)$$

Kako bi se trajanje baterije produžilo, moguće je koristiti Arduino pločicu koja troši manje struje kao što je Nano verzija koja troši minimalno 19 mA dok u *sleep* stanju troši 23  $\mu$ A. Također je moguće povoljno nabaviti 12 V akumulatorsku bateriju koja je prihvatljivih dimenzija i puno većih kapaciteta. Na taj način se trajanje sustava produžuje na desetak dana.

**Tab. 5.1.** Potrošnja struje elektroničkih komponenti

<b>Napajanje: 9V alkalna baterija (550 mAh)</b>	
<b>Komponenta</b>	<b>Potrošnja struje</b>
LCD LED	100 mA
Arduino UNO R3	45 mA – 80 mA
1602 LCD I2C	1.1 mA
HX711	< 1.5 mA
DHT11	0.5 mA
PCF85063A	0.2 $\mu$ A
<b>Napajanje: 3.7V li-ion baterija (2100 mAh)</b>	
<b>SIM800L stanje</b>	<b>Potrošnja struje</b>
<i>Sleep</i>	1 mA
<i>Stand by</i>	18 mA
GPRS	453 mA
<i>Transmisson burst</i>	2 A

SIM800L modul koristi 3.7 V li-ionsku bateriju čiji kapacitet iznosi 2100 mAh. Pošto je modul u aktivnom stanju samo na jednu minutu dnevno, a u *sleep* stanju troši 1 mA, baterija kapaciteta 2100 mAh je i više nego dovoljna za dugotrajno napajanje (87 dana). No treba uzeti u obzir da je u toj jednoj minuti moguće da modul zahtjeva do 2 A što će smanjiti trajanje baterije, ali usprkos tome je ova baterija vremenski prihvatljiva.

Nakon teorijskih izračuna su odrađena i praktična. Kako je poznato da alkalna baterija koja napaja gotovo cijeli sustav neće moći raditi duže od 15 sati, za testiranje potrošnje baterija je odabran rad od šest sati. Nakon zadanog vremena, na alkalnoj bateriji je izmjereno 3.27 V manje od početne vrijednosti kada je baterija bila puna kao što je prikazano u tablici 5.2. To dokazuje teorijsku pretpostavku o kratkotrajnosti i neučinkovitosti ovog napajanja. Zbog toga se planira preći na Arduino Nano pločicu i napajanje akumulatorskom baterijom od barem 4500 mAh što produžuje trajanje sustava i na više nego dovoljnih 30 dana. Voltaža litij ionske

baterije koja napaja SIM800L modul je ostala ista, jer je zadanih 6 sati rada bilo premalo da bi se primijetio pad napona baterije.

**Tab. 5.2.** Potrošnja baterija nakon šest sati rada

Baterija	Početna vrijednost [V]	Nakon 6 sati rada [V]
Alkalna 9V	9.34 V	6.07 V
Li-ion 3.7 V	4.02 V	4.02 V

U tablici 5.3. su prikazani rezultati ispitivanja senzora mase nakon 24 sata rada. Početna masa od 20 kg je ostala ista i nakon 24 sata, ali pri tome se mora uzeti u obzir da se ispitivanje održalo u temperaturno idealnoj okolini. Zbog toga je napravljeno ispitivanje kako bi se vidjele razlike prilikom porasta temperature. Prije samog zagrijavanja senzora, sobna temperatura je iznosila 22.8 °C, a teret na vagi je iznosio 20 kg. Nakon utvrđenih početnih vrijednosti, lagano se započelo sa zagrijavanjem sve do 32 °C nakon čega se vrijednost s 20 kg povisila na 20.8 kg te se na osnovu toga može zaključiti da s porastom temperature raste i vrijednost mase u odnosu  $\approx 80\text{g}/^\circ\text{C}$ . Ovaj test je teoretski potvrđen u potpoglavlju 3.4 gdje je opisano da širenjem tenzometra raste i otpor na krajevima terminala tenzometra.

**Tab. 5.3.** Utjecaj vremena i temperature na senzor mase

Temperatura [°C]	Početna vrijednost [kg]	Nakon 24 sati rada [kg]
22.8 °C	20 kg	20 kg
32 °C	20.8 kg	-

## 6. ZAKLJUČAK

Cilj diplomskog rada je bio dizajnirati pčelarsku vagu koja će pratiti stanje košnice i informacije (masa, temperatura, vlaga) odašiljati korisniku putem SMS poruke i u bazu podataka. Mjereni podaci se pomoću određenih senzora prosljeđuju Arduino platformi te se nakon obrade odašilju korištenjem SIM800L modula. Također je korištenjem RTC sata podešeno postavljanje vremena i alarma te se na taj način omogućilo jednodnevno mjerenje podataka. Samo u tom kratkom trenutku je Arduino platforma upaljena, inače je u stanju mirovanja jer se ne zahtjeva pristup podacima u svakom trenutku nego je bitna jednodnevna promjena podataka. Osim toga, na taj način se znatno smanjuje potrošnja baterije.

Nakon istraživanja tržišta je donesena odluka da se izradi verzija pčelarske vage kao na slici 2.2. Omjer cijena profesionalne pčelarske vage i potrebnih elektroničkih komponenti za realizaciju slične je jedan od ključnih razloga zašto se odlučilo na samostalnu izradu. Osim što su se realizirali zadani ciljevi, omogućeno je i proširenje sustava dodatnim sensorima i modulima. Ovisno o potrebama za izradu dodatnih pčelarskih vagi moguće je kreiranje sustava za registraciju i prijavu korisnika, grafički prikaz podataka i mobilna aplikacija za prikaz podataka.

Osim realiziranog rješenja, usvojena su osnovna i dodatna znanja koja će sigurno pomoći pri izradi budućih IoT projekata (pametni staklenik, sustav za praćenje potrošnje poljoprivredne mehanizacije).

## LITERATURA

- [1] Apiloger, <https://apiloger.com/apiloger.htm>, 22.9.2020.
- [2] SMSVAGA, <http://www.smsvaga.com/index.php/sr/sms-vaga>, 22.9.2020.
- [3] XLOGbee, <http://www.xlogbeevaga.com/tehnicekarakteristike.html>, 22.9.2020.
- [4] OpenHiveScale, <http://www.openhivescale.org/>, 22.9.2020.
- [5] BeeHiveMonitoring, <https://beehivemonitoring.com/en/content/8-manual>, 22.9.2020.
- [6] Elprocus, <https://www.elprocus.com/atmega328-arduino-uno-board-working-and-its-applications/>, 6.7.2020.
- [7] TheEngineeringProjects, <https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html>, 26.9.2020.
- [8] Atmega-Datasheet, Atmel, 2015, [https://web.archive.org/web/20160412093102/http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Summary.pdf](https://web.archive.org/web/20160412093102/http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf), 6.7.2020.
- [9] Arduino, <https://store.arduino.cc/arduino-uno-rev3>, 26.9.2020.
- [10] The Engineering Projects, <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html>, 6.7.2020.
- [11] Arduino, <https://www.arduino.cc/en/Reference/PortManipulation>, 6.7.2020.
- [12] Cisco, [https://www.cisco.com/c/en/us/td/docs/ios/12\\_4t/mw\\_ggsn/configuration/guide/ggsnov er.html](https://www.cisco.com/c/en/us/td/docs/ios/12_4t/mw_ggsn/configuration/guide/ggsnov er.html), 21.9.2020.
- [13] Tutorials Point, [https://www.tutorialspoint.com/gprs/gprs\\_architecture.htm](https://www.tutorialspoint.com/gprs/gprs_architecture.htm), 12.7.2020.
- [14] ElectronicsNotes, <https://www.electronics-notes.com/articles/connectivity/2g-gsm/network-architecture.php>, 26.9.2020.
- [15] Last Minute Engineers, <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>, 12.7.2020.
- [16] SIM800H&SIM800L\_Hardware Design\_V2.02, SimCom, 2015, <https://datasheetspdf.com/pdf-file/989664/SIMCom/SIM800L/11> 12.7.2020.
- [17] SIM800 Series\_AT COMmand Manual\_V1.09, SimCom, 2015, <https://e-radionica.com/productdata/SIM800-AT.pdf>, 12.7.2020.
- [18] Flintec, <https://www.flintec.com/weight-sensors/load-cells>, 21.9.2020.

- [19] 800loadcel, <https://www.800loadcel.com/load-cell-and-strain-gauge-basics.html>, 21.9.2020.
- [20] <https://medium.com/@encardio/strain-gauge-principle-types-features-and-applications-73fc998f0837>
- [21] ElectronicsTutorials, <https://www.electronics-tutorials.ws/blog/wheatstone-bridge.html>, 26.9.2020.
- [22] 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales, AVIA Semiconductor, [https://cdn.sparkfun.com/assets/b/f/5/a/e/hx711F\\_EN.pdf](https://cdn.sparkfun.com/assets/b/f/5/a/e/hx711F_EN.pdf), 7.7.2020.
- [23] DHT Humidity & Temperature Sensor, <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>, 7.7.2020.
- [24] Adafruit, <https://learn.adafruit.com/dht>, 7.7.2020.
- [25] Components101, <https://components101.com/16x2-lcd-pinout-datasheet>, 7.7.2020.
- [26] Battery University, [https://batteryuniversity.com/learn/archive/understanding\\_lithium\\_ion](https://batteryuniversity.com/learn/archive/understanding_lithium_ion), 8.7.2020.
- [27] Turbo Future, <https://turbofuture.com/misc/Which-is-better-Nickel-Metal-Hydride-NiMH-or-Lithium-Ion-Li-ion-batteries>, 8.7.2020.
- [28] Battery University, [https://batteryuniversity.com/learn/article/charging\\_lithium\\_ion\\_batteries](https://batteryuniversity.com/learn/article/charging_lithium_ion_batteries), 8.7.2020.
- [29] NXP, <https://www.nxp.com/docs/en/data-sheet/PCF85063A.pdf>, 6.9.2020.
- [30] w3schools, [https://www.w3schools.com/php/php\\_mysql\\_intro.asp](https://www.w3schools.com/php/php_mysql_intro.asp), 7.9.2020.
- [31] TutorialsPoint, <https://www.tutorialspoint.com/php/index.htm>, 26.9.2020.
- [32] SiteGround, [https://www.siteground.com/tutorials/phpmyadmin/#What\\_is\\_phpMyAdmin](https://www.siteground.com/tutorials/phpmyadmin/#What_is_phpMyAdmin), 26.9.2020.

## SAŽETAK

Ovaj rad se bavi razvojem sustava za mjerenje mase, temperature i vlage košnice. Sustav je u potpunosti automatiziran te je realiziran korištenjem Arduino platforme i omogućuje da se mjereni podaci jednom dnevno odašilju korisniku putem SMS poruke i u bazu podataka. Odašiljanje podataka je omogućeno korištenjem GSM/GPRS mreže, odnosno korištenjem SIM800L modula. Kao krajnji rezultat projekta, omogućeno je lakše praćenje potrošnje i unosa meda u košnice, a svojim dizajnom se omogućuje dodatno proširenje projekta.

**Ključne riječi:** Arduino, GSM/GPRS, SIM800L, masa, temperatura, vlaga, RTC, baza podataka, SMS

## **ABSTRACT**

### **SMART BEEHIVE**

This paper deals with the development of a system for measuring the mass, temperature, and humidity of the beehive. The system is fully automated and is realized using the Arduino platform and allows the measured data to be sent to the user once a day via SMS and in the database. Data transmission is enabled by using a GSM/GPRS network, i.e. by using the SIM800L module. As a final result of the project, it is easier to monitor the consumption and intake of honey in the beehives, and its design allows for further expansion of the project.

**Key words:** Arduino, GSM/GPRS, SIM800L, weight, temperature, humidity, RTC, database, SMS



## **ŽIVOTOPIS**

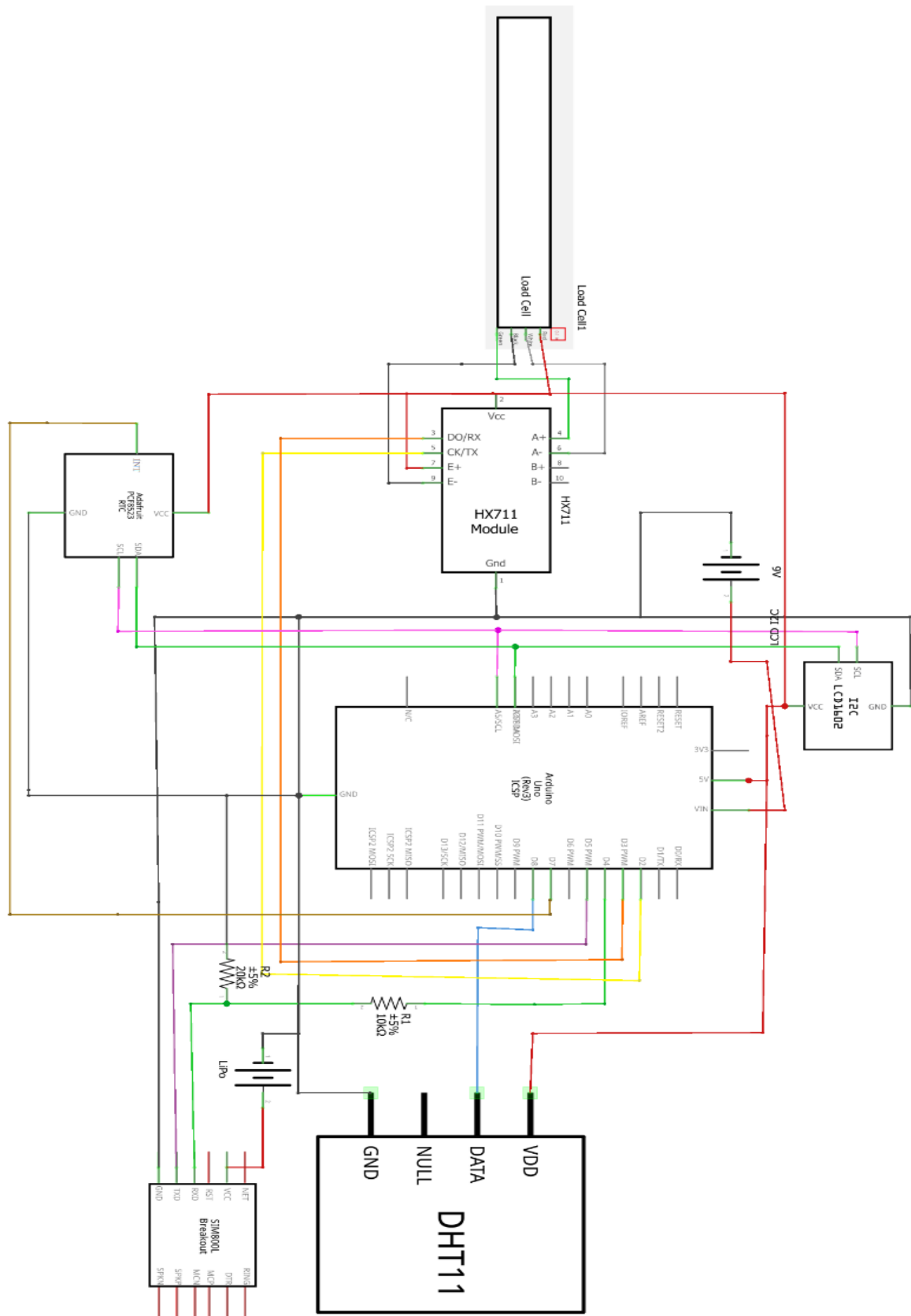
Stjepan Salopek rođen je 20. srpnja 1994. godine u Bjelovaru. Nakon završetka osnovne škole u Siraču, 2008. godine upisuje Tehničku školu u Daruvaru, smjer tehničar za računalstvo. Položenom državnom maturom, 2013. upisuje Elektrotehnički fakultet u Osijeku, stručni studij elektrotehnike, smjer informatika. Nakon obrane završnog rada na stručnom studiju, 2017. godine upisuje razlikovne obveze te nakon godinu dana upisuje diplomski studij računarstva, smjer podatkovne i informacijske znanosti.

Stjepan Salopek

---

# PRILOZI

fritzing



P.4.1. Električna shema pametne košnice