

# Sustav za nadzor parametara radne okoline za čuvanje glazbenih instrumenata

---

Florschütz, Luka

Master's thesis / Diplomski rad

2019

*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:200:556312>

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

*Download date / Datum preuzimanja:* **2024-05-14**

*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 diplomski studij**

**SUSTAV ZA NADZOR PARAMETARA RADNE OKOLINE**  
**ZA ČUVANJE GLAZBENIH INSTRUMENATA**

**Diplomski rad**

**Luka Florschütz**

**Osijek, 2018**

# SADRŽAJ

1. UVOD .....	3
2. SKLOPOVSKO I PROGRAMSKO RJEŠENJE .....	4
2.1 Sklopovsko rješenje.....	4
2.1.1 Croduino BASIC2 .....	5
2.1.2 DHT22 senzor temperature i vlage.....	6
2.1.3 Bluetooth .....	7
2.1.4 MicroSD to SD adapter .....	9
2.2 Programsko rješenje .....	11
2.2.1 Platforma Arduino .....	11
2.2.2 Croduino pločica .....	12
2.2.3 Fritzing software.....	12
2.2.4 Android operacijski sustav i Android Studio .....	13
2.2.5 PHP.....	14
3. OSTVARENJE SUSTAVA .....	15
3.1 Arduino programsko rješenje .....	16
3.2 Android aplikacija i baza podataka .....	20
4. TESTIRANJE SUSTAVA .....	30
5. ZAKLJUČAK .....	32
6. LITERATURA.....	33
POPIS POJMOVA .....	35
SAŽETAK.....	39
ŽIVOTOPIS .....	40
PRILOZI.....	41

# 1. UVOD

Glazba je umjetnost koju čovjek percipira, čuje i shvaća pomoću tonova koji glazbeni instrumenti proizvode. Svaki čovjek posjeduje jedan mali dio kojeg ispunjava glazba, kod nekog je to više a kod nekog manje izraženo. Glazbenu građu stvaraju zvukovi koji se pojavljuju udaranjem ili titrajem nekakvog tijela. Zvuk koji nastaje se sastoji od valova koji kroz zrak dolaze do čovjekovog uha. Kako bi stvorili glazbu, čovjeku je potreban barem jedan instrument (izvor) pri kreaciji. Instrument se definira kao sprava ili naprava, oruđe, alat ili mehanizam na kojem se proizvode tonovi. Tonovi koje proizvode instrumenti moraju biti što točniji tj. ugodniji uhu čovjeka kako bi se stvorila zadovoljavajuća glazba. Problem nastaje s vremenom. Instrumenti koji su najčešće izrađeni od probranih i najkvalitetnijih komada drva, gube na svojoj kvaliteti jer mikro klimatske promjene utječu na strukturu drveta, kao i laka, a time padaju i vrijednosti instrumenta, te se osjećaj sviranja pogoršava, a često nastaju i problemi na tonu koji proizvode, te na posljetku i veće štete na instrumentu. Najveći krivci za propadanje instrumenata su vlaga i temperatura u zraku. Primjerice, ako govorimo o instrumentu gitari, vlaga i temperatura utječu uvelike na njezine žice i tako ju kvare. Mnoge skupocjene gitare su završile kao simboli jedne ere ali s problemom što se više ne mogu koristiti. Na svu sreću, tehnologija uznapređuje svakoga dana i pojavila se ideja o brizi i očuvanosti instrumenata. Zadatak ovog diplomskog rada jest da se očuvaju instrumenti, koji su vrijedni i bitni korisnicima aplikacije na način raspolaganja podataka. Dobiveni podaci iz okruženja u kojem se nalazi sustav prenose se do korisnikovog osobnog telefona pri čemu aplikacija određuje postoji li opasnost od vlage i temperature u trenutnoj okolini. Podaci se spremaju i korisnik ih može pregledati u bilo kojem trenutku. Ukoliko se pojavi opasnost, sustav će poslati upozorenje korisniku na koje se mora adekvatno reagirati iz razloga što je okolina opasna za njegov instrument/e.

U drugom poglavlju diplomskog rada opisuje se sklopovsko i programsko rješenje sustava, u kojem će se opisivati komponente i parametri potrebni za realizaciju dva sustava koja međusobno komuniciraju. U trećem poglavlju se prikazuje ostvarenje ukupnog sustava koji pomoću komponenata dobiva informacije iz okoline, te uz pomoć komunikacije, sprema i šalje podatke na Android pametni telefon.

## 2. SKLOPOVSKO I PROGRAMSKO RJEŠENJE

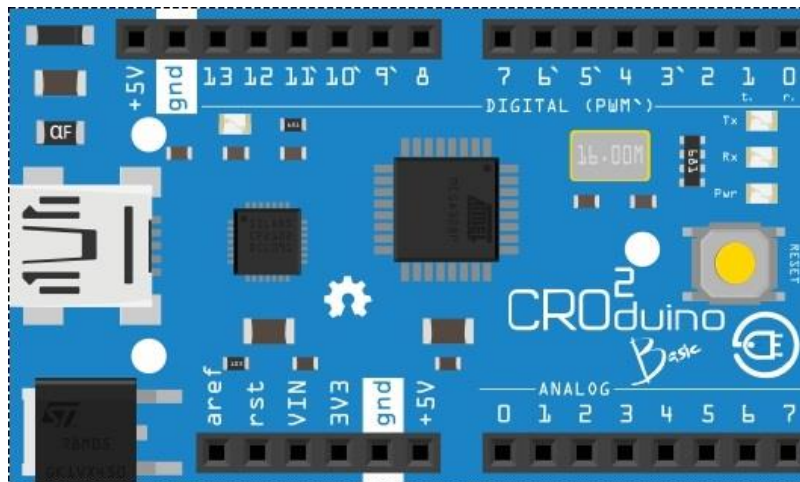
U ovom radu korištena je kombinacija sklopovlja i mobilne aplikacije, te će sljedeći dio rada biti podijeljen na sklopovsko i programsko (engl. *software*) rješenje. U sklopovskom dijelu prikazuje se i objašnjava cjeloukupno korišteno sklopovlje i njihova realizacija i komunikacija dijelova sustava, a u programskom dijelu objašnjavaju se korištene programske tehnologije za izradu mobilne aplikacije, te su prikazani i objašnjeni ključni dijelovi koda.

### 2.1 Sklopovsko rješenje

Sklopovlje (engl. *Hardware*) se sastoji od razvojne pločice Croduino Basic2, Bluetooth modula HC-06, senzora temperature i vlage DHT22, microSD kartice na koju se spremaju podaci iz okoline i služi kao povijest mjerenja korisniku Android pametnog telefona. Na razvojnoj pločici Croduino Basic2 su spojeni DHT22 senzor temperature i vlage, Bluetooth modul HC-06 i microSD kartica. Senzor temperature i vlage očitava trenutne vrijednosti temperature i vlage u okolini u kojoj se nalazi, a Bluetooth modul omogućava prijenos podataka između Croduina i Android pametnog telefona.

### 2.1.1 Croduino BASIC2

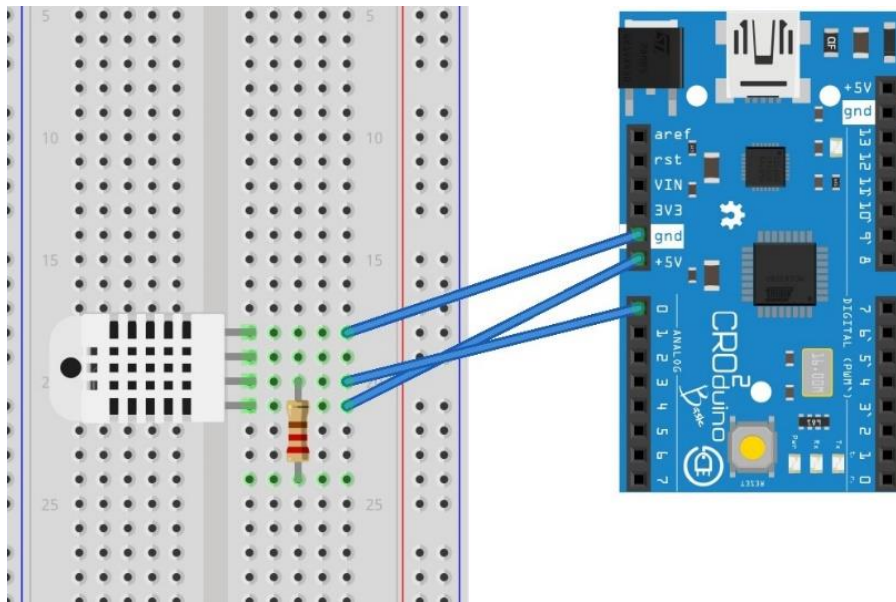
Croduino Basic2 prema [1] je Arduino kompatibilna pločica hrvatske proizvodnje prikazana na slici Sl. 2.1, koja je bazirana na Atmelovom mikroupravljaču (engl. *microcontroller*) Atmega328 i Silabsovom USB pretvaraču CP2102 uz brojne druge promjene dizajna naspram Arduina, točnije Arduina Nano s kojim je u potpunosti kompatibilan. Sastoji se od 22 ulaznih i izlaznih (engl. *input/output*) nožica koji su omogućeni kao digitalni ulazi ili izlazi (engl. *digitalWrite/digitalRead*), šest od kojih ima opciju modulacije širine impulsa (engl. *Pulse-width modulation, PWM*) upisivanja analognih vrijednosti (engl. *analogWrite*) u rezoluciji od 0 do 255, a 8 njih ima mogućnost analognog ulaza u vrijednosti od 0 do 1024 (engl. *analogRead*). PWM je tehnika dobivanja analognih vrijednosti pomoću digitalnih impulsa konstantne amplitude. UART most (engl. *Universal asynchronous receiver-transmitter bridge*), koji je omogućen preko USB (engl. *Universal Serial Bus*) veze i kabela, stvara komunikaciju između razvojne pločice i računala u oba smjera, pomoću naredbe učitavanja (engl. *upload*). Veličine je 3 x 5cm, te sadrži svjetleću diodu (engl. *Light Emitting Diode, LED*) na trinaestoj nožici prema [2]. Sastoji se još od regulatora koji pruža mogućnost vanjskog napajanja (baterije, pretvarača izmjenične u istosmjernu struju (engl. *AC/DC adapter*) i od zaštitnog osigurača.



Sl. 2.1. Prikaz Croduino Basic2 pločice

### 2.1.2 DHT22 senzor temperature i vlage

Na slici Sl. 2.2. prikazano je spajanje DHT22 senzora na razvojnu pločicu Croduino Basic2. DHT22 je kapacitativni senzor relativne vlažnosti i temperature zraka prema [3]. Sastoji se od 8-bitnog mikrokontrolera i OTP (engl. *Programmable non-volatile memory*) memorije preko kojih se izvršava kalibriranje svakog očitavanja. Komunikacija prema Croduinu se vrši preko nožica (engl. *Pin*) serijske podatkovne linije (engl. *Serial Data Line*) i I/O pin na Croduinu, te se tako stvara pojednostavljena single-bus komunikacija. Zbog svoje strukture koja funkcionira na principu glavni/podređeni (engl. *master/slave*), potreban mu je podizni (engl. *Pull-up resistor*) otpornik. Senzor ima raspon mjerenja temperature od  $-40\text{ }^{\circ}\text{C}$  do  $80\text{ }^{\circ}\text{C}$  i vlažnosti koju iskazuje od 0-100 %, a pogreška mu je  $\pm 2\%$  kod određivanja vlažnosti i  $\pm 0.5\text{ }^{\circ}\text{C}$  za temperaturu. Ovom senzoru potrebno je kratko vrijeme zagrijavanja kako bi dobili rezultat (potrebno zagrijavanje na 40 do  $80\text{ }^{\circ}\text{C}$ ).

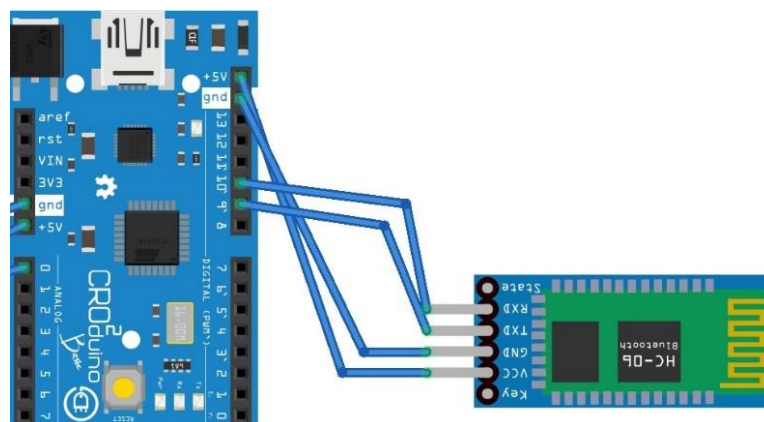


Sl. 2.2. DHT22 senzor temperature i vlage

### 2.1.3 Bluetooth

Bluetooth je tehnologija komunikacije između dva ili više uređaja koji mogu razmjenjivati ili prenositi podatke. Osmišljen je od strane Ericssona 1994. godine kao alternativa za RS-232 žični prijenos podataka. Kako bi se veza između dva uređaja uspostavila, šalju se radio valovi u frekvencijskom području između 2.4 i 2.48 GHz, te snage oko 1 mW prema [4]. Iz razloga što se komunikacija odvija preko radio veze povezani uređaji ne moraju biti u optičkoj vidljivosti jer signal prolazi kroz zidove i druge predmete, niti međusobno usmjereni kao npr. infracrvena komunikacija (engl. *Infrared communication, IR*). Vezu je moguće upostaviti u radiusu do 10 metara ako se radi o drugoj Bluetooth klasi koju koriste današnji pametni telefoni, a moguće ju je i uspostaviti na više od 100 metara u prvoj Bluetooth klasi. Bluetooth može biti povezan na 8 uređaja istovremeno u prostoru od 10 metara bez da smetaju jedan drugom. Puno je štedljiviji (baterija) i lako povezivi uređaj pa iz tog razloga je korišten u radu umjesto WiFi komunikacije.

U sustavu, Bluetooth je omogućen pomoću HC-06 modula koji se sastoji od četiri nožice prikazane na slici Sl. 2.3 Prva nožica (*Vcc*) je spojena na izvor od +5V na Croduinu BASIC2, druga nožica (*GND*) je nožica uzemljenja. Druge dvije nožice su serijski ulaz i izlaz modula koji se spaja na nožicu mikroupravljača. *TXD* predstavlja serijski izlaz, a *RXD* serijski ulaz modula, prema [5].



Sl. 2.3. Prikaz spajanja HC-06 modula na Croduino Basic2



Bluetooth komunikacija je ograničena svojim dometom, pa se tako razlikuju tri klase koje koriste određenu snagu kao izvor energije. Primjerice, Bluetooth klasa tri ima raspon od jednog metra pa iz tog razloga koristi vrlo malo energije za rad, koje je prikazano u tablici Tab. 2.1. Druga klasa Bluetooth komunikacije koristi neznatno više energije ali je komunikacija moguća u dometu između pet do deset metara koja se najčešće pojavljuje u mobilnim uređajima i u ovom diplomskom radu. Prva klasa Bluetootha se koristi u industrijskim slučajevima čiji je domet do 100 metara ali zbog tolikog dometa koristi i puno više energije.

**Tab. 2.1. Tablica dometa Bluetooth klase**

Klasa	Max snaga		Domet
	mW	dBm	m
<b>1</b>	<b>100</b>	<b>20</b>	<b>100</b>
<b>2</b>	<b>2.5</b>	<b>4</b>	<b>10</b>
<b>3</b>	<b>1</b>	<b>0</b>	<b>1</b>

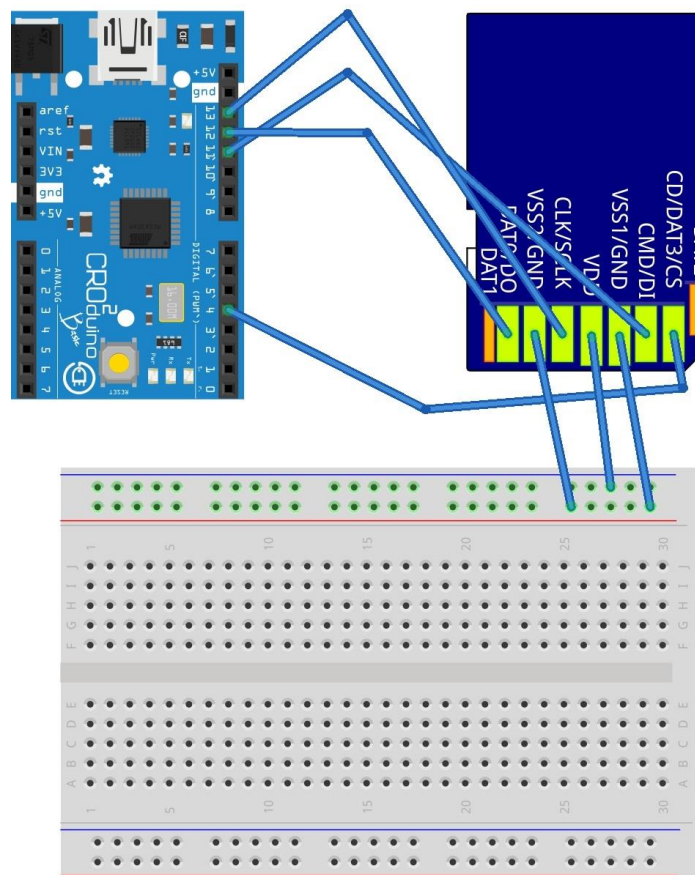
Brzine prijenosa podataka Bluetooth komunikacije su porasle nizom godina usavršavanja tehnologije. Prva inačica Bluetooth komunikacije koja je napravljena 1994. godine imala je prijenos podataka do 768 kbps, a današnja inačica ima brzinu do 48 Mbps prema tablici Tab. 2.2.

**Tab.2. 2. Brzine prijenosa podataka prema Bluetooth inačicama**

Verzija	Brzina prijenosa
Bluetooth 1.0 i 1.0B	768 kbps
Bluetooth 1.1	768 kbps
Bluetooth 1.2	1 Mbps
Bluetooth 2.0 i 2.1 EDR	3 Mbps
Bluetooth 3 + HS	24 Mbps
Bluetooth Smart (4.0 i 4.1)	24 Mbps
Bluetooth 5	48 Mbps

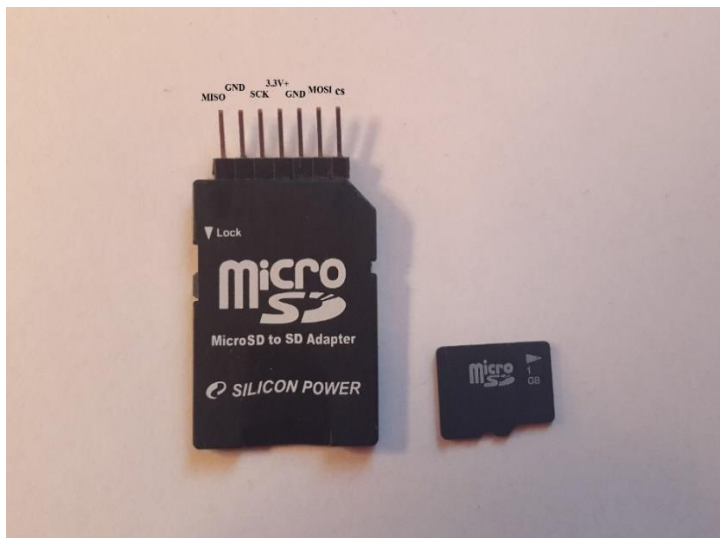
## 2.1.4 MicroSD to SD adapter

MicroSD adapter je uređaj koji omogućuje korisniku čitanje ili prijenos datoteka s microSD kartice na računalo. Adapter je veličine standardne SD kartice u koju se može umetnuti microSD kartica te spremati ili čitati željene podatke. MicroSD kartica u projektu služi kao memorija tj. baza podataka iščitanih vrijednosti s senzora DHT22. Komunikacija između kartice i Croduino Basic2 mikroupravljača koristi serijsko periferno sučelje (engl. *Serial Peripheral Interface, SPI*), koje zauzima digitalne nožice 11,12 i 13 na Croduino Basic2 prema slici Sl. 2.4, a digitalna nožica 4 poziva SD karticu za početak rada pomoću poziva *SD.begin()*. Nožice 11 i 12 se koriste zbog slanja podataka između njihovog glavnog i podređenog (engl. *Master – slave*) odnosno, microSD kartice i mikroupravljača prema [6] .



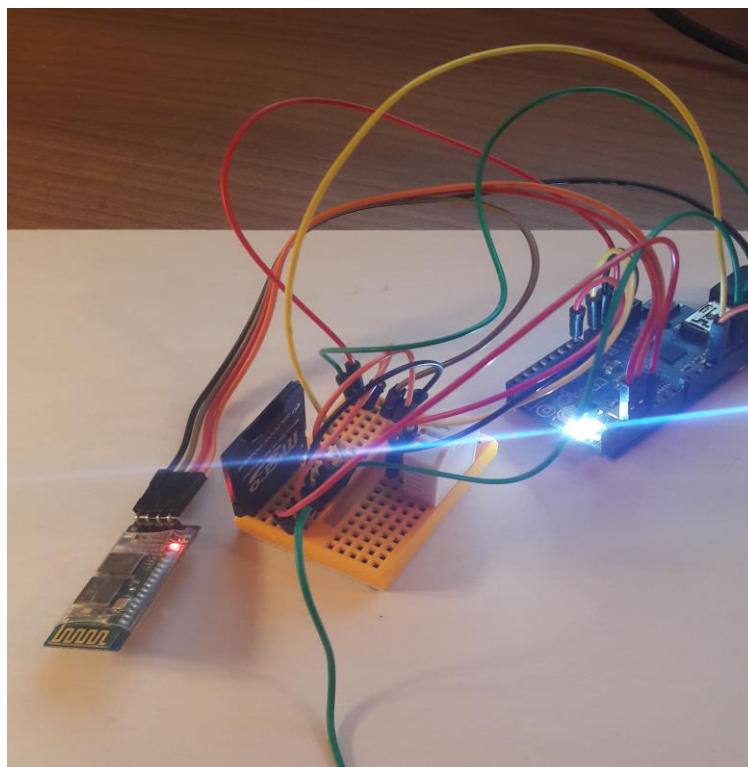
Sl. 2.4. Prikaz spajanja microSD to SD adaptera na Croduino Basic

Nožica 12 se naziva MISO zbog kratice glavnog ulaza, podređenog izlaza (engl. *Master In, Slave Out*), a nožica 11 MOSI zbog kratice glavni izlaz, podređeni ulaz (engl. *Master Out, Slave in*) prema [7] . Na slici Sl. 2.5. prikazan je adapter koji je meko lemljen na nožice.



Sl. 2.5. MicroSd to SD adapter

Sklopovlje Croduino sustava se sastoji od DHT22 senzora, microSD kartice i Bluetooth modula HC-06 koji su spojeni na nožice Croduino Basic2, a prikaz je na slici Sl. 2.6.



Sl.2.6. Prikaz sklopovlja Croduino Basic2

## 2.2 Programsko rješenje

U ovom potpoglavlju opisano je nešto više o Arduino programskoj tehnologiji i Android platformi koja je potrebna za realizaciju ukupnog sustava. Podatci koji se dobivaju sa senzora i koji se obrade s jednostavnim C programskim jezikom unutar Arduino IDE koriste se za izradu mobilne aplikacije. Nadalje, opisan je program Fritzing koji se koristi za izradu nacrt sklopovlja.

Mobilne aplikacije postale su nužni dio pri izradi složenijih sustava. Ponajviše govorimo o Android platformi, a u radu se koristi Android Studio koji je detaljnije opisan u potpoglavlju 2.2.4.

### 2.2.1 Platforma Arduino

Arduino je elektronička platforma koja je namijenjena za izradu elektroničkih projekata. Platforma se sastoji od sklopovskog dijela koji predstavlja mikroupravljač i programskog dijela koji se naziva IDE (engl. *Integrated Development Environment*) koji se izvodi na računalu te se pomoću njega programira upravljanje same pločice prema [8]. Pločica je nastala u Italiji na Ivrea Interaction Design Institute 2005. godine od strane grupe studenata. Svoju popularnost je stekao zbog cijene, jednostavnosti, prilagodbi na više platformi (engl. *Cross-platform*), otvorenog je koda (engl. *Open Source*) i zbog proširivog *hardwarea*. Programiranje pločica ne zahtjeva ništa drugo osim USB kabela koji se može povezati sa računalom, neovisno na kojem je računalo operacijskom sustavu. Programiranje se obavlja u vrlo sličnoj C++ verziji programskog jezika, koji je pojednostavljen do osnovnih naredbi za upravljanje i programiranje pločice. Arduino pločice mogu čitati ulaze kao na primjer svijetlo na senzoru, prst na gumbu ili poruku sa društvenih mreža i pretvoriti ih u izlaze poput aktivacije aktuatora, uključivanje LED ili objavljivanja poruke na internetu.

Struktura je jednostavna koja sadržava dva dijela, a to su funkcije `setup()` i `loop()` prikazne na slici Sl. 2.7. Funkcija `setup()` služi za pripremu i poziva se jednom kada se program pokrene, a koristi se za inicijalizaciju nožica i za započinjanje serijske veze. Nakon završetka funkcije `setup()`, `loop()` funkcija se stalno izvršava omogućujući programu da se mijenja, reagira i kontrolira u ovom slučaju, Croduino pločicu.



Sl. 2.7. Prikaz Arudino strukture

## 2.2.2 Croduino pločica

CRODUINO je hrvatska verzija Arduina. Iz razloga što je Arduino objavljen pod licencom otvorenog koda (engl. *Open Source*), sve razlike, izmjene i prilagođene verzije Arduina su dopuštene, pa je tako hrvatska firma e-Radionica napravila prvu hrvatsku Arduino kompatibilnu pločicu gdje sve što se može izvoditi na Arduino, moguće je pokrenuti i na Croduinu prema [9].

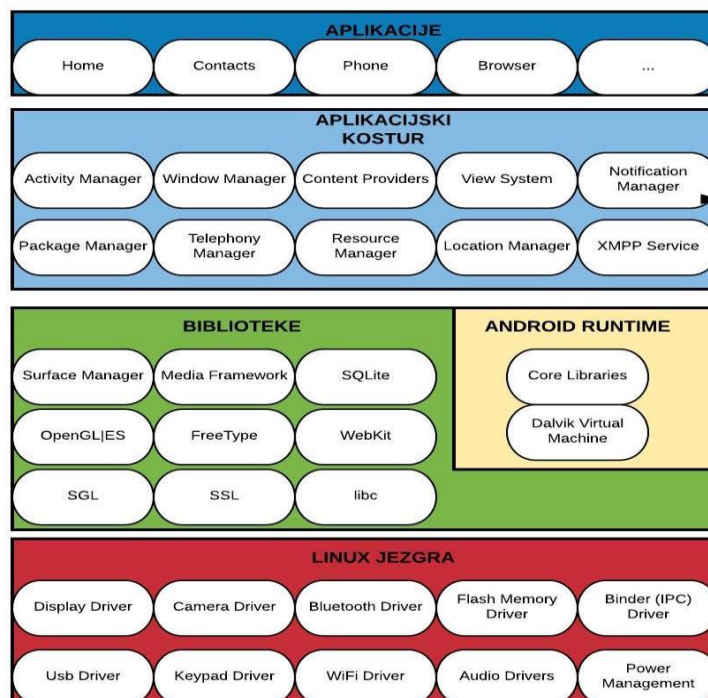
Croduino je alat koji u ispravnoj kombinaciji može kreirati rješenja za brojne primjene jer Croduino može biti interaktivan sa bilo kojom elektroničkom komponentom ili objektom kao na primjer svjetlo, LED, motor, aparat, internet, pametni telefon, te komunicirati sa uređajima preko Bluetootha ili Wi-Fi.

## 2.2.3 Fritzing software

Fritzing jest okruženje otvorenog koda za razvoj programske podrške koja se koristi za dizajniranje sklopovlja. Idealan je za dokumentaciju Arduino sklopovlja te se zbog toga i koristi u diplomskom radu. Ograničen je samo na crtanje projekata, a ne i simulacije. Izvorni kod za Fritzing je pisan u programskom jeziku C++ prema [10] .

## 2.2.4 Android operacijski sustav i Android Studio

Android su osnovali Andy Rubin, Rich Miner, Nick Sears i Chris White 2003. godine u svrhu razvoja programa za pametne mobilne uređaje prema [11]. Nakon nekoliko godina skrivenog rada, Google je odlučio kupiti Android. Prvo predstavljanje Androida kao operacijskog sustava dogodio se 2007. godine, a prvi komercijalni uređaj u koji je bio ugrađen Android OS bio je T-Mobile G1, proizvođača pametnih telefona HTC. Osnivanjem ovakvog saveza, Android je bačen u trku s ostalim mobilnim platformama na tržištu. Android je sustav otvorenog koda koji je napisan u C i C++ programskom jeziku, te svoje osnove zasniva na Linux jezgri. S obzirom da je Android baziran na otvorenom kodu, aplikacije putem programskog među sloja sadrže mogućnost komuniciranja i pokretanja raznih aplikacija (npr: poruke i pozivi, korištenje prednje i zadnje kamere i slično). Iako su C i C++ programski jezici kao programski okvir operacijskog sustava, većina aplikacija pisana je u Java programskom jeziku koristeći Android set razvojne alate (engl. *Software Development Kit, SDK*) prema [12]. Trenutno, Android ima preko dvije milijarde aktivnih korisnika i više od 4 milijuna aplikacija i taj broj sve više raste. Arhitektura sadrži nekoliko razina. Arhitektura Androida se sastoji od aplikacija i njezinog kostura, biblioteka i linux jezgre prikazane na Sl. 2.8.



Sl.2.8. Prikaz arhitekture Androida[13]

Na dnu arhitekture Androida nalazi se Linux jezgra koja obuhvaća pogonske programe (engl. *Upravljački program*) od kojih je najvažniji program za komunikaciju između procesa (engl. *Inter-process communication*, IPC). IPC komunikacija se koristi za izmjenu podataka između procesa ili niti, te pogonski program za upravljanjem napajanja (engl. *Power Management*). Iznad Linux jezgre su programske biblioteke napisane u programskom jeziku C i C++. Pored njih je sloj Android izvršenja (engl. *Android Runtime*) koji se koristi za pokretanje aplikacija, a sastoji se od knjižica (engl. *Core Libraries*), koje sadrže većinu biblioteka Jave, te Dalvik virtualni stroj (engl. *Dalvik Virtual Machine*, DVM) koja izvodi aplikacije kao proces direktno na Linux jezgri, kroz svoj instance na virtualnom stroju. DVM prebacuje sve Java datoteke u svoj format .dex, jer on ostavlja 50% manji memorijski otisak od Java koda. Aplikacijski okvir (engl. *Application framework*) je sloj koji pruža mnoge usluge aplikacijama iz više razine u obliku Java klasa. Programerima je dozvoljeno korištenje usluga iz aplikacijskog programskog okvira u njihovim aplikacijama. Aplikacije su na vrhu arhitekture Android sustava, a sastoje se od ugrađenih aplikacija koje dolaze s uređajem i postavljenih aplikacija nakon kupovine uređaja.

Android studio je integrirano razvojno okruženje za razvoj mobilnih aplikacija za Android operacijski sustav koji se bazira na JetBrains razvojnom okruženju. Sastoji se od skupa alata za izgradnju aplikacije, testiranja i pokretanja aplikacija.

Android Studio koristi Android virtualni uređaj (engl. *Android virtual device*, AVD) kao alat za pokretanje aplikacija virtualnim uređajima. Pomoću AVD-a kreira se vlastiti virtualni uređaj koji se može prilagoditi potrebama korisnika te stvoriti imitator (engl. *Emulator*) pomoću kojega se mogu testirati aplikacije. Prikaz svakog projekta organiziran je po modulima kako bi omogućio brži pristup izvornim datotekama. Svaki modul posjeduje mape: *manifests* .xml datoteke, *Java* datoteke izvornog koda, *Res* resurse koji nisu vezani za kod poput XML, UI niza (engl. *String*) slike i ostalo što je potrebno da bi se aplikacija ispravno izvršavala i prikazala na zaslonu mobitela.

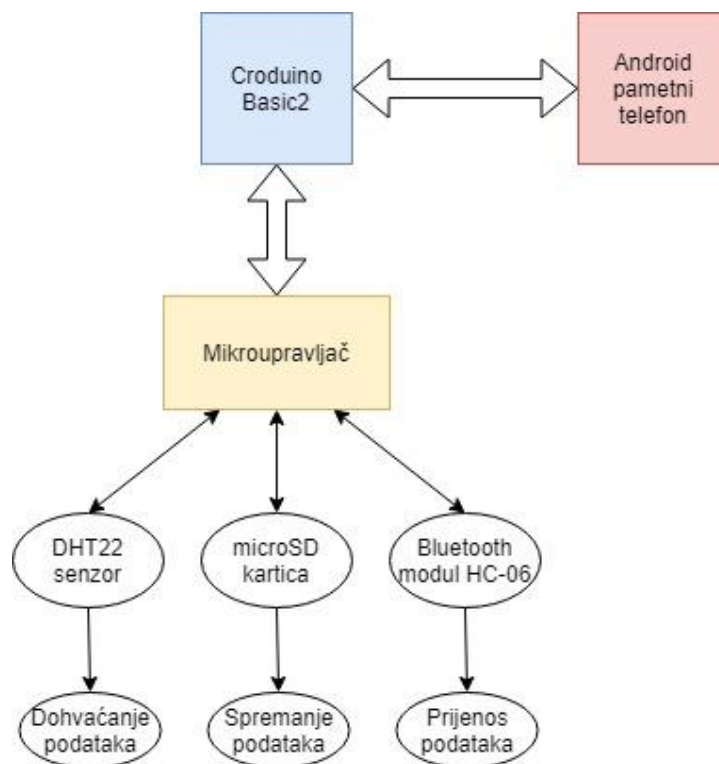
## 2.2.5 PHP

Hypertext Preprocessor ili PHP je skriptni jezik koji se nalazi na strani poslužitelja. Njegova glavna zadaća je razvoj internet stranica, a može se koristiti i kao programski jezik opće namjene. *PHP* je napravio Rasmus Lerdorf 1994. godine izvornog imena osobne početne stranice (engl. *Personal Home Page*). *PHP* kod se može ugraditi unutar *HTML* (engl. *Hyper Text Markup Language*) koda, a može

se iskoristiti i u kombinaciji s raznim internet sustavima ili u sustavima za upravljanjem internetskim sadržajem i internetskim okvirima. *PHP* kod obično obrađuje *PHP* tumač (engl. *PHP Interpreter* ) implementiran kao modul na poslužitelju. Poslužitelj kombinira rezultate tumačenog i izvršenog *PHP* koda, što može biti bilo koja vrsta podataka, uključujući slike s generiranom internet stranicom. *PHP* kod se također može izvršiti pomoću sučelja naredbenog retka ili se može koristiti za implementaciju samostalnih grafičkih aplikacija.

### 3. OSTVARENJE SUSTAVA

Diplomski rad se sastoji od 2 sustava koji su prikazani na slici Sl. 3.1, koji međusobno komuniciraju pomoću Bluetooth komunikacije. Prvi sustav se sastoji od DHT22 senzora temperature i vlage, Bluetooth modula HC-06 i MicroSD to SD pretvarača (engl. *adapter*) koji su spojeni na Croduino Basic2 pločicu, a drugi sustav je Android aplikacija koja pomoću gumba kontrolira sustav Croduino Basic2, te dobivene rezultate ispisuje unutar aplikacije i prikazuje idealne uvjete očuvanja gitare. Svi podaci se spremaju na MicroSD karticu koja služi kao baza podataka očitavanja senzora.



Sl.3.1. Prikaz komunikacije između dva sustava



U tablici Tab. 3.1 prikazane su vrijednosti koje su idealne i štetne za glazbeni instrument gitaru.

Tab. 3. 1. Tablica idealne vremenske prilike za gitaru

	Hladno	Idealno	Vruće
Temperatura	< 20°C	21°C-24°C	>24°C
Vlaga	< 40%	40%-50%	>50%

### 3.1 Arduino programsko rješenje

Kako bi se očitavale vrijednosti sa DHT22 senzora u programskom okruženju Arduino IDE, vrši se inicijalizacija senzora na način da se pridodaje analogna nožica i tip senzora, te se uključuje dodatna biblioteka unutar Arduino IDE okruženja koja dolazi unutar Adafruit upravljačkih programa za senzore (engl. *Adafruit sensor driver*). U određene varijable (h,t,i) spremaju se informacije o temperaturi i vlazi pomoću funkcija `.readHumidity()` i `.readTemperature()` u okolišu u kojem se senzor nalazi te se izračunava pravi osjećaj (engl. *Real feel*) pomoću funkcije `.computeHeatIndex()` prikazane na slici Sl. 3.1. Ukoliko DHT22 senzor nije adekvatno priključen na Croduino Basic2 pločicu i/ili ako nije ispravno inicijaliziran unutar Arduino IDE, program će izbaciti grešku. Ako su svi uvjeti zadovoljeni program će ispisivati rezultate prikazane na slici Sl.3.2. Svako očitavanje ima stanku koja traje dvije sekunde zbog sigurnosti sustava. Funkcija `.computeHeatIndex` parametre temperaturu i vlagu unosi u svoj izračun stvarnog osjećaja, te ga ispisuje u °C zbog naredbe `false`, dok je `true` za °F.

```
#include "DHT.h"
int pinDHT = A0;
int tipDHT = DHT22;
DHT dht(pinDHT, tipDHT);

void setup() {
  ...
  dht.begin();
  ...
}

void loop() {
  ...
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float i = dht.computeHeatIndex(t, h, false);
  delay (2000);
  ...}
```

Sl.3. 2. Inicijalizacija i rad DHT22

Nakon što se uspješno provede programski kod u Arduino IDE okruženju, ispis se izvršava pomoću funkcije *serial.print()* koja prikazuje rezultate kao što su na slici Sl. 3.3.

```
Temperature: 21.30 *C Humidity: 49.70 % Heat index: 20.78 *C  
Temperature: 21.40 *C Humidity: 48.90 % Heat index: 20.87 *C  
Temperature: 21.30 *C Humidity: 48.70 % Heat index: 20.76 *C
```

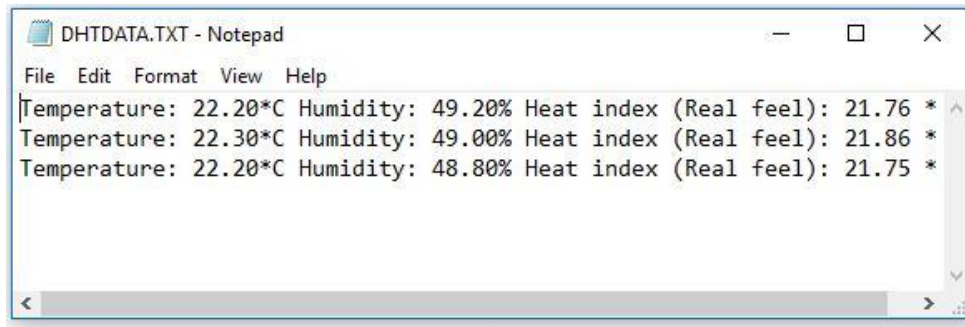
### Sl. 3.3. Prikaz temperature i vlage na ekran

Kako bi korisnik aplikacije imao pristup i pregled svih očitavanja sa senzora, omogućena je baza podataka u obliku tekstualne datoteke na microSD kartici. Nakon svakog očitavanja sa senzora, Croduino Basic2 pločica zapisuje informacije u datoteku pod nazivom *dhtData.txt*. Datoteka se sastoji od ispisa temperature, vlage i izračunatog stvarnog osjećaja u okolišu. Inicijalizacija microSD kartice unutar Arduino IDE započinje sa uključivanjem biblioteke *SPI* (engl. *Serial Peripheral Interface*) prema slici Sl. 3.4. kojeg koristi mikroupravljač za komunikaciju s jednim ili više perifernih uređaja na kratkim udaljenostima, u ovom slučaju sa microSD karticom. Pomoću nožice na kojoj je spojen *MISO*, microSD kartica šalje podatke prema Croduino Basic2, a pomoću *MOSI* nožice mikroupravljač upisuje podatke u datoteku na microSD kartici. Osim *SPI.h*, potrebna je i *SD.h* biblioteka koja dopušta čitanje i pisanje po microSD kartici.

```
#include <SD.h>  
#include <SPI.h>  
File myFile;  
void setup() {  
  ...  
  SD.begin(4); //SS Pin  
  ...  
}  
void loop() {  
  ...  
  //senzor  
  ...  
}  
myFile = SD.open("dhtData.txt", FILE_WRITE);  
if(myFile){  
  ...  
  myFile.print(t);  
  myFile.print(h);  
  myFile.print(i);  
  ...  
}
```

### Sl. 3.4. Inicijalizacija microSD kartice

Pročitane vrijednosti iz okoline spremaju se u tekstualnu datoteku prikazanu na slici Sl.3.5.



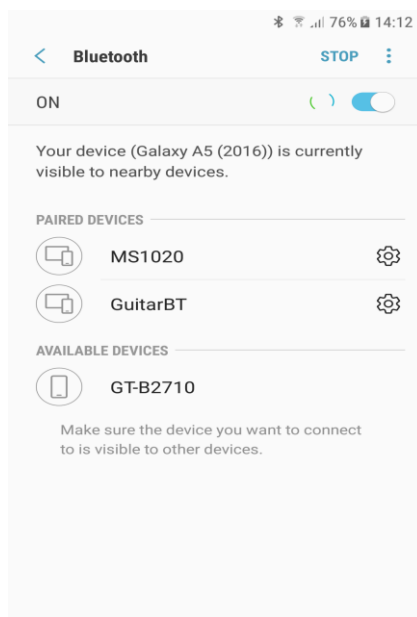
Sl. 3.5. Zapis unutar tekstualne datoteke

Osim rada DHT22 senzora i microSD kartice, podatci se šalju putem Bluetooth komunikacije (Bluetooth modul HC-06) na Android pametni telefon. Inicijalizacija unutar Arduino IDE vrši se na način da se dodjeljuje dvije digitalne nožice (u ovom slučaju 2 i 3) za serijsku komunikaciju uz biblioteku *SoftwareSerial.h* prema slici Sl. 3.6, te se na funkciju *bt.begin*, Bluetooth pokreće izvršavanje komunikacijskog protokola.

```
#include <SoftwareSerial.h>
SoftwareSerial bt(2,3);
...
void setup() {
  ...
  bt.begin(9600);
  ...
}
void loop() {
  ...
  if(bt.available()){
    bt.print((String) readSensor(c));
    ...
  }
  ...
float readSensor (char _c) {
  // temperatura, vlaga, Real Feel
}
```

Sl.3. 6. Rad sa Bluetooth HC-06

Kako bi se komunikacija u oba smjera izvršavala moraju se Android pametni telefon i Bluetooth modul HC-06 (GuitarBT) upariti. Sparivanje se odrađuje ugrađenim programskim kontrolama Android operacijskog sustava prikazanog na slici Sl.3.7.



**Sl. 3.7. Sparivanje mobilnog telefona na GuitarBT**

Nakon sparivanja pomoću Android operacijskog sustava ili korištenjem aplikacije na pametnom telefonu, uređaji su spremni za komunikaciju kao što se je prikazano na Sl.3.8. LED, koja se nalazi na Bluetooth modulu HC-06, prestaje sa treptanjem te konstantno svijetli što označava uspostavu veze. Nakon što Android pametni telefon pošalje paket zahtjeva očitavanja temperature i vlage u okolišu putem Bluetooth komunikacije, sustav uključuje senzor, iščitava podatke, sprema ih na microSD karticu i odgovara podacima na Android pametni telefon.

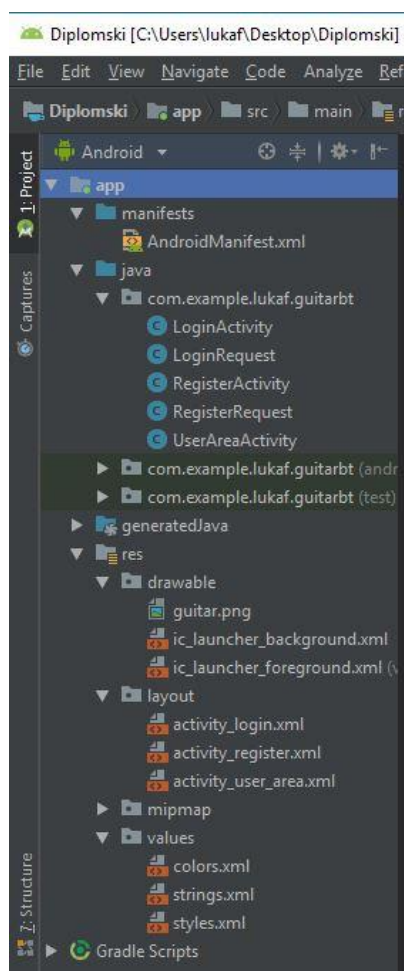


**Sl.3.8. HC-06 modul nakon sparivanja**

## 3.2 Android aplikacija i baza podatka

Android studio je integrirano razvojno okruženje za razvoj mobilnih aplikacija za Android operacijski sustav koji se bazira na JetBrains razvojnom okruženju. Sastoji se od skupa alata za izgradnju aplikacije, testiranja i pokretanja aplikacija. Svaki modul posjeduje mape: *manifests* .xml datoteke, *Java* datoteke izvornog koda, *Res* resurse koji nisu vezani za kod poput XML, UI niza (engl. *String*) slike i drugo što je potrebno da bi se aplikacija ispravno izvršavala i prikazala na zaslonu mobitela kao što je prikazano na slici Sl.3.9.

*Java* datoteke izvornog koda potrebne su za svaki prozor aplikacije GuitarBT. Datoteke *LoginActivity* i *RegisterActivity* prikazuju formu prijave i registracije korisnika aplikacije, a dok *UserAreaActivity* upravlja Croduino sustav pomoću gumba, te ispisuje podatke u tekstualnom dijelu aplikacije koje je Croduino sustav pročitao iz okoliša.



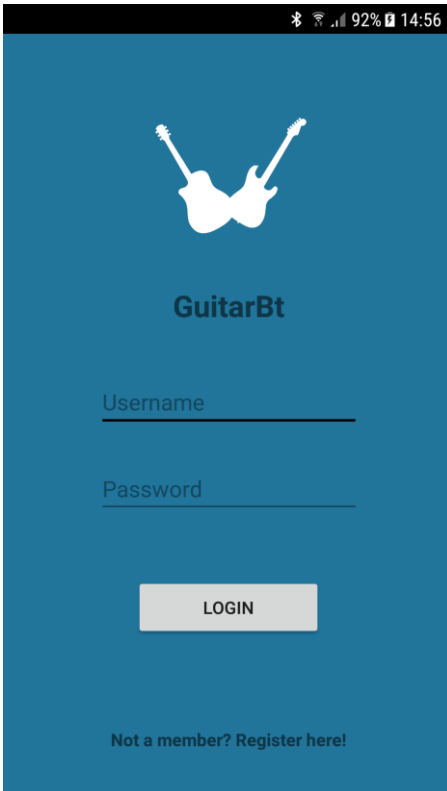
Sl.3.9. Prikaz strukture Android projekt

U Android projektu, u manifest datoteci programirana je dozvola za Bluetooth i internet komunikaciju koja je prikazana na slici Sl. 3.10. Traži od korisnika dopuštenje prilikom ulaznja u aplikaciju, kako bi se Bluetooth i internet pokrenuo i kako bi se ostvarila komunikacija između Android pametnog telefona s Croduino Bluetooth modulom i bazi podataka na internetu .

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.BLUETOOTH" />
```

**Sl. 3.10. Dozvole za pristup internetu i Bluetooth-u**

Nakon uspostavljanja dozvola pri ulasku u aplikaciju, stvara se početna stranica prikazana na slici Sl. 3.11, koja sadrži formu prijave i registracije (engl. *Login/Register form*) korisnika.



92% 14:56

**GuitarBt**

Username

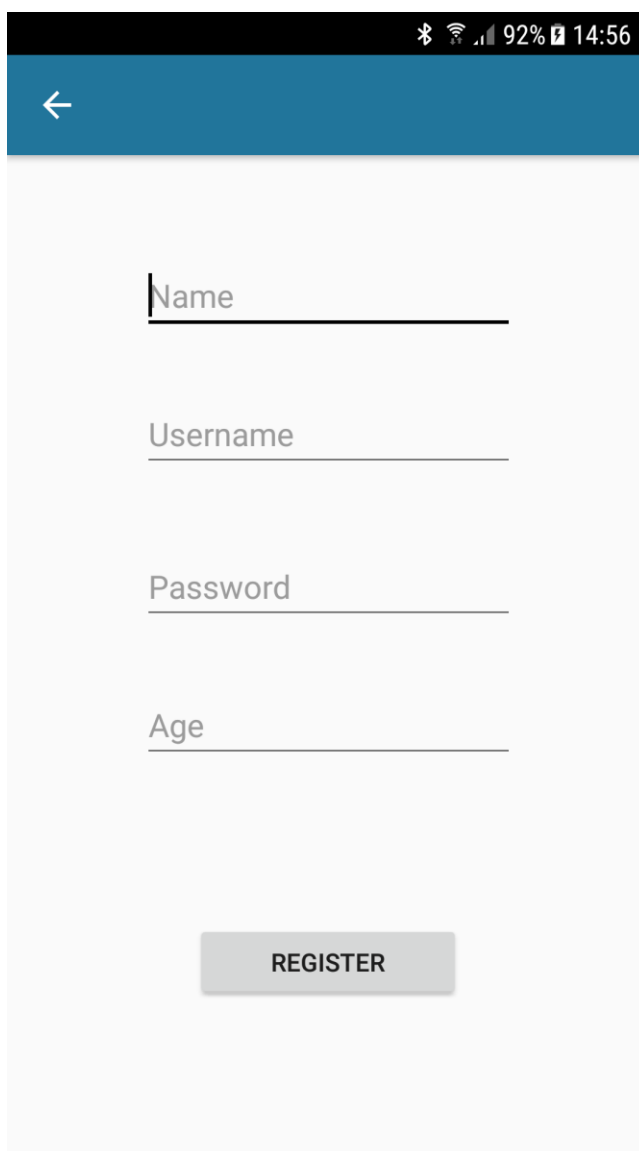
Password

LOGIN

Not a member? Register here!

**Sl.3.11. Početna stranica aplikacije**

Ukoliko korisnik ne posjeduje račun (engl. *Account*) u aplikaciji pritiskom na tekst „Register here!“ otvara formu u kojoj upisuje potrebne podatke za otvaranje računa prema slici Sl. 3.12. Ne postoji dva korisnika aplikacije koji mogu koristiti isto ime (engl. *Username*) stoga, ukoliko se dogodi takav događaj, aplikacija će javiti pogrešku. Podaci za otvaranje računa su: Ime, korisničko ime, korisnička lozinka i godine. Pritiskom gumba „Register“ stvara se račun koji se sprema na internet bazi podataka.

The image shows a mobile application interface for user registration. At the top, there is a black status bar with icons for Bluetooth, Wi-Fi, cellular signal, 92% battery, and the time 14:56. Below this is a blue header bar with a white back arrow icon. The main content area is light gray and contains four text input fields, each with a label above it: "Name", "Username", "Password", and "Age". The "Name" field has a cursor at the end. At the bottom of the form is a gray button with the text "REGISTER" in black capital letters.

Sl. 3.12. Forma za otvaranje računa

Formu prijave i registracije potrebno je povezati sa bazom podataka na internetu. Kako bi Android aplikacija postigla komunikaciju s mrežnom bazom podataka potreban je *API* (engl. *Application Programming Interface*) koji je napisan u *PHP* skriptnom jeziku. *PHP* je popularan skriptni jezik koji je namijenjen za izradu dinamičkih internetskih stranica, baza podataka i aplikacija. Koristi se za izvršavanje skripti na poslužiteljskoj strani (engl. *Server-side scripting*), a moguće ga je i koristiti s komandnog linijskog sučelja.

Zbog dvije korištene forme unutar aplikacije (prijava/registracija) potrebne su dvije *PHP* datoteke, koje su nazvane *Login.php* i *Register.php*. Gotove *PHP* datoteke se predaju mrežnoj bazi podataka (*000webhost.com*) koja će kasnije komunicirati s Android aplikacijom.

U datoteci *Register.php* prikazana na slici Sl. 3.13. nalaze se dvije ključne stvari. Prva je spajanje na internet bazu podataka sa potrebnim parametrima (osobna internet stranica, korisničko ime, lozinka baze podataka i ime baze podataka) a druga je da sadrži četiri parametra (ime, godine, korisničko ime, korisnička lozinka). Kada Android aplikacija pozove datoteku *Register.php*, ona će mu predati četiri parametra koje će korisnik aplikacije popuniti unutar forme registracije. Prilikom pritiska gumba „Register“ , *Register.php* preuzima parametre i zapisuje ih u internet bazu podataka i na taj način registrira korisnika aplikacije, a u slučaju da se ponove isti parametri vraća pogrešku.

```
<?php
    $con = mysqli_connect("██████████", "██████████",
    "██████████", "██████████");

    $name = $_POST["name"];
    $age = $_POST["age"];
    $username = $_POST["username"];
    $password = $_POST["password"];

    $statement = mysqli_prepare($con, "INSERT INTO user (name,
    username, age, password) VALUES (?, ?, ?, ?)");
    ...
```

Sl. 3.13. Dio programskog koda datoteke *register.php*



Unutar GuitarBT aplikacije, potrebno je napraviti *Java* datoteku koja će tražiti zahtjev za *register.php*. Postavlja se internet stranica datoteke *register.php* koja se nalazi na mrežnoj bazi podataka (*000.webhost.com*) prikazano na slici Sl. 3.14.

```
private static final String REGISTER_REQUEST_URL = "https://lflguitarbt.000webhostapp.com/Register.php";
private Map<String,String> params;

public RegisterRequest(String name,String username, int age, String password,Response.Listener<String> listener){
    super(Method.POST, REGISTER_REQUEST_URL,listener, errorlistener: null);
    params = new HashMap<>();
    params.put("name",name);
    params.put("age",age+"");
    params.put("username",username);
    params.put("password",password);
}
```

Sl. 3.14. Zahtjev za datoteku register.php

Nakon što korisnik ispuni formu registracije unutar GuitarBT aplikacije i klikne na gumb „Register“, stvara se korisnički račun na bazi podataka koja sadrži njegovo ime, korisničko ime, godine i korisničku lozinku prikazanu na slici Sl. 3.15.

+ Options					user_id	name	username	age	password			
					31	luka	lukaflo	25	123			
		Edit		Copy		Delete	46	marko	marko123	25	123456	
		Edit		Copy		Delete	47	Petar	petar321	35	petar321	
		Check all	With selected:			Edit		Copy		Delete		Export

Sl. 3.15. Baza korisnika aplikacije GuitarBT

U datoteci *Login.php* se nalaze dva parametra (korisničko ime i korisnička lozinka) prema slici Sl. 3.16, koje *Login.php* uspoređuje sa već kreiranom bazom podataka i njezinim parametrima. Ukoliko se parametri podudaraju, korisnik je uspješno pristupio Android aplikaciji, a u protivnom izbacuje grešku i ponovno otvara formu prijave.

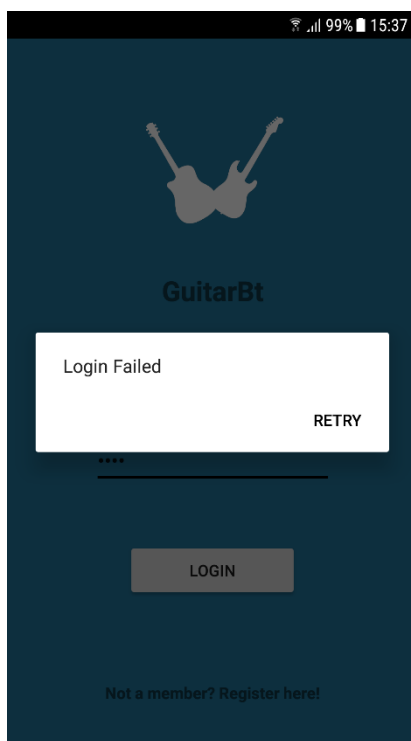
```
<?php
$con = mysqli_connect("██████████", "██████████",
"██████████", "██████████");

$username = $_POST["username"];
$password = $_POST["password"];

$stmt = mysqli_prepare($con, "SELECT * FROM user WHERE
username = ? AND password = ?");
...|
```

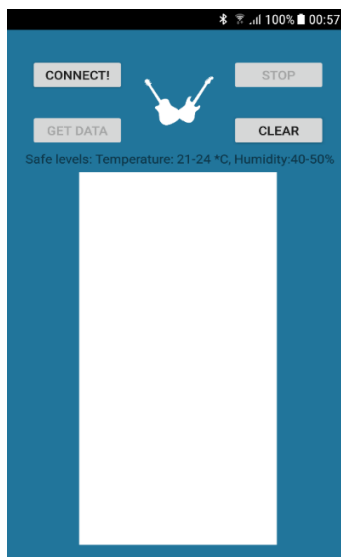
**Sl. 3.16. Dio programskog koda datoteke Login.php**

Nakon uspješnog otvaranja računa, aplikacija ponovno usmjerava korisnika na početnu stranicu na kojoj je forma prijave. Ukoliko korisnik unese krive podatke u formu, otvara se novi prozor upozorenja koji ga usmjerava na ponovni pokušaj prijave prikazan na slici Sl. 3.17.



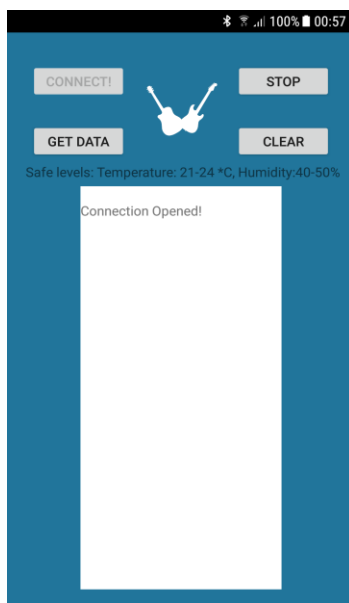
**Sl. 3.17. Neuspješna prijava korisnika**

Ako korisnik unese sve podatke točno, otvara mu se sučelje aplikacije GuitarBT koje obavlja interakciju s Croduino sustavom preko pritiskom gumba. Aplikacija se sastoji od četiri gumba i tekstualnog dijela prikazanog na slici Sl. 3.18, u kojemu se ispisuju podatci sa Croduino sustava.



**Sl. 3.18. Prikaz sučelja nakon prijave korisnika**

Gumb „Connect!“ služi da se aplikacija GuitarBT poveže sa Croduino sustavom pomoću adrese uređaja i serijskog porta, te ukoliko je povezivanje uspostavljeno da se u tekstualnom dijelu ispiše potvrдна poruka prikazana na slici Sl. 3.19.



**Sl. 3.19. Povezanost s Croduino Basic2 pomoću Bluetooth modula**

Kako bi se Android pametni telefon upario s Croduino sustavom, potrebna mu je adresa uređaja Bluetooth modula HC-06 i serijskog porta. Programski kod prikazan na slici Sl. 3.20. uspostavlja komunikaciju između dva sustava koja se koriste u diplomskom radu.

```
private final String DEVICE_ADDRESS="00:21:13:01:5A:E3";  
private final UUID PORT_UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB")
```

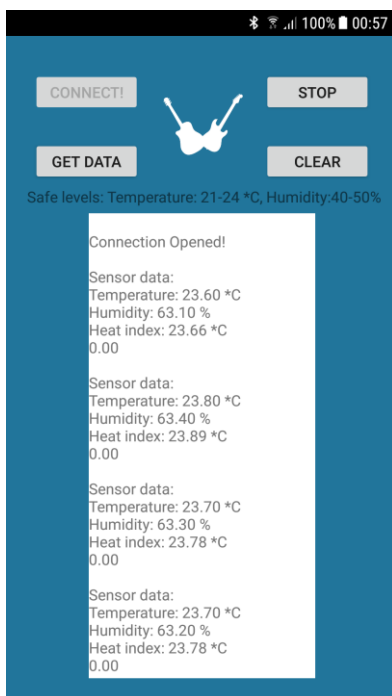
Sl. 3.20. Spajanje pomoću adrese i serijskog porta

Nakon uspostave komunikacije između dva sustava, Android aplikacija pokreće funkciju *beginListenForData()* prikazana na slici Sl. 3.21, koja cijeli ulazni tok preusmjerava na tekstualni dio aplikacije, gdje ga i ispisuje.

```
int byteCount = inputStream.available();  
if(byteCount > 0)  
{  
    byte[] rawBytes = new byte[byteCount];  
    inputStream.read(rawBytes);  
    final String string=new String(rawBytes, charsetName: "UTF-8");  
    handler.post(new Runnable() {  
        public void run()  
        {  
            textView.append(string);  
        }  
    });  
}
```

Sl.3.21. Ulazni tok na aplikaciju

Pristikom na gumb za dohvaćanje podataka (engl. *GET DATA*) Android pametni telefon šalje paket preko Bluetooth komunikacije Croduinu Basic2 koji odgovara s podacima na senzoru te ih sprema na microSD karticu, a Android pametni telefon ih ispisuje u tekstualnom dijelu aplikacije koje je prikazano na slici Sl. 2.22.



Sl. 3.22. Prikaz podataka sa senzora

Dohvaćanje podataka ima kašnjenje od 2 sekunde zbog sigurnosti sustava. Ispis „0.00“ označava da je sustav spreman za novi zahtjev, te je korisnik u mogućnosti ponovno pritisnuti gumb „dohvati podatke“ (engl. *GET DATA*) kojemu je programski kod prikazan na slici Sl. 3.23.

```
public void onClickSend(View view) {
    String string = new String( original: "1");
    string.concat("\n");
    try {
        outputStream.write(string.getBytes());
    } catch (IOException e) {
        e.printStackTrace();
    }
    textView.append("\nSensor data: "+" \n");
}
```

Sl.3. 23. Dio Programskog koda za gumb GET DATA

Tekstualni dio aplikacije je izrađen da se može pomicati i iščitavati do trenutka kada se microSD kartica napuni podacima. Pritiskom na gumb za čišćenje tekstualnog dijela aplikacije „Clear“, brišu se podatci koji su ispisani u tekstualnom dijelu i na microSD kartici.

Ukoliko korisnik želi prekinuti komunikaciju s Croduino sustavom, pritiskom na gumb „STOP“, zatvara se komunikacija prema Croduino Basic2 sustavu, brišu se podatci te prestaje slušanje podataka ostvareno pomoću funkcije (*beginListenerForData*) koje je prikazano na slici Sl. 3.24.

```
public void onClickStop(View view) throws IOException {  
    stopThread = true;  
    outputStream.close();  
    inputStream.close();  
    socket.close();  
    setUiEnabled(false);  
    deviceConnected=false;  
    textView.append("\nConnection Closed!\n");  
}
```

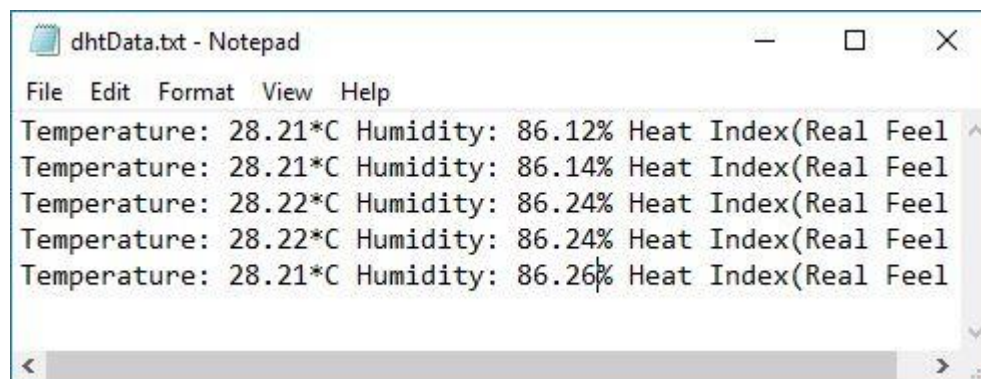
Sl.3. 24. Dio programskog koda za gumb STOP

## 4. TESTIRANJE SUSTAVA

Testiranje sustava se održavalo u mjesecu srpnju i u listopadu. Sustav je prikupljao podatke i spremao na microSD karticu dok se nije dogodilo zasićenje u memoriji. Zbog velikog broja informacija prikazivati će se datoteka *dhtData.txt*. U prvom testu, sustav se nalazio izvan građevinske ustanove pokriven limenim pokrivačem kako bi se zaštitio od pada oborina, dok je drugi test napravljen unutar stambene zgrade.

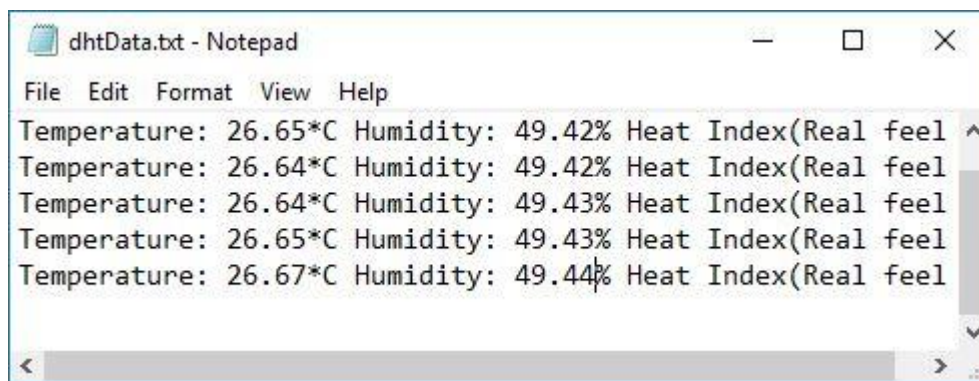
U petak, 28.07.2018 godine u Republici Hrvatskoj u gradu Osijeku palo je 48 litara kiše na kvadratni metar. Temperatura zraka je bila između 27 °C do 31 °C.

Dobivena mjerenja koja su prikazana na slici Sl. 4.1, nisu pogodna za ostavljanje gitare na specifičnom mjestu jer se gitara ostavlja na mjestu gdje je temperatura zraka između 21 °C i 24 °C i relativni postotak vlage između 40% i 50%.



Sl. 4.1. Prikaz očitavanja na dan 28.07.2018

U nedjelju, 07.10.2018 godine u gradu Osijeku temperature su se penjale do 25 °C preko dana, a temperatura preko noći pala je na 13 °C. Dobivene vrijednosti temperature unutar radne sobe su prikazane na slici Sl. 4.2.



#### Sl.4. 2. prikaz očitavanja na dan 07.10.2018

Iako je mnogo bolje očitavanje, gitaru u ovakvoj radnoj sobi nije preporučeno ostaviti jer ne zadovoljava točne uvjete.



## 5. ZAKLJUČAK

U ovom radu opisana je opasnost od propadanja glazbenih instrumenata, te je napravljena aplikacija koja brine o gitarama. Gitare, kao vrlo osjetljivi glazbeni instrument, moraju imati posebne uvjete kako bi se što više očuvale. Drvo i žice gitare su osjetljivi na temperaturu, pa s toga temperatura u kojoj se nalazi bi trebale varirati između 21 °C i 24 °C. Osim temperature, vlaga je posebna prijetnja za drvo. Ukoliko je zrak presuh dolazi do puknuća drveta na određenim mjestima te se tako gitara uništava. Vlaga zraka bi trebala varirati između 40 % do 50 % očitavanja sa senzora.

Realiziran je sustav koji očitava vrijednosti iz okoliša te ih prosljeđuje na pametni telefon. Opisane su sve sklopovske komponente korištene u radu koje su u suradnji s mobilnom aplikacijom stvorile jednu funkcionalnu cjelinu. Kao nadogradnja ove aplikacije, korisnik bi trebao imati na svojem korisničkom računu izbor između više sustava na koji se može spojiti. Na taj način mogao bi dodati više instrumenata koje posjeduje te ih nadzirati kako se ne bi dogodile velike štete.

## 6. LITERATURA

- [1] E-radionica, Tutorijali, KKM DHT22, 2015. Posjećeno 15.02.2018 na mrežnoj stranici:  
<https://e-radionica.com/hr/blog/2015/10/18/kkm-dht22/>
- [2] E-radionica, Tutorijali, Croduino Basic2, 2016. Posjećeno 17.02.2018 na mrežnoj stranici:  
<https://e-radionica.com/hr/blog/2015/10/11/novi-croduino-je-tu-croduino-basic2/>
- [3] E-radionica, shop, Croduino Basic2, description. Posjećeno 15.03.2018 na mrežnoj stranici:  
<https://e-radionica.com/en/croduino-basic2.html>
- [4] Scientific American, a Division of Springer Nature America, *How does Bluetooth work?*, 2007. Posjećeno 23.06.2018 na mrežnoj stranici:  
<https://www.scientificamerican.com/article/experts-how-does-bluetooth-work/>
- [5] Guangzhou HC Information Technology Co, *HC-06 data sheet*, 2011. Posjećeno 3.3.2018 na mrežnoj stranici:  
<https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>
- [6] SanDisk microSD, *OEM product Manual*, 2010. Posjećeno 27.03.2018 na mrežnoj stranici:  
<https://www.alliedelec.com/m/d/04db416b291011446889dbd6129e2644.pdf>
- [7] Adafruit Learning system, *microSD card breakout board tutorial*, 2018. Posjećeno 5.04.2018 na mrežnoj stranici:  
<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-micro-sd-breakout-board-card-tutorial.pdf>
- [8] Aduino, *getting started – introduction*, 2018. Posjećeno 22.05.2018 na mrežnoj stranici:  
<https://www.arduino.cc/en/Guide/Introduction>

- [9] E-radionica, *što je Arduino a što croduino?* 2017. Posjećeno 20.02.2018 na mrežnoj stranici:  
<https://e-radionica.com/hr/blog/2015/10/08/sto-je-arduino-i-croduino/>
- [10] West Florida Components, *all about fritzing*. Posjećeno 20.06.2018 na mrežnoj stranici:  
<https://www.westfloridacomponents.com/blog/all-about-fritzing/>
- [11] Wikipedia, Android (operating system). Posjećeno 21.06.2018 na mrežnoj stranici:  
[https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [12] AndroidPIT, *What is Android?*, 2016. Posjećeno 22.06.2018 na mrežnoj stranici:  
<https://www.androidpit.com/what-is-android>
- [13] Wikipedia, Android (operating system). Posjećeno 25.06.2018 na mrežnoj stranici:  
[https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)#/media/File:Android-System-Architecture.svg](https://en.wikipedia.org/wiki/Android_(operating_system)#/media/File:Android-System-Architecture.svg)

## POPIS POJMOVA

**UART** – Univerzalni asinkroni prijemnik – odašiljač (engl. *Universal Asynchronous Receiver – Transmitter*) je računalni sklopovski uređaj za asinkrono serijsko komuniciranje u kojem su format podatka i brzine mogući za konfiguraciju. UART je obično pojedinačni dio integriranog kruga koji se koristi za komunikacije preko računala ili priključka periferne jedinice. UART komunikacija između dva uređaja se izvršava na način da se podatci prenose u bitovima koji su posloženi u slijed, a na drugoj strani (odredište) UART sve bitove prebacuje u bajtove. Prilikom komunikacije UART koristi registar pomaka koji je temelj konverzije paralelnih i serijskih formi.

Sve operacije UART sklopovlja kontrolirane su internim taktim signalom koji radi na brzini između 8 ili 16 puta brzine prijenosa. Prijemnik provjerava stanje dolaznog signala na svakom impulsu sata, tražeći početak bita. Ukoliko počeni bit traje najamnije polovicu vremena ukupnog bita smatra se valjanim i signalizira se početak novog znaka.

Operacija prijenosa je jednostavnija jer vrijeme ne mora biti određeno, niti je vezana za bilo koji fiksni vremenski interval. Čim sustav slanja unese znak u registar, UART generira početni bit, pomiče potrebni broj bitova podataka na liniju, generira i šalje paritetni bit i šalje zaustavne bitove.

**USB** – Univerzalna serijska sabirnica (engl. *Universal Serial Bus*) je port unutar uređaja koji se koristi za prijenos podataka ili komunikaciju između više spojenih uređaja. Do pojave USB-a koristile su se mnoga sučelja (serijska i paralelna) koja su bila komplicirana na svoje načine, te je prijenos podataka bio uvelike sporiji nego USB. Zvezdasta stuktura se stvara jer USB funkcionira kao jedan poslužitelj (engl. *host*) na kojem su spojene više jedinica. Svaki USB port može imati više razina grananja, te se po svakom *host*-u može priključiti do 127 istih ili različitih uređaja.

USB se kroz vrijeme unapređivao te je danas dostupna inačica 3.0. Inačica 1.0 je omogućavala brzinu od 1.5 Mbit/s koja nije omogućavala kabele za produživanje. Nakon nekoliko godina stvorila se inačica 2.0 koja se zavladała svijetom sve do 2008 godine kada ju je inačica 3.0 zamjenila. Inačica 2.0 je omogućavala brzinu prijenosa podataka do 480 Mbit/s, a dok 3.0 omogućuje brzinu prijenosa do 3.2 Gbit/s.

**Bluetooth** – tehnologija komunikacije između dva ili više uređaja koja mogu razmjenjivati ili prenositi podatke. Kako bi se veza između dva uređaja (ili više) uspostavila, šalju se radio valovi u frekvencijskom području između 2.4 GHz i 2.48 GHz. Bluetooth trenutno stoji na inačici 5.0 koja je veoma brza za bežični prijenos podataka. Osim inačice, postoje i klase Bluetooth komunikacije. Klase se temelje na udaljenostima između uređaja koji su upareni. Klasa koja može postići najveću udaljenost između dva uparena uređaja je prva klasa i ona je u mogućnosti da unutar 100 metara može uspostaviti komunikaciju između uređaja.

**RS-232** – je komunikacijski standard napravljen 1960. godine. Dugi niz godina se koristio kao komunikacija između dva uređaja, sve dok ga nije zamjenio USB. Komunikacija se izvršavala pomoću kabela (muško-ženski kabel) koji posjeduje devet nožica. Iz razloga što je duljina i prijenosna linija kabela bila drugačija za svaki uređaj, ne može se točno utvrditi njegova brzina prijenosa, no uzima se standard za koje vrijedi da prenosi podatke 20 Kbps. Njegove naponske razine imaju raspon između -12V do +12V, gdje vrijedi da napon od 12V predstavlja binarno 0, a napon od -12V predstavlja binarno 1.

**Infrared communication, IR** - jeftina i jednostavna bežična komunikacijska tehnologija. Infracrvena svjetlost je vrlo slična vidljivoj svjetlosti, osim što ima malo dužu valnu duljinu, što znači da nije vidljiva ljudskom oku. Jednostavni primjer infracrvene komunikacije je daljinski upravljač. Infracrveno svjetlo se više puta uključuje i isključuje, 38 000 puta u sekundi prema senzoru na televiziji.

**Upravljački program** – kako bi ispravno funkcioniralo sklopovlje i operacijski sustav računala, mora postojati međusloj koji obavlja komunikaciju između njih. Upravljački programi se pišu specifično za određene uređaje i operacijske sustave. Ukoliko operacijski sustav želi postići komunikaciju sa sklopovljem, ono prvo šalje naredbe upravljačkim programima koji ih obrađuju i prosljeđuju na sklopovlje. Prilikom kupnje sklopovlja, upravljački programi se uglavnom dobivaju na CD-u, te se pokreću nakon što se određeni uređaj spoji na računalo. Noviji operacijski sustavi poput Windows 8.1 ili Windows 10 nedugo nakon spajanja, tada nepoznatog uređaja na računalo, uz pretpostavku da je računalo spojeno na internet, pronalazi upravljačke programe na internetu i postavlja u pogon uređaj, te mu daje potpunu funkcionalnost.

**DVM** – (engl. *Dalvik Virtual Machine*) procesni virtualni stroj na Android operativnom sustavu koji izvršava aplikacije napisane za Android. DVM je bio sastavni dio Androida 4.4 pod nazivom „KitKat“ i ostalim ranijim verzijama koji su se upotrebljavali na mobilnim uređajima kao što su mobilni telefoni, tablet računala i ponekih pametnih televizora. Otvorenog je koda kojeg je izvorno napisao Dan Bornstein a ime je dobio po ribarskom mjestu Dalvik u Islandu.

**Aplikacijski okvir**- (engl. *Application framework*) sastoji se od programskog okvira kojeg koriste ravojni programeri za implementaciju standardne strukture aplikacija. Postali su popularni pri usponu grafičkih sučelja (engl. *Graphical User Interfaces*, GUI). Pri kreiranju okvira obično se koristi objektno orijentirano programiranje (engl. *Object. Oriented programming*).

**Android studio** - integrirano razvojno okruženje za razvoj mobilnih aplikacija za Android operacijski sustav koji se bazira na JetBrains razvojnom okruženju. Sastoji se od skupa alata za izgradnju aplikacije, testiranja i pokretanja aplikacija.

**Android virtualni uređaj** – (engl. *Android virtual device*, AVD) - alat za pokretanje aplikacija virtualnim uređajima.

**Emulator** – Android studio alat koji simulira rad Android uređaja na osobnom računalu u svrhu testiranja, bez potrebe za fizičkim uređajem. Emulator pruža gotovo sve mogućnosti stvarnog Android uređaja. Moguće je simulirati dolazne telefonske pozive ili tekst poruke, simulirati različite brzine mreže, rotacije i ostale senzore sklopovlja, pristupiti trgovini, no najčešća stvar je testiranje aplikacija koje programeri stvaraju.

**phpMyAdmin** – Besplatni programski alat napisan u PHP-u, namjenjen za upravljanje MySQL-a preko interneta koji podržava široki raspon operacija. Često korištene operacije (upravljanje bazama podataka, tablica, stupaca, odnosa, indeksa, korisnika, dozvola i dr.) mogu se obavljati putem korisničkog sučelja.

**SDA** – (engl. *Serial Data Line*) serijska podatkovna linija unutar komunikacije između mikroupravljača i procesora.

**PWM** – (engl. *Pulse-Width Modulation*) – pulsno širinska modulacija je način dobivanja analognog signala iz digitalnog. To se postiže tako da digitalnom signalu fiksne frekvencije mijenjamo faktor popunjenosti. Dakle, mijenja se omjer između vremena uključenosti i isključenosti. Ako je trajanje

jednog ciklusa manje od vremenske konstante nekog sustava sa inercijskom dinamikom (npr. RC krug, zalet elektromotora), na izlazu će se dobiti neka vrijednost između dvije krajnosti digitalnog signala. Osim toga, *PWM* je jedan od dva glavna algoritma koji se koriste u fotonaponskim punjačima.

**LED** – (engl. *Light Emitting Diode*) svjetleća dioda koju je napravio Oleg Vladimirovich Losev između 1920 i 1930 godine. Dioda pretvara elektronički signal u svjetlost različitih boja ovisno u poluvodiču. Kako bi ispravno radile koriste različite otpornike.

**Vcc** – (engl. *Voltage, common collector*) – Pozitivno nabijen napon integriranog električnog kruga.

**GND** – (engl. *Ground*) - uzemljenje

**TXD** – (engl. *Bluetooth serial signal sending PIN*) – Serijski izlaz modula koji se spaja na mikroupravljač

**RXD** – (engl. *Bluetooth serial signal receiving PIN*) - Serijski ulaz modula koji se spaja na mikroupravljač

**HTML** – (engl. *HyperText Markup Language*) - označni jezik za stvaranje internet stranica i web aplikacija.

**PHP Interpreter** – u računalnoj znanosti, tumač (engl. *Interpreter*) je računalni program koji izravno izvršava tj. izvodi upisane upute napisane na programskom ili skriptnom jeziku, bez potrebe da se prethodno sastavljaju na program strojnog jezika.

## SAŽETAK

U radu je detaljno opisan nastanak mobilne aplikacije koja uz pomoć sklopovskog dijela, prikazuje korisniku u kakvom okolišu se nalazi glazbeni instrument gitara. Dana je teorijska podloga problema mikroklima na gitaru te je u praktičnom dijelu implementirano rješenje koje pomoću navedenih elektroničkih komponenti i mikroupravljačkog sustava komunicira s mobilnom aplikacijom te daje trenutne rezultate iz okoliša. Mobilna aplikacija je testirana te je prikazano nekoliko očitavanja.

**Ključne riječi:** glazbeni instrument, mikroklima, mikroupravljački sustav, mobilna aplikacija

## ABSTRACT

Device and software development for the musical instrument preservation

This thesis describes in detail the process of development a mobile application and hardware solution for the guitar preservation. It gives a theoretical background and describes problem with microclimate and keeping guitar safe. In the practical part, the solution has been implemented using electronic components and microcontroller system to communicate with application to produce present result. In the end, mobile application was tested and few given results were shown.

**Keywords:** musical instrument, microclimate, microcontroller system, mobile application



## **ŽIVOTOPIS**

Luka Florschütz rođen je 11.09.1992 godine u gradu Osijeku. Osnovnu školu završava u Osijeku. U 2007. godini upisuje III. Gimnaziju u Osijeku koju završava sa vrlo dobrim uspjehom. U 2011/2012. godini upisuje Sveučilišni preddiplomski studiji računarstva na Elektrotehničkom fakultetu u Osijeku i završava preddiplomski studiji 2013/2014. godine. Nakon završetka studija upisuje Sveučilišni diplomski studij programskog inženjerstva.

## **PRILOZI**

1. Diplomski rad u .docx formatu
2. Diplomski rad u .pdf formatu
3. Programsko rješenje projekta