# Klijent

Primeri se mogu videti u fajlovima esplora.js što se funkcionalnosti tiče. Konekcija preko socketa I komunikacija sa serverom, tj primanje podataka se mogu vide na dnu fajla index.html.

Za učitavanje skripti vezanih za bootstrap I jquery potrebna je internet konekcija.

## Movement control

### Dugmad

Klijent preko socketa prima informaciju od servera o tome u kom pravcu treba da se pomeri, u zavisnosti od toga koje je dugme pritisnutno na kartici. Informacija se dobija kao JSON objekat spakovana u argumente **up**, **down**, **right** i **left**. Pri prijemu podataka klijent proverava da li postoji propery **up** i tada poziva odgovarajuće funkcije.

Funkcije moveInsideUp(), moveInsideDown(), moveInsideRigth() i moveInsideLeft() su zadužene za pomeranje plavog četvorougla unutar div-a Movement control. I rade tako što na osnovu veličine div-a i četvorougla proveravaju da li četvorougao sme da se pomeri u tom pravcu a da ne ispadne iz div-a. Potom se četvorougao ukoliko je moguće pomera za 5px u naznačenom pravcu.

### Joystick

Klijent preko socketa prima informaciju od servera o trenutnom položaju jostick-a u intervalu od -512 do 512 kao i o tome da li je joystick pritisnut. Informacija se dobija kao JSON objekat spakovana u argumente **joystickX**, **joystickY** i **button**. Pri prijemu podataka klijent proverava da li postoji propery **joystickX** i tada poziva odgovarajuće funkcije.

Funkcija joystickMove mapira primljene podatke na intervale koji odgovaraju dužini i širina div-a u kome je prikazano kretanje joysticka.

Ukoliko je joystick pritisnut, boja kontrole se menja u zeleno i kretanje je sporije.

## RGB led control

Kod ove kontrole korisnik vrši interakciju sa nekim od pet postojećih slajdera koji menjaju interval pauze i signala kao i boju RGB diode na esplora kartici. Podaci se preko socketa šalju u vidu JSON objekta putem funckije sendRGB. Ova funkcija se poziva na svaku promenu vrednosti slajdera i podatke šalje u atributu **RGB.**

## Temperature

Klijent preko socketa prima informaciju o trenutnoj temepraturi u farenhajtima i celzijusima. Informacija se dobija kao JSON objekat pri čemu se vrši provera da li postoji property celsius a podaci se dalje prosleđuju određenim kontrolama na stranici. Vreme se ažurira u trenutku primanja podataka.

## Piezo control

Ova kontrola omogućava korisniku interakciju sa piezo elementom na kartici. Kada korisnik klikne na neki od nota na klaviru, informacija o noti se putem funkcije sendKey preko socketa šalje na server u atributu PiezoPlay.

## Volume

Klijent preko socketa prima informaciju o jačini zvuka očitanoj preko esplora kartice i prikazuje je. Informacija se prima preko socketa u vidu JSON objekta u atributu loudness i mapira se na interval od 0 do 100.

## Slider

Klijent preko socketa prima informaciju o naponu očitanom sa potenciometra na kartici. Informacija se prima preko socketa u vidu JSON objekta u atributu slider i mapira se na interval od 0 do 100.

## Light

Klijent preko socketa prima informaciju o trenutnoj osvetljenosti u vidu JSON objekta u atributu ligth pri čemu poziva funkciju changeBrightness koja dobijenu vrednost mapira na interval od 0 do 255 i menja pozadinsku boju sijalice.

# Server

Klijent se na server konektuje preko soketa, preko kog se vrši razmena podataka. Serveru prima zahtev u vidu poruke sa naslovom sendEsplora. U poruci se šalje objekat sa podacima o zahtevu. Klijent može da pošalje dve vrste objekata. Jedan je za kontrolu Led diode, a druga za kontrolu zvučnika. Primere objekata možete videti u tabeli 1.

|  |  |
| --- | --- |
| Objekat za kontrolu led diode | Objekat za kontrolu mikrofona |
| **{**  **"RGB": {**  **"red": 255,**  **"green": 255,**  **"blue": 255,**  **"pause": 500,**  **"signal": 500**  **}**  **}** | **{**  **PiezoPlay: "c#"**  **}** |

Table 1 - Primer objekata koji se salju soketom

Kako bi sever mogao da komunicira sa epslora karticom, prvo mora da se poveže preko serialPorta. Kada server treba da zatraži neki podatak od kartice, on kreira objekat koji sadrži potrebnu komandu(Komande su navedene u delu teksta vezanom za esplora karticu), a zatim se taj objekat stavlja na stek, koji radi po principu FIFO. Ovakav pristup je uzet da ne bi dolazilo do konkurentnosti. Komande za JoyStick i led diodu se kreiraju kada se primi zahtev od korisnika. Komande za temperaturu, osvetljenost, slider, i mikrofon se kreiraju na serveru u odredjenom intervalu u funkciji **repeatForUsers**, koja kreira zahteve samo ako postoji konektovanih korisnika na serveru. Podatke o dugmetu i stanju joystika kartica sama šalje podatke, kada se desi promena. Na serveru postoji funkcija **repeat**, koja, ako ima zahteva za obradu šalje zahtev koji je na dnu steka kartici. Kada primi odgovor na taj zahtev, prosleđuje ga preko socket-a klijetnu na obradu, a kartici šalje novu komandu sa steka. U slučaju da nema zahteva na steku, funkcija ce na svaku sekundu krenuti da proverava postojanje nekog zahteva, a kada se pojavi na steku, ponovo će krenuti da šalje komande kartici, preko serialPorta.

# Esplora kartica

Arduino kartica je isprogramirana tako da čita odgovarajuće komande za očitavanje određenih senzora. Tako na primer preko serijske konekcije prima niz karaktera i u zavisnosti od poslatog niza karaktera očitaće odgovarajući senzor i vratiti poruku sa rezultatima u JSON formatu. Pored zahteva, Arduino konstantno očitava senzore za joystick i button-e, i prosleđuje ih preko serijskog porta u JSON formatu.

## Komande

Komande koje Arduino čita sa serijskog porta su sledeće:

**LOxxx|xxx|xxx|xxxx|xxxx** – uključivanje led diode, pri čemu x predstavljaju brojeve. Tri cifre prvog broja označavaju crvenu boju, a zatim slede brojevi za zelenu, plavu, vreme za koje će dioda biti uključena, i vreme za koje će dioda biti isključena respektivno

**LF** – komanda za isključivanje diode

**BOxxxx|xxxx** – komanda za uključivanje zvuka na Esplora kartici. Prvi broj označavaće frekvenciju a drugi vreme trajanja za koje će zvuk biti uključen izražen u milisekundima

**BF** – komanda za isključivanje zvuka

**M** – komanda za očitavanje mikrofona

**T** – komanda za očitavanje temperature

**Q** – komanda za očitavanje senzora svetlosti

**S** – komanda za očitavanje slider-a

**J** – komanda za očitavanje joystick-a

**A** – komanda za očitavanje akcelerometar

**W** – komanda za očitavanje pritisnutih button-a

Arduno je isprogramiran tako da čita slovo po slovo gore navedenih komandi. Nakon prepoznavanja komande Arduino vraća odgovarajuće JSON formate.

## JSON odgovori

Nakon očitavanje komande sa serijskog porta, Arduino će vratiti sledeće JSON podatke

**{ "loudness”: loudness }** – prilikom zahteva za očitavanje senzora mikrofona

**{ "celsius" : " celsius , "fahrenheit" : fahrenheit }** – prilikom zahteva za očitavanje senzora za temperaturu

**{ "light" : value }** –prilikom zahteva za očitavanje senzora svetlosti

**{ "slider" : value }** – prilikom zahteva za očitavanje slider-a

**{ "joystickX" : xValue, "joystickY" : yValue, "button" : button }** – prilikom zahteva za očitavanje joysticka

**{ "axisX" : xAxis, "axisY" : yAxis, "axisZ" : zAxis }** – prilikom zahteva za očitavanje akcelerometar

**{ "up" : swu, "down" : swd, "left" : swl, "right" : swr }** – prilikom zahteva za očitavanje button-a