Unaprijeđenje stolnog nogometa primjenom mikroupravljačke i internet tehnologije

Delaš, Mate

Master's thesis / Diplomski rad

2017

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://urn.nsk.hr/urn:nbn:hr:200:442242

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

UNAPRJEĐENJE STOLNOG NOGOMETA PRIMJENOM MIKROUPRAVLJAČKE I INTERNET TEHNOLOGIJE

Diplomski rad

Mate Delaš

Osijek, 2017. godina

SADRŽAJ

1. UVOD ... 1
   1.1. Zadatak diplomskog rada ... 1
   1.2. Link za pristup internet stranici ... 1
2. STOLNI NOGOMET ... 2
3. PRIMJENJENE TEHNOLOGIJE I ALATI ... 3
   3.1. Arduino razvojno okruženje ... 3
      3.1.1. Arduino Nano razvojni sustav ... 3
      3.1.2. Arduino Mega razvojni sustav ... 4
      3.1.3. Infracrveni senzor (IR sensor) ... 5
      3.1.4. Tipkalo ... 6
      3.1.5. Sedam segmentni pokazivač ... 7
      3.1.6. LED matrični pokazivač (LED Matrix Display) ... 7
      3.1.7. WiFi modul ESP8266 ... 8
   3.2. Internet tehnologije ... 9
      3.2.1. HTML (engl. Hypertext Markup Language) ... 9
      3.2.2. CSS (engl. Cascading Style Sheets) ... 10
      3.2.3. MySQL (engl. My Structured Query Language) ... 11
      3.2.4. PHP (engl. Hypertext Predprocessor) ... 12
4. REALIZACIJA SUSTAVA ... 13
   4.1. Sustav za praćenje i ispis rezultata ... 13
      4.1.1. Unos rezultata ... 17
      4.1.2. Ispis rezultata ... 21
   4.2. Sustav za prijavu igrača i praćenje odigranih utakmica ... 27
5. ZAKLJUČAK ... 33
LITERATURA ... 34
   SAŽETAK ... 35
   ABSTRACT ... 36
   ŽIVOTOPIS ... 37

1. UVOD

Cilj ovog diplomskog rada je izraditi mikroupravljački sustav za brojanje golova u stolnom nogometu, ispis rezultata na semaforu te internet stranici. Prvi dio ovog diplomskog rada je izrada sklopovlja. Za to će se koristiti Arduino Nano kao pomoćni uređaj za brojanje golova pomoću IR senzora te prikaz na semaforu. S druge strane, razvojna maketa Arduino Mega će se koristit kao glavni uređaj za povezivanje mikroupravljačkog sustava s internet stranicom pomoću WiFi modula ESP8266. Drugi dio rada obuhvaća izradu internet stranice s obrascima za prijavu ekipa, rasporeda utakmica te praćenje rezultata.

1.1. Zadatak diplomskog rada

U ovom diplomskom radu potrebno je izraditi mikroupravljački sustav za brojanje golova u stolnom nogometu, prikaz rezultata te komunikaciju pomoću WiFi tehnologije. Potrebno je izraditi Internet stranicu za organiziranje turnira koja omogućuje korisnicima prijavu ekipa te uvid u raspored utakmica. Neke od dodatnih mogućnosti bile bi prikazivanje videa te slikanje tijekom igre.

1.2. Link za pristup internet stranici

http://www.etfos.unios.hr/~mdelas/

2. STOLNI NOGOMET

Stolni nogomet je sport koji svoje osnove veže uz nogomet. Ne zna se točno kada je i gdje nastao, ali se većina povjesničara slaže da su se prvi stolovi pojavili u Njemačkoj osamdesetih i devedesetih godina 19. stoljeća. Iako je taj sport u svijetu relativno popularan, u Republici Hrvatskoj ima registrirano svega par klubova te ga se kao takvog najčešće može sresti po barovima kao i bilijarske stolove, pikado itd.

Pravila igre su slijedeća. Potrebno je ubaciti loptu u protivnički gol kružnim pomicanjem šipki na kojima se nalaze figure igrača. Igra započinje tako što jedan od igrača ubacuje loptu na sredinu terena, a igrač do kojega je lopta došla započinje sa igrom. Jedina stvar koja je zabranjena jest brzo okretanje šipki na kojima se nalaze figure igrača. Za razliku od nogometa gdje se igra na vrijeme, ovdje se igra dok jedan od igrača ne zabije određen broj golova (najčešće 11). U slučaju rezultata 10 : 10, igra se po sistemu tko prvi zabije dva gola razlike.

Kao i kod nogometnog igrališta, tako i kod stola za stolni nogomet, dimenzije mogu varirati. Najčešće dimenzije stola za stolni nogomet su 120 centimetara dužine i 60 centimetara širine. Na stolu se nalazi 11 figura igrača po ekipi, a podijeljeni na 4 šipke po ekipi. Najčešći raspored figura igrača prikazan je u tablici 2.1, dok na velikim natjecanjima igrači imaju mogućnost samostalno rasporediti figure, ovisno o tome kakvu taktiku igre koriste [1].

Red 1	Vratar	1 igrač
Red2	Obrana	2 igrača
Red 3	Napad	3 igrača
Red 4	Vezni red	5 igrača
Red 5	Vezni red	5 igrača
Red 6	Napad	3 igrača
Red 7	Obrana	2 igrača
Red 8	Vratar	1 igrač

Tablica 2.1. Raspored igrača na stolu.

3. PRIMJENJENE TEHNOLOGIJE I ALATI

U ovom poglavlju su opisane korištene tehnologije i alati potrebni za izradu opisanog mikroupravljačkog sustava te internet stranice.

3.1. Arduino razvojno okruženje

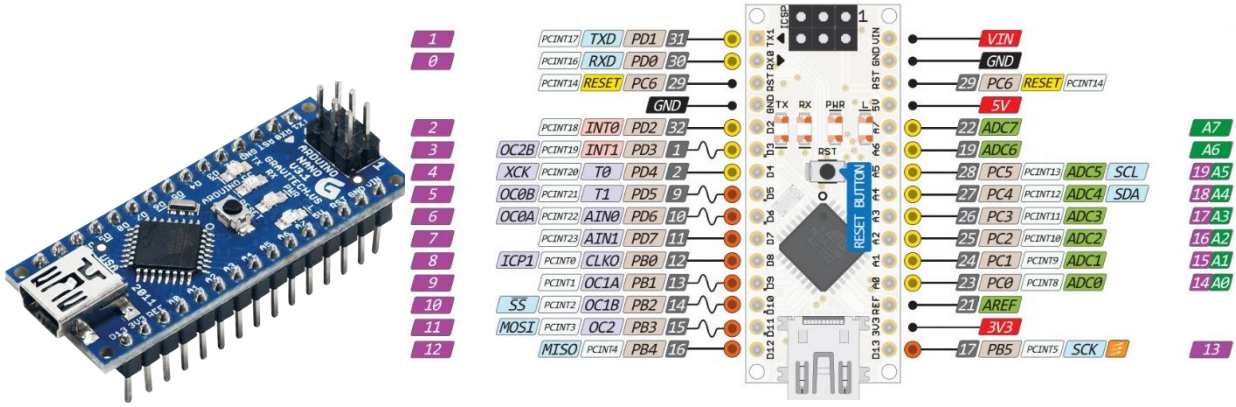
Arduino je platforma otvorenog koda (eng. *open-source*) za kreiranje elektroničnih prototipova bazirana na sklopovlju i programskom paketu koji je fleksibilan i jednostavan za korištenje. Koriste ga programeri, električari, tehničari, ali i hobisti.

Arduino je kao platforma nastao prije desetak godina, a u zadnje vrijeme je postao izrazito popularan budući da je vrlo povoljan i na jednostavan način omogućuje izradu zabavnih i poučnih projekata spajanjem jednostavnih programskih i elektroničnih dijelova u složenije cjeline. Budući da je Arduino platforma otvorenog sustava, omogućeno je potpuno besplatno preuzimanje programa i shema sa službene stranice [2] te drugih stranica gdje se mogu preuzeti gotovi projekti.

Program je podjeljena u dvije glavne funkcije: *setup()* i *loop()*. Funkcija *setup()* se poziva samo jednom i to na početku programa. U njoj se uglavnom inicijaliziraju potrebne varijable. Funkcija *loop()* se poziva nakon funkcije *setup()* i pozva se neprestano kao beskonačna petlja. Ovdje se upisuje program koji se neprestano poziva, a time se omogućuje aktivno kontroliranje određenih dijelova mikroupravljačkog sustava [3].

3.1.1. Arduino Nano razvojni sustav

Arduino Nano razvojni sustav se temelji na upravljanju mikroupravljačem ATmega328p koji sadrži 14 digitalnih i 8 analognih ulazno-izlaznih pinova. Analogni pinovi A0-A5 se mogu koristiti i kao digitalni pinovi, dok su pinovi A6 i A7 isključivo analogni. Sadrži USB priključak koji služi za napajanje, programiranje mikroupravljača i prijenos podataka. Na razvojnoj pločici se nalazi i tipka za resetiranje. Radni napon je 5V, dok su preporučljive vrijednosti ulaznog napona, na nožici V_{IN}, u rasponu od 7 do 12V (min. 6V do max. 20V). Detaljniji prikaz je na slici 3.1. i 3.2 [3].

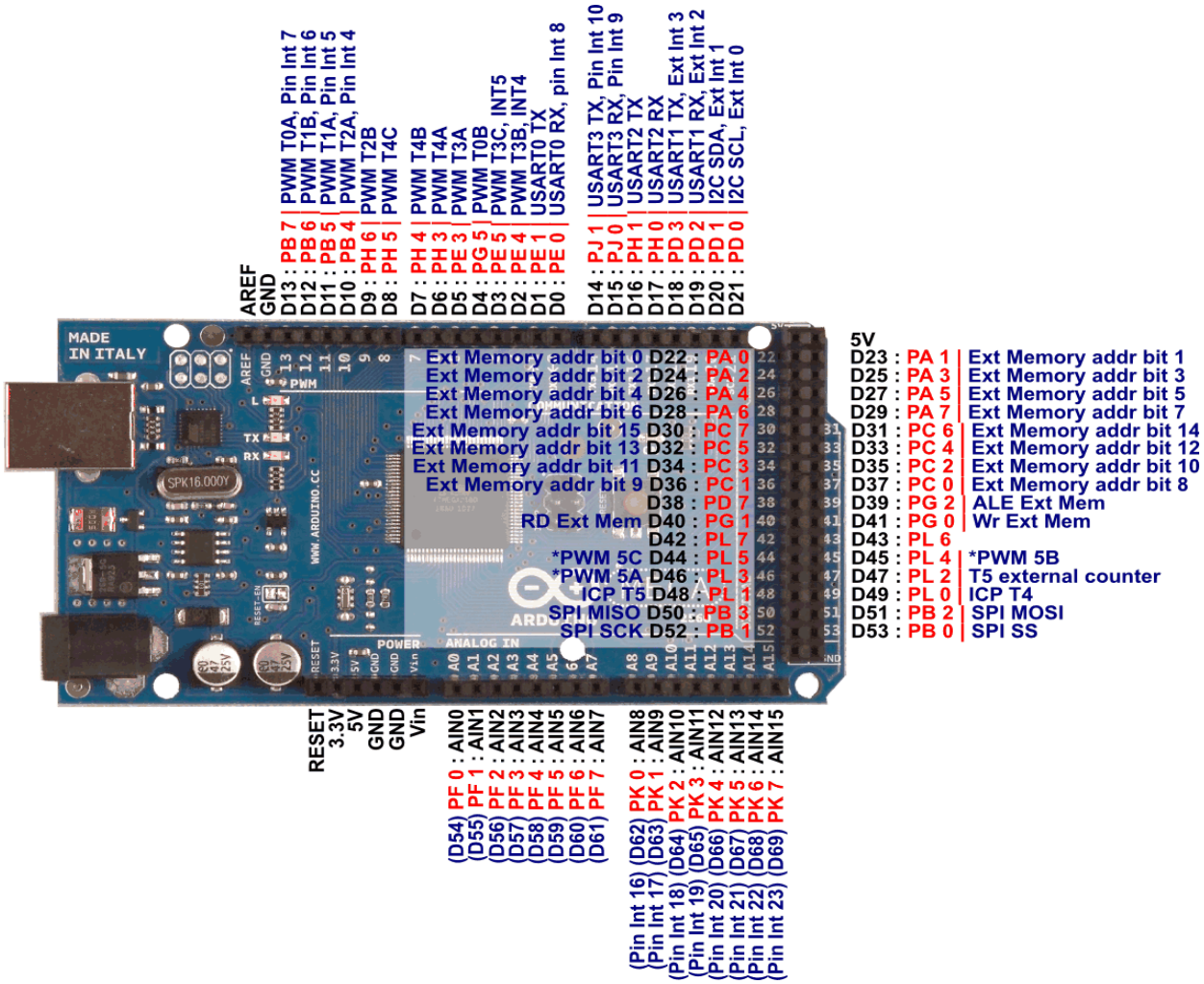


(a) (b)

Slika 3.1. Arduino Nano (a) i raspored pinova (b).

3.1.2. Arduino Mega razvojni sustav

Arduino Mega je robusniji i složeniji razvojni sustav za razliku od Arduino Nano razvojnog sustava. Arduino Mega svoje upravljanje temelji na mikroupravljaču ATmega2560, posjeduje 54 digitalnih i 16 analognih ulazno-izlaznih pinova. Svih 16 analognih pinova se mogu koristiti kao digitalni pinovi. Također posjeduje USB priključak za napajanje i prijenos podataka, ali i dodatni priključak za napajanje pomoću AC/DC adaptera ili baterije. Radni napon iznosi 5V, ulazni napon je od 7 do 12V (granice 6V do 20V). Struja ulaza i izlaza po pinu iznosi 20mA. Detaljan prikaz pinova na razvojnom sustavu Arduino Mega dan je na slici 3.2 [3].

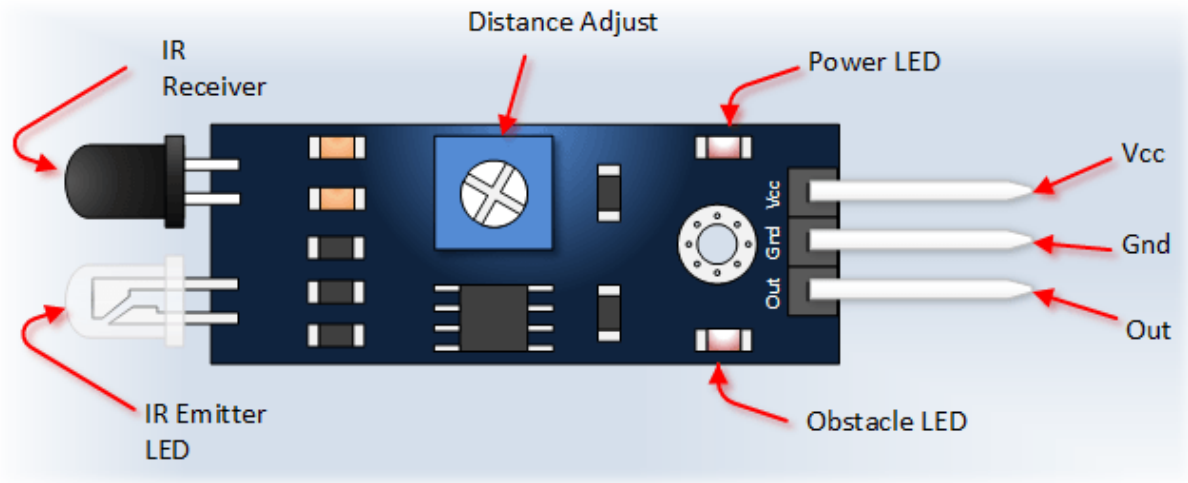


Slika 3.2. Prikaz pinova kod Arduino Mega sustava

3.1.3. Infracrveni senzor (IR sensor)

Za brojanje pogodaka je korišten infracrveni senzor FC-51, a sastoji se od dvije diode od kojih jedna služi za odašiljanje svjetlosnog signala, a druga za njegovo prihvaćanje. Također posjeduje potenciometar s kojim se može podesiti jačina svjetlosnog signala, odnosno udaljenost do koje svjetlosni signal može putovati, dvije LE diode (crvena koja svijetli kada se na senzor dovede napajanje i zelena koja svijetli kada senzor očita prepreku) i tri pina pomoću kojih se povezuje na Arduino razvojno okruženje. Na pin VCC se dovodi napajanje 5V, na pin GND se spaja

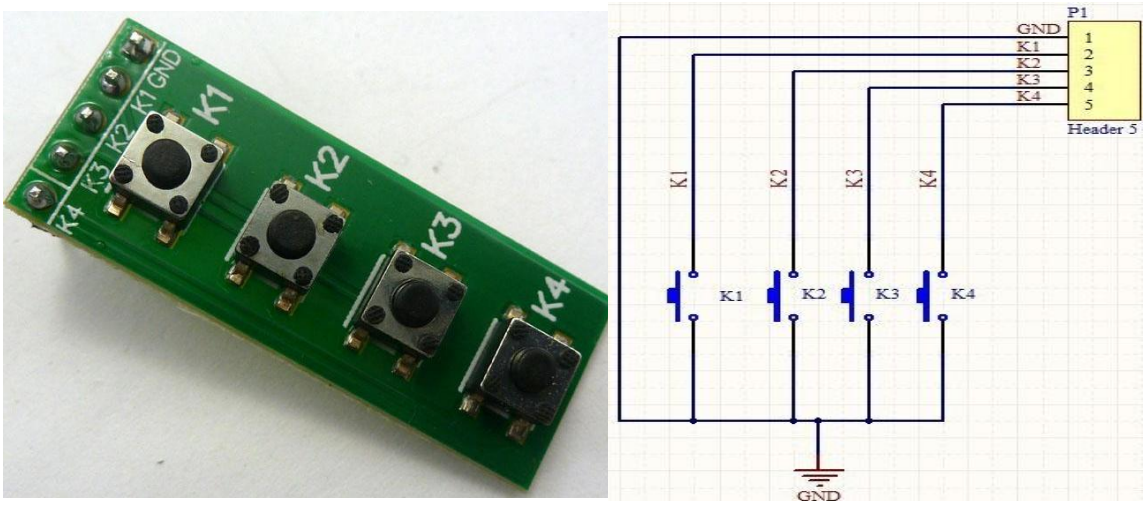
uzemljenje, dok se pin OUT spaja na jedan od analognih ili digitalnih ulazno-izlaznih pinova. Infracrveni senzor prikazan je slikom 3.3 [3].



Slika 3.3. Infracrveni senzor.

3.1.4. Tipkalo

Za ručno podešavanje rezultata korišteno je modul tipkala TB371 koji se sastoji od 4 tipke i 5 pinova. Svaka tipka K1-K4 je pomoću pinova K1-K4 spojena na ulazno-izlazne pinove na Arduinu kao i pin GND. Maksimalni napon je 48V, a maksimalna struja je 1A. Budući da modul tipkala TB371 posjeduje samo 4 tipke, za rad sustava je potrebno 6, tipke K1 i K2 imaju dvije funkcije, modul tipkala TB371 prikazan je na slici 3.4.

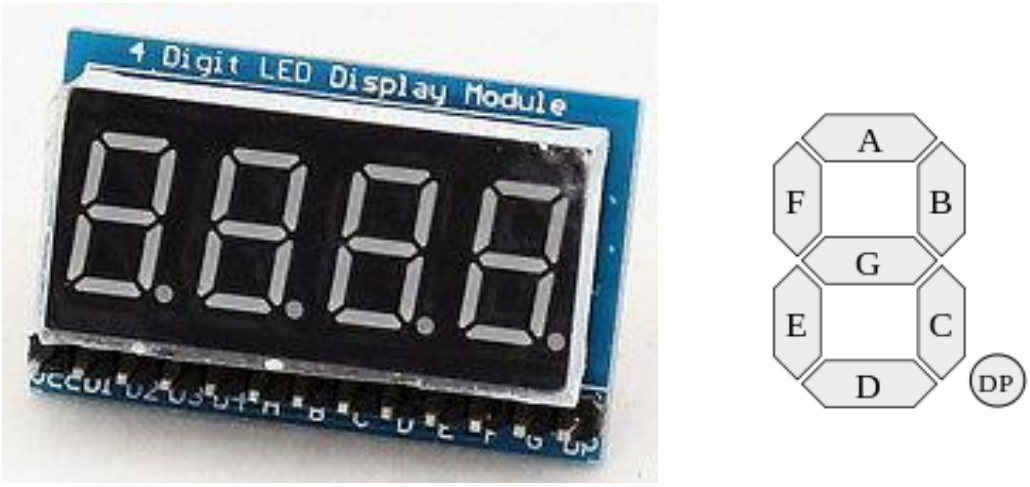


(a) (b)

Slika 3.4. Polje tipkala (a) i shema razvojne pločice (b)

3.1.5. Sedam segmentni pokazivač

Za ispis rezultata korišten je modul s četiri povezana sedam segmentna pokazivača, a prikazan je na slici 3.5a. Korištenje ovakvog modula uvelike olakšava i pojednostavljuje digitlne sustave budući da se na pločici već nalaze potrebni otpornici i tranzistori. Radni naponom je u rasponu od 3.3V do 5V. Na pin VCC se dovodi napajanje, pinovi D1-D4 upravljaju s paljenjem/gašenjem pojedine znamenke sedam segmentnog pokazivača, a pinovi A-G upravljaju s pojedinom diodom na pokazivaču (adresiranje prikazano na slici 3.5b), dok pin DP upravlja s točkom [3].



(a) (b)

Slika 3.5. Modul sedam segmentnog pokaznika s četiri znamenke (a). Adresiranje segmenata (b).

3.1.6. LED matrični pokazivač (LED Matrix Display)

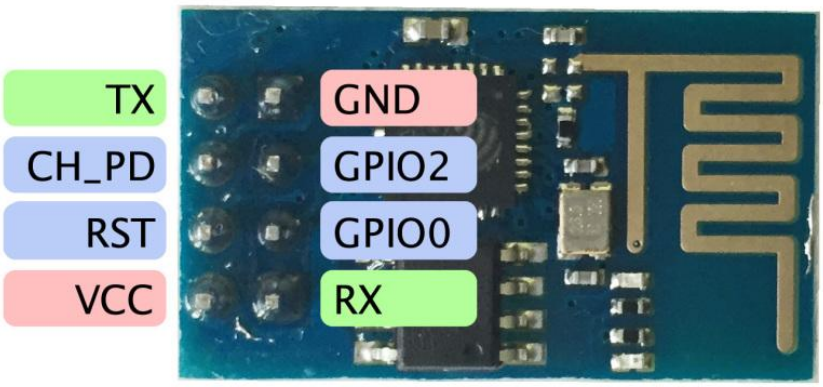
Za ispis rezultata korišten je i 8x8 matrični pokazivač na FC-16 modulu na slici 3.6. Ovaj modul omogućuje povezivanje više matričnih pokazivača u jednu cjelinu. Sastoji se od dva skupa pinova {VCC, GND, DIN, CS, CLK}. Jedan skup pinova služi kao ulaz, dok drugi služi kao izlaz. Postoji više različitih načina adresiranja. U ovom radu je korišteno adresiranje na način da se za svako slovo ili broj stvori objekt veličine 8x8 te za svaki stupac na pokazivaču se dodjeljuje binarna vrijednost. Logička jedinica pali LE diodu, dok ju logička nula gasi. Primjenjuje se za ispis jednostavnih brojeva, slova i znakova kao i za igre: Tetris, Pong, Zmija, itd [3].



Slika 3.6. LED matrični pokazivač.

3.1.7. WiFi modul ESP8266

ESP8266 je WiFi modul s ugrađenim TCP/IP protokolom koji omogućava mikroupravljačima pristup WiFi mreži. Ima ugrađenu Flash memoriju, RAM (*engl. Random Access Memory*) i EEPROM (*engl. Electricaly Erasable Programmable Read-Only Memory*) što omogućuje direktno programiranje iz Arduino sučelja bez potrebe za dodatnim mikroupravljačom. Podržani su WiFi protokoli 802.11b/g/n, ima 1 MB Flash memorije, a radni raspon je u rasponu od 0.8 do 3.6 V. ESP8266 Wifi modul prikazan je na slici 3.10. Sastoji se od Xtens 32-bitnog mikroupravljača, Flash memorije, antene i 8 pinova [4].



Slika 3.10. ESP8266 WiFi modul

3.2. Internet tehnologije

U ovom poglavlju opisane su korištene klijentske i poslužiteljske tehnologije. Pomoću njih je moguća implementacija kako klasičnih web sadržaja, tako i modernih (interaktivnih) rješenja. Korisnicima osiguravaju korištenje različitih internetskih usluga upotrebom nekog od internet preglednika (engl. *browser*). Svaka od obrađenih tehnologija prezentirana je na primjeru izrade web aplikacije zadane u zadatku.

3.2.1. HTML (engl. Hypertext Markup Language*)*

HTML je meta-jezik koji služi za strukturiranje web dokumenta. Web dokument je tekstualni dokument i kao takav se može uređivati u bilo kojem tekstualnom programu. Sadržaj se piše unutar oznaka (engl. *tag*). Sadržaj koji se unosi određuje koja će se oznaka koristiti. Internet preglednik prilikom učitavanja web dokumenta prevodi oznake te se sukladno tome uređuje njihov sadržaj. Svaki web dokument ima svoj URL (engl. *Uniform Resource Locator*), odnosno jedinstveni identifikator lokacije po kojemu se razlikuje od klasičnih elektronskih dokumenata. Osnovna strukturna jedinica *HTML* dokumenta jest *HTML* element, a sintaksno se opisuje na slijedeći način:

- Počinje otvarajućom oznakom (engl. *opening tag*).
- Završava zatvarajućom oznakom (engl. *closing tag*).
- Sadržaj elementa je sve što se nalazi između otvarajuće i zatvarajuće oznake.
- Neki elementi imaju prazan sadržaj.
- Prazni elementi (bez sadržaja) su zatvoreni u otvarajućoj oznaci.
- Većina HTML elemenata može imati atribute koji dodatno opisuju specifičnosti pojedinih oznaka [5].

```
<head>
    <meta charset="UTF-8">
    <title>Stolni nogomet - Naslovnica</title>
    <link rel="stylesheet" href="/style.css">
</head>
<body>
    <div class="wrapper">
        <header>
            <h1>Stolni nogomet</h1>
        </header>
        <nav>
            <a href="administracija.php">Administracija</a>
        </nav>
```

Primjer programa 3.1. HTML zapis internet stranice.

U prikazanom primjeru programa 3.1 sadrži oznaku <*head*> s naslovom stranice i vezom s CSS datotekom. U preostalom dijelu, u oznaci <*body*>, izrađene su oznaka naslova <*h1*> te poveznice <*a*> na internet stranicu "administracija.php".

3.2.2. CSS (engl. Cascading Style Sheets)

CSS je meta-jezik s kojim se uređuje i definira izgled dokumenata kreiranih s tzv. *markup* jezikom. Uređivanje i izgled dokumenata moguće je definirati u bilo kojem od programa za uređivanje teksta. Izgled internet dokumenata dobiva se tako da se elementima ili grupama elemenata dodjeljuju stilovi kao što su: boja, pozicija na stranici, font, visina. Pri tome se prezentacija stranice odvaja od njezinog sadržaja, a *HTML* kod postaje čitljiviji i kraći. *CSS* je donio čitav niz načina uređivanja prikaza podataka koji do tada nisu postojali u *HTML-u*. S *CSS-om* je moguće jednostavnom promjenom nekoliko parametara promijeniti cjelokupni izgled stranice.
CSS također omogućuje izradu prilagodljivih (*engl. responsive*) internet stranica. Taj način izrade stranice je u novije vrijeme posebno zastupljen budući da uređaji s kojima je moguće povezivanje na internet dolaze u različitim veličinama[6].

U primjeru programa 3.2, prikazan je *CSS* kod za vizualno uređivanje dijela internetske stranice. Oznaka *h1* označava da se radi na vizualnoj promjeni glavnog naslova na stranici. Unutar vitičastih zagrada nalazi se način na koji je taj naslov uređen. Prvo je naslov poravnat u sredinu, a zatim mu je dodan font *Open Sans* te veličina slova od 55 točaka (*engl. pixel).* Nakon toga, dodijeljena je boja slova zapisana heksadekadski. Na kraju je još dodan razmak između slova, izgled i boja sjene.

```
h1{
        text-align: center;
        font-family: 'Open Sans', sans-serif;
   font-size: 55px;
        color: #6EABCF;
        letter-spacing: 1px;}
```

Primjer programa 3.2. CSS zapis oblikovanja na internet stranici.

3.2.3. *MySQL (engl.* My Structured Query Language)

Poslužitelj baze podataka (u ovom slučaju MySQL) je program koji može pohraniti veliku količinu informacija te ih organizirati tako da im se lako može pristupiti putem programskih jezika poput *PHP*-a (*engl. Hypertext Preprocessor*) [7]. MySQL je besplatan te je čest izbor kod odabira alata za izradu baze podataka, pogotovo za projekte otvorenog koda. MySQL baze su relacijskog tipa. Taj tip se pokazao kao najbolji način skladištenja i pretraživanja velikih količina informacija.

Prije početka rada u *MySQL Workbench-u* potrebno je osmisliti shemu baze. Ta se shema daljnjim radom pretvara u određeni broj tablica, povezanih relacijama, koje služe za pohranu podataka. Osnovni elementi tablice nazivaju se entiteti, a entitet može biti bilo što: naziv ekipe, redni broj prijave, broj odigranih utakmica, broj pobijeđenih ili izgubljenih utakmica, itd.

Nakon izrade ER dijagrama, pomoću opcije *forward engineer MySQL Workbench* pretvara dijagram u povezane tablice koji čine bazu podataka. *MySQL Workbench* je alat za dizajniranje baze podataka stoga u njemu nije moguće programiranje baze podataka, ali se program može zatražiti putem opcije *forward engineer*.

U primjeru programa 3.3, naredbom *CREATE TABLE* nastaje nova tablica kojoj se dodjeljuje naziv i entiteti. Entitetima se također dodjeljuje naziv i atributi (tip podatka, mora li entitet uvijek imati neku vrijednost, hoće li se sam automatski popunjavati...). Također se dodjeljuju ključevi. *PRIMARY KEY* je jedinstveni ključ za tu tablicu te je nemoguće imati dva ista *PRIMARY KEY-a,* a svaka tablica mora imati jedan *PRIMARY KEY. FOREIGN KEY* je strani ključ koji služi kako

bi se dvije tablice međusobno povezale. Povezivanje se odvija tako da se *FOREIGN KEY* u jednoj tablici referira na *PRIMARY KEY* u drugoj tablici [8].

```
CREATE TABLE ekipe (
  id INTEGER PRIMARY KEY AUTO_INCREMENT NOT NULL,
  ime VARCHAR (128) NOT NULL,
  igrac_1 VARCHAR (128),
  igrac_2 VARCHAR (128)
);
```

Primjer programa 3.3. MySQL program za izradu tablice.

3.2.4. *PHP* (engl. *Hypertext Predprocessor*)

PHP je skriptni jezik pomoću kojega se može kreirati *HTML* stranica na poslužitelju (*engl. server)* prije nego što se ona, popunjena dinamičkim sadržajem, pošalje klijentu. Ovim načinom generiranja sadržaja klijent ne može vidjeti program (skriptu) koji je generirao HTML sadržaj kojeg korisnik vidi. *PHP* je jedan od trenutno najpopularnijih skriptnih jezika te nije ni čudo što broj stranica koji koriste *PHP* raste iz dana u dan. Razlog tomu je što se s nevjerojatnom lakoćom može stvoriti opširna web aplikacija s velikom količinom podataka. Kada bi se ista web aplikacija radila isključivo u *HTML-u* bilo bi potrebno napraviti posebnu stranicu za svaku relaciju. Kako bi se olakšao postupak izrade aplikacije, ali i njezino korištenje, podaci se upisuju u bazu podataka, a pomoću *PHP* skriptnog jezika ti se podaci dohvaćaju na stranici.
Hijerarhija *PHP* datoteka u projektu je takva da se skripte dijele na *GET* skripte, *POST* skripte i skripte opće namjene. *GET* skripte služe za dohvaćanje podataka iz baze podataka. *POST* skripte služe za upisivanje podataka iz obrazaca u bazu podataka dok skripte opće namjene služe za uspostavljanje veze s poslužiteljem ili dohvaćanje manje količine podataka iz baze podatka. Sve *PHP* skripte napisane su čisto u svrhu demonstracije rada s bazom te nisu dodatno osigurane od zlonamjernih napada [9].

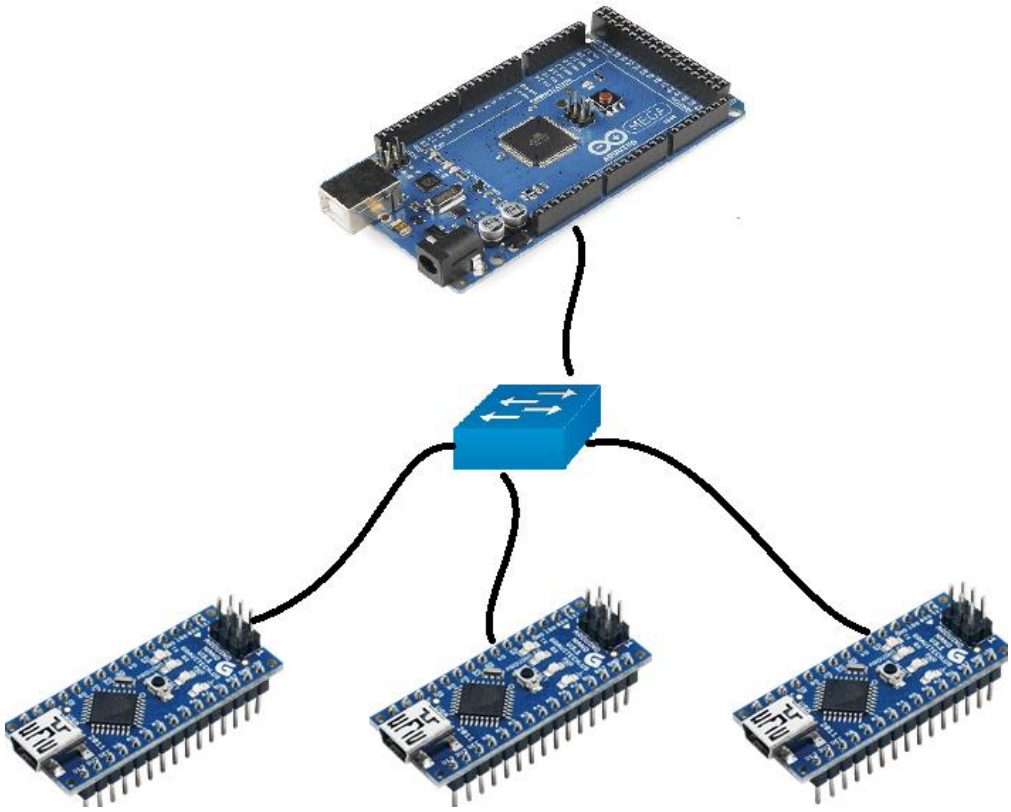
4. REALIZACIJA SUSTAVA

Sustav za unapređenje stolnog nogometa primjenom mikorupravljačke i Internet tehnologije realiziran je tako da su na Arduino Nano povezana dva svjetlosna senzora za automatsku detekciju lopte koja ulazi u gol, sedam segmentnog pokazivača i tipkala koje služi za ručno podešavanje rezultata, ali i za resetiranje brojača, te slanje konačnog rezultata na Internet stranicu radi pohrane u bazu podataka. Na Arduino Mega je spojeno četiri međusobno povezanih matričnih LED pokazivača i WiFi modul, dok se na Internet stranici nalazi stranica za unos ekipa i igrača, postavljanje utakmice koja se mora odigrati, ali i ispis odigranih utakmica poredanih u tablicu gdje se ekipa koja ima najviše bodova nalazi na prvom mjestu, a ekipa sa najmanje bodova na zadnjem mjestu.

Najveći problem predstavljao je povezivanje ta tri zasebna podsustava u jedan zajednički sustav. Povezivanje Arduina i Internet stranice odrađeno je pomoću WiFi modula, dok je za povezivanje Arduino Nano i Arduino Mega razvojnog sustava korištena *I²C* serijska komunikacija.

4.1. Sustav za praćenje i ispis rezultata

Sustav za praćenje i ispis rezultata je zamišljen na način da je jedan ili više podsustava koji čine Arduino Nano s pripadajućim komponentama povezan sa glavnim podsustavom kojeg čini Arduino Mega s pripadajućim komponentama prikazano na slici 4.1.. Na svaki Arduino Nano razvojni sustav spaja se tipkalo, dva svjetlosna senzora i sedam segmentni pokazivač, te se kao takav postavlja na stol za stolni nogomet i čini radnu stanicu. Svaka radna stanica može raditi zasebno i u potpunosti je funkcionalna budući da ima sposobnost unošenja i prikaza rezultata. Takav način rada pogodan je ako se odigravaju prijateljske utakmice. Ukoliko se odigrava turnir, tada se radne stanice spajaju na glavni podsustav pomoću *I²C* serijske komunikacije koji omogućava prikaz rezultata na semaforu, ali i praćenje rasporeda utakmica, rezultata i poziciji na ljestvici.



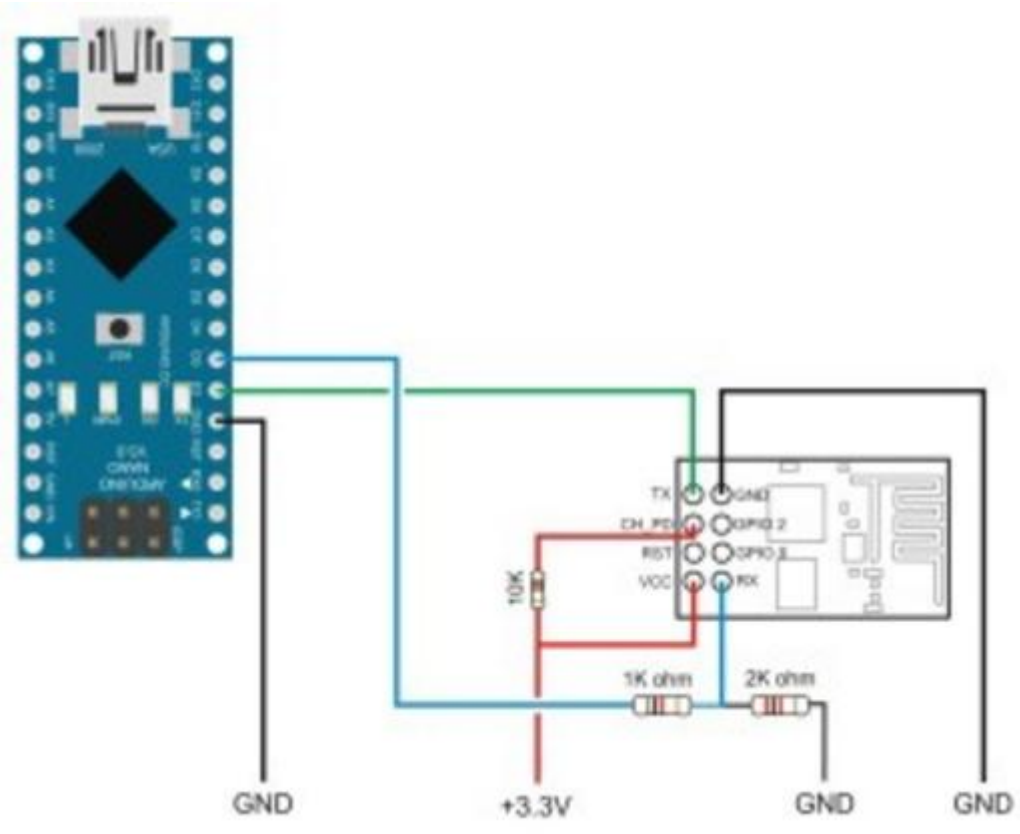
Slika 4.1. Prikaz povezivanja sustava

Možemo reći da je serijska komunikacija zapravo proces slanja dijelova podataka spremljenih u bajtove preko jednog kanala kojeg nazivamo sabirnica (*engl. Bus*). Ti podaci (bajtovi) se šalju uzastopno jedan za drugim, za razliku od paralelne komunikacije gdje se podaci šalju na način da se istovremeno šalje više bajtova kroz više kanala. Prednosti serijske komunikacije su te da se koristi manji broj žica prilikom povezivanja (lakša obrada vremena propagacije), sustav je jeftiniji, povećana je sigurnost kanala,…

Sabirnica *I²C* (*engl. Inter-Integrated Circuit*) vrši komunikaciju putem dvije linije **SDA** (Serial Data - prenosi podatke) i **SCL** (Serial Clock - prenosi takt). Svaka komponenta koja je spojena na sabirnicu ima svoju adresu, koja na toj sabirnici mora biti jedinstvena. Svaka komponenta može implementirati *I²C* kao upravljač ili kao izvršitelj (*engl. master/slave*). To je model komunikacijskog protokola gdje jedan uređaj ili proces (*master*) kontrolira jedan ili više uređaja ili procesa (*slave*). Kod postavljanja komunikacije u Arduino razvojnom sustavu odabrana je način komunikacije u kojem izvršitelj (Arduino Nano) šalje podatke, a upravljač (Arduino Mega) prima podatke [9]. Sam proces prenošenja podataka se odvija na način da upravljač „ispituje“ izvršitelje imaju li kakve podatke za poslati. Ukoliko ima, odradi se prijenos i upravljač nastavlja dalje sa upitom prema preostalim izvršiteljima iz razloga što bi došlo do zagušenja komunikacijskog kanala ukoliko bi svaki izvršitelj u trenutku promjene slao tu istu promjenu

upravljaču. U ovom slučaju postoji samo jedan upravljač i jedan izvršitelj, komunikacijski kanal je stalno otvoren.

Kako bi omogućili povezivanje Arduino razvojnog sustava i Internet stranice korišten je WiFi modul ESP8266 sa ugrađenim TCP/IP protokolom. Samo pozivanje Arduina i ESP8266 odrađen je na način da se napon od 3.3 V sa Arduina dovodi na pinove Vcc i Ch_Pd, Gnd se spaja sa Gnd, a pinovi 10 (*Software Rx)* i 11 (*Software TX*) se spajaju na Txd, odnosno na Rxd kao što je prikazano na slici 4.2.



Slika 4.2. Shematski prikaz povezivanja Arduina i ESP8266

Nakon što se ispravno odradi povezivanje Arduina i ESP8266, potrebno ga je i pravilno programirat. Na početku je potrebno utvrditi ispravnost povezivanja ta dva sustava. To je moguće napraviti sa slijedećim kodom.

```
#include <SoftwareSerial.h>
SoftwareSerial esp8266(2, 3);
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  Serial.println("Started");

  // set the data rate for the SoftwareSerial port
  esp8266.begin(115200);
  esp8266.write("AT\r\n");
}
void loop() {
  if (esp8266.available()) {
    Serial.write(esp8266.read());
  }
  if (Serial.available()) {
    esp8266.write(Serial.read());
  }
}
```

Kako bi ispitali ispravnost navedenog programskog koda potrebno je koristiti AT naredbe.

AT naredbe su naredbe kojima je moguće upravljat WiFi modulima uključujući i ESP8266. Neke od naredbi su *AT* (provjerava uspješnost povezivanje i šalje povratnu informaciju. Ukoliko je veza uspješna, ispiše se „OK“). Zatim za resetiranje modula koristi se naredba *AT+RST*, a

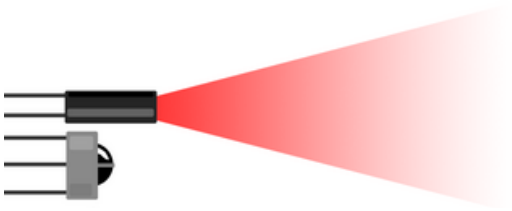
naredba *AT+CWJAP="SSD","PASSWORD"* povezuje ESP8266 sa određenom mrežom čiji se naziv i lozinka mora naznačit u programskom kodu [4].

Nakon što je uspješno odrađeno povezivanje i napisan ispravan kod, pritiskom na K1 taster dužim od 3 sekunde šalje se zahtjev za unošenje rezultata za aktivnu utakmuci u obliku *localhost:8080/rezultat.php?rezultat_ekipe_1=(x)&rezultat_ekipe_2=(y)* (na mjestima gdje je (*x*) *i* (*y*) stoji stvarna vrijednost varijable *x* i y.

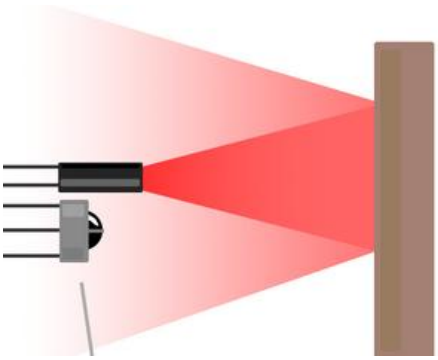
4.1.1. Unos rezultata

Unos i praćenje rezultata omogućeno je na dva različita načina. Primarni način je pomoću svjetlosnih IR senzora, a sekundarni pomoću tipkala.

Svjetlosni IR senzor radi na principu odašiljanja svjetlosnog signala čija se jačina određuje pomoću potenciometra. Ukoliko se ispred LE diode koja odašilje svjetlosni signal nađe prepreka, svjetlosni signal će se odbiti i vratiti nazad do senzora te na taj način svjetlosni IR senzor prepoznaje nalazi li se ispred njega kakva prepreka. Na slici 4.3. i 4.3. prikazana su dva stanja rada svjetlosnog IR senzora.



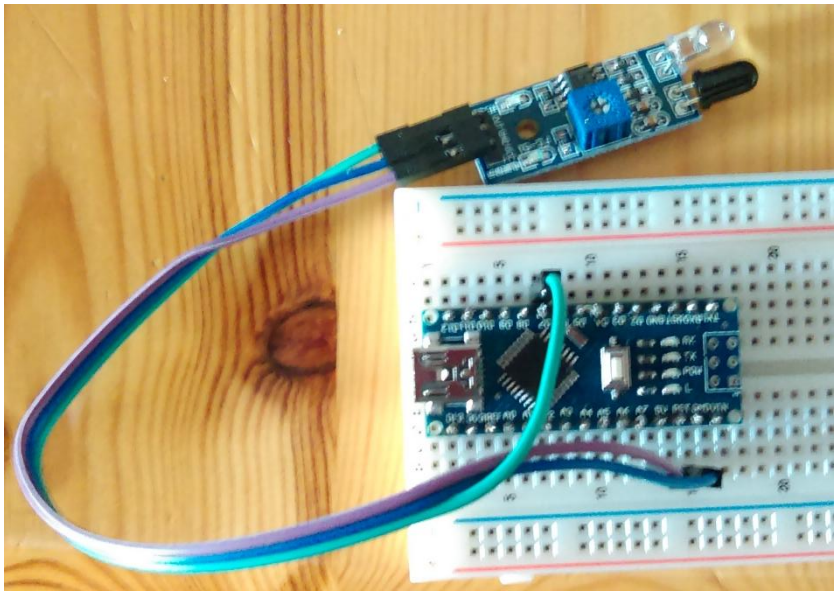
Slika 4.3. Nema prepreke - nema povratnog snopa svjetlosti



Slika 4.4. Prepreka - ima povratnog snopa svjetlosti

Kod stolnog nogometa svjetlosni IR senzor treba postaviti unutar gola tako da svaki put kada lopta uđe u gol, prekine svjetlosni snop i tako poveća ukupan rezultat. Kod postavljanja senzora treba prilagoditi jačinu svjetlosnog signala kako senzor ne bi očitavao suprotnu stranu gola i tako davao kriva mjerenja.

Način spajanja prikazan je na slici 4.5. Kada se senzor spoji na Arduino, upali se crvena LE dioda koja pokazuje da je senzor pod naponom. Zelena LE dioda pali se samo ako senzor očita prepreku, tj. ako očita povratnu svjetlost.

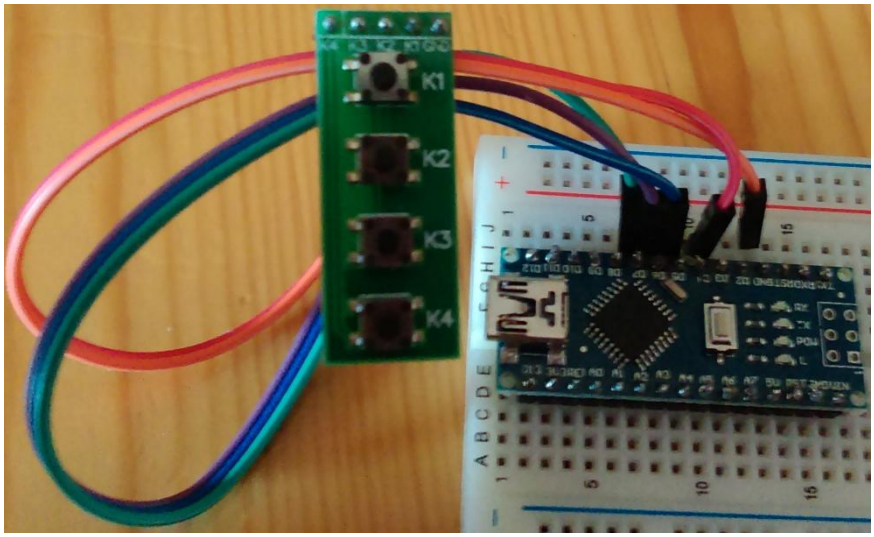


Slika 4.5. Spoj IR senzora i Arduina

Nakon što su IR senzor i Arduino Nano povezani, napisani programski kod prebačen je na Arduino kako bi se testirala njegova ispravnost. Prilikom programiranja potrebno je obratiti pozornost na to da mikroupravljači rade izrazito velikom brzinom i ponavljaju beskonačnu petlju. Način na koji se to rješava jest taj da sustav treba provjeravati prošlo i trenutno stanje te ukoliko se ta dva stanja razlikuju, povećava se vrijednost varijable. Također je potrebno da senzor „okida“ samo na rastući brid kako bi se vrijednost varijable uvećala samo za jedan, neovisno o tome koliko kojom brzinom lopta prolazi .

Prilikom rada sa IR senzorom potrebno je paziti da se senzor ne nalazi u blizini nekog jakog izvora svjetlosti budući da onda prijemnik prepozna tu svjetlost kao prepreku i daje netočna očitanja.

Drugi način detektiranje pogotka i unošenja rezultata je tipkalo. Tipkalo je zapravo pomoćni sustav koji se koristi u slučaju da iz nekog razloga svjetlosni IR senzor ne očita prolazak lopte kroz gol. Način na koji su spojeni prikazan je na slikom 4.6.



Slika 4.6. Spoj tipkala i Arduina

Korišten je modul tipkala TB371 koje na sebi ima 4 tipke. Svakom od tipki dodijeljena je sljedeća funkcija:

- K1 - povećavanje rezultata crvenog tima / slanje rezultata na Internet stranicu
- K2 - smanjivanje rezultata crvenog tima / resetiranje brojača
- K3 - povećavanje rezultata plavog tima
- K4 - smanjivanje rezultata plavog tima

Svakoj tipki moguće je dodijeliti više funkcija rada, samo treba definirati uvjete u programskom kodu kada će tipka odraditi zadanu funkciju. Budući da taster K1 i K2 imaju samo dvije funkcije, korištena je logika dugog i kratkog pritiska tipke. Ukoliko je tipka pritisnuta manje od 3 sekunde odradit će se prvi uvjet koji je povećavanje odnosno smanjivanje rezultata, a ako je tipka pritisnuta duže od tri sekunde odradit će se drugi uvjet što je prikazano u programskom kodu ispod.

```
//Reset rezultata-------------------------------------
float milliSeconds = 0;
int counter_time = 1;
int restart_time = 3000;
int counter = 0;
voide setup()
{
pinMode(buttonPin, INPUT_PULLUP);
}
void loop()
{
  while (digitalRead(1) == LOW ){
    delay(100);
    milliSeconds = milliSeconds + 100;
    }
  if (milliSeconds >= restart_time){
    x=0;
    y=0;
  }
  else if(milliSeconds >= counter_time){
    x--;
  }
  milliSeconds = 0;
}
```

Programski kod je napisan tako da cijelo vrijeme dok je pritisnut taster provjerava koliko je vremena prošlo i sprema u varijablu *milliSeconds*, te u *if* petlj uspoređuje to vrijeme sa 3000 milisekundi (3 sekunde) koje je zadano varijablom *reset_time*. Ukoliko su vrijednosti jednake ili je varijabla *milliSeconds* veća od varijable *reset_time* tada se vrijednost varijabli *x* i y postavlja u nula. Varijabla x je rezultat crvenog tima, a varijabla y rezultat plavog tima. Ukoliko taj uvjet nije zadovoljen, odnosno taster je pritisnut manje od 3000 milisekundi, odradi se inkrement varijable *x*. Na kraju je potrebno brojač *milliSeconds* postaviti u nula.

4.1.2. Ispis rezultata

Ispis rezultata se također odvija na dva različita načina. Rezultat se ispisuje na sedam segmentni i matrični LED pokazivač.

Kako bi se izbjeglo usložavanje sustava, korišten je modul sa sedam segmentnim pokazivačem. Sedam segmentni pokazivač sadrži 4 znamenke (*engl. Digit*). Na lijeve dvije znamenke ispisuje se rezultat crvene ekipe, a na desne dvije rezultat plavog tima. Slika 4.7. prikazuje način spajanja.



Slika 4.7. Spoj sedam segmentnog pokazvača i Arduina

Sedam segmentni pokazivač učitava vrijednosti varijabli *x* i *y* pali određenu kombinaciju LE dioda pokazivača koje su zadane u varijabli *volatile byte segments[]*. Adresiranje je odrađeno u binarnom obliku gdje za logičku nulu dioda ne svijetli, a za logičku jedinicu svijetli. Raspored adresa binarnih vrijednosti je DP/G/F/E/D/C/B/A. U kodu ispod je prikazan način kreiranja brojeva od 0 do 9.

```
volatile byte segments[]={
  0b00111111, // 0
          …
  0b01101111, // 9
 };
```

Ovakvim načinom adresiranja kreirani su brojevi od 0 do 9 te takav način omogućuje ispis bilo koje vrijednosti rezultata. Kako bi se mogli ispisati brojevi veći od 9, potrebno je odrediti način ispisa desetica i jedinica. Pomoću ovog modula najveća vrijednost koju je moguće ispisati je 99, a budući da rezultat ne može biti u minusu, najniža vrijednost za ispis je 0. Ograničavanje raspona brojeva prikazano je u programskom kodu ispod.

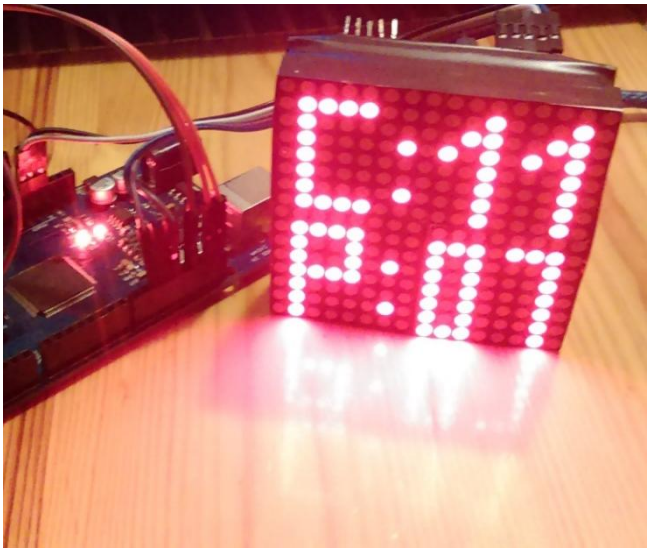
```
  if(x>99) { x=99; }
  if(x< 0) { x= 0; }
  if(y>99) { y=99; }
  if(y< 0) { y= 0; }
  number[0] = x/10;
  number[1] = x%10;
  number[2] = y/10;
  number[3] = y%10;
```

Za ograničavanje vrijednost rezultata u rasponu od 0 do 99 korištene su *if* petlje. Ukoliko je broj veći od 99, ispisuje se kao 99, a ukoliko je manji od 0, ispisuje se kao 0.

Varijable *x* i *y* sastoje se od 2 grupe znamenaka, od desetica i jedinica. Desetice se određuje tako da se vrijednost koja je unesena u varijablu *x* ili *y* pomoću sustava za unošenje rezultata dijeli s brojem 10. Kada je zadana funkcija u formatu *number[0] = x/10;* tada se u *number[0]* sprema vrijednost cijelog broja koja je nastala prilikom dijeljenja (npr. 73/10=7.3 → *number[0]=7*). Za

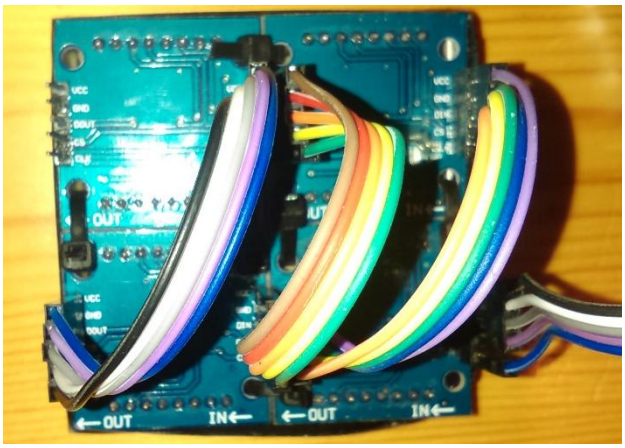
određivanje jedinica koristi se operator *modulo* (*%*). Taj operator radi dijeljenje varijable *x* ili *y*, ali kao rezultat uzima ostatak od cjelobrojnog dijeljenja (npr. 73%10=7.3 → *number[0]=3*).

Matrični LED pokazivač spojen je na Arduino Mega, a nakon što se tako povežu, omogućena im je komunikacija pomoću *I²C* sabirnice, prethodno spomenute u tekstu. Povezivanje Arduino Mega i modula FC-16 prikazan je na slici 4.8.



Slika 4.8. Povezivanje modula LED matričnih pokazivača i Arduino Mega razvojnog sustava

Razlog zašto je modul FC-16 pogodan za izradu semafora za stolni nogomet jest taj što omogućuje serijsko povezivanje više modula sa LED matričnim pokazivačima u jednu cjelinu. Sa svoje stražnje strane ima naznačeno koji su pinovi za ulaz, a koji za izlaz. Međusobno povezivanje četiri modula prikazano je na slici 4.9.



Slika 4.9. Međusobno povezivanje više modula

Iz slike 4.9. je također vidljiv način na koji semafor ispisuje rezultat. Sva slova i brojevi spremljeni su kao objekti, a pozivanjem objekta izvodi se ispis na semaforu. Za kreiranje ovakve vrste ispisa na LED matričnom pokazivaču potrebno je učitati biblioteke *Sprite.h* i *Matrix.h*. koje se pozivaju navedenim redoslijedom. Nakon što su biblioteke čitane, potrebno je kreirati objekte koji će se kasnije pozivati prilikom ispisa

```
Sprite C = Sprite(8, 8, B00000000, B01111111, B01000001, B01000001, B01000001, B00000000, B00100100, B00000000);

Sprite P = Sprite(8, 8, B00000000, B01111111, B00001001, B00001001, B00001111, B00000000, B00100100, B00000000);

Sprite nula = Sprite(8, 8, B01111111, B01000001, B01111111, B00000000, B01111111, B01000001, B01111111, B00000000);
```

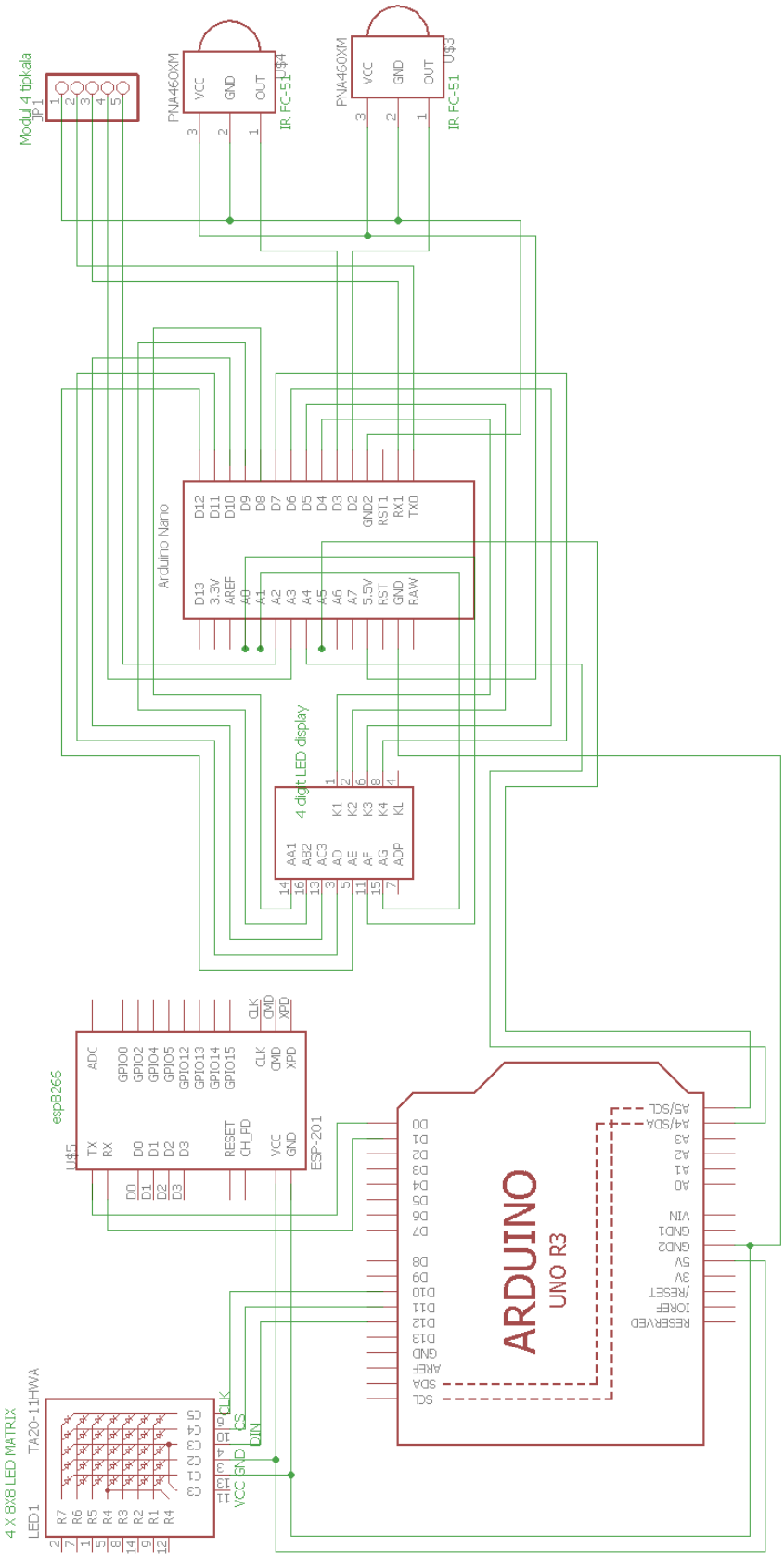
U kodu iznad prikazano je kreiranje objekata za ispisivanje „C:“, „P:“ i broja 0 na matričnom pokazivaču. Prve dvije znamenke su x i y koordinate tj. širina i visina matričnog pokazivača. Potrebno je paziti da prvi broj koji predstavlja širinu odgovara broju bitova u binarnom adresiranju. Adresiranje se vrši tako da svaki binarni broj predstavlja stupac s lijeva na desno, a znamenke u broju predstavljaju LE diode na pokazniku tako da lijeva brojka upravlja sa najnižom LE diodom, a desna brojka sa najvišom LE didom.. U ovom slučaju logička nula gasi LE diodu na pokazivaču, a logička jedinica pali.

```
  myLeds.write(0, 0, C);
  myLeds.write(16, 0, P);
if (x == 0) {
myLeds.write(8, 0, nula);
}
```

U kodu iznad pokazano je ispisivanje kreiranih objekata na LED matrični pokazivač pomoću naredbe *myLeds.write*. I ovdje je potrebno naznačiti točke koordinatnog sustava na kojima želimo započeti sa ispisom. Točka (0,0) je donja lijeva LE dioda prvog LED matričnog pokazivača, dok je točka (16,0) također donja lijeva točka, ali trećeg LED matričnog pokazivača.

Za razliku od ispisivanja početnih slova ekipa koji se ne mijenjaju, rezultat se mijenja svakom promjenom varijable *x* i *y* te je zato nužno koristiti *if* petlju. Na takav način provjerava se vrijednost varijabli te se shodno tome prikazuje ispis objekta kojem je dodijeljena vrijednost varijable trenutnog rezultata.

Konačna shema sustava za unos i ispis rezultata prikazana je na slici 4.10.,a izgleda ovako:

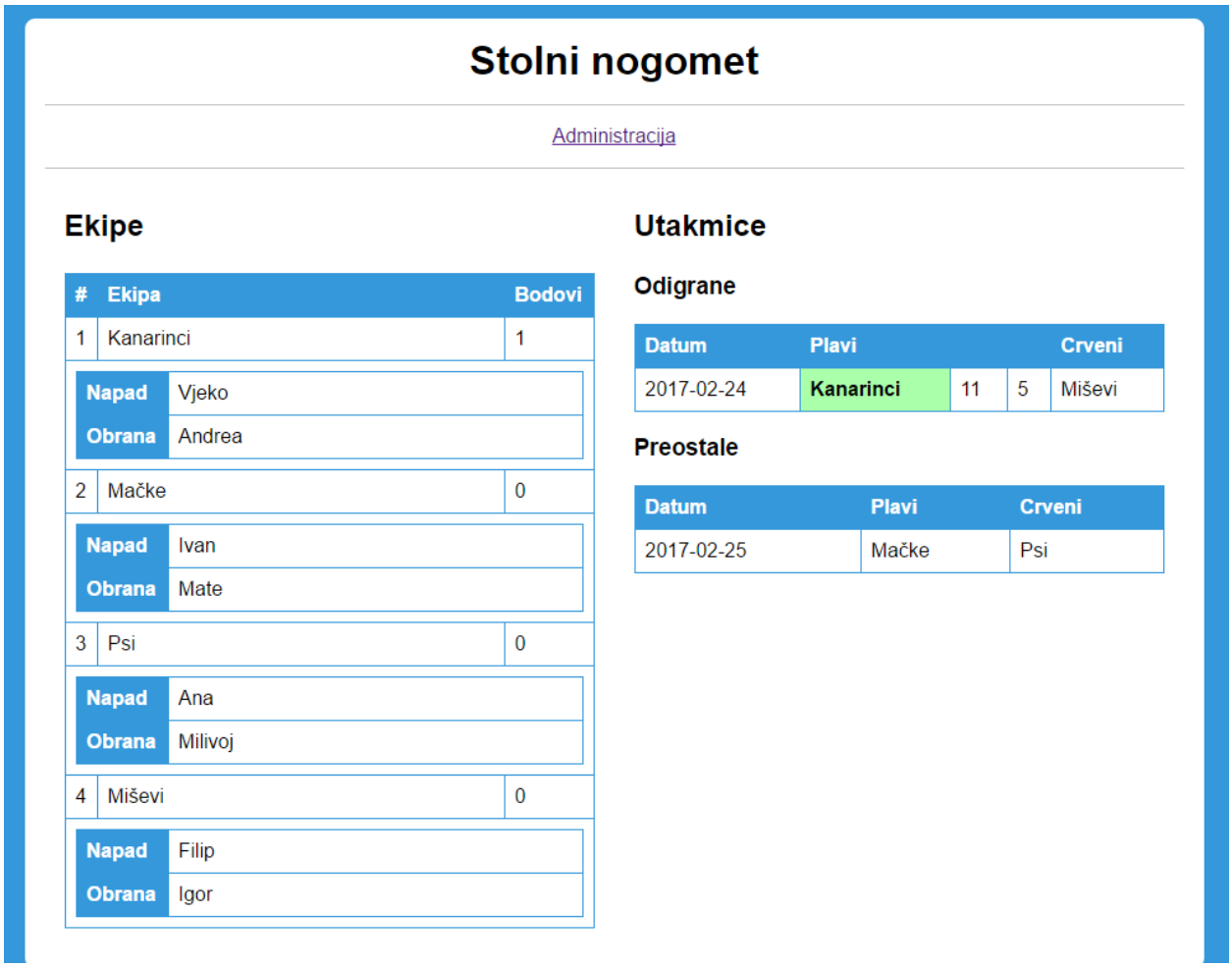


Slika 4.10. Shema mikroupravljača sa perifernim uređajima

4.2. Sustav za prijavu igrača i praćenje odigranih utakmica

Sustav za prijavu igrača te praćenje odigranih utakmica napravljen je u obliku Internet stranice za čiju su izradu korištene već navedeni alati za izradu Internet stranica. Sami sustav zamišljen je kao pomoć pri odigravanju turnira za što olakšava proces kod prijave ekipa te praćenje odigranih utakmica.

Internet stranica je podijeljena na dva dijela. Prvi dio je Naslovnica koji je odstupna svim, a na naslovnici se nalaze opći podaci o turniru. Na slici 4.11. prikazan je izgled naslovne stranice.



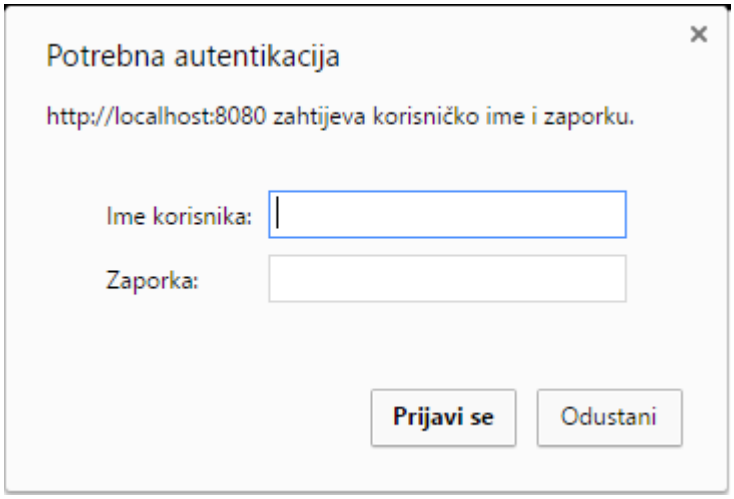
Slika 4.11. Izgled naslovne stranice

Kao što je vidljivo na slici 3.10. sa lijeve strane nalazi se popis ekipa koje sudjeluju na turniru, te je također vidljivo koji igrači pripadaju kojoj ekipi i koju poziciju igraju. Poredak ekipa se vrši prema tome koja ekipa ima više bodova tako da se ekipa sa najvećim brojem bodova nalazi na prvom mjestu, a ekipa sa najmanjem brojem bodova na zadnjem mjestu. Budući da se u stolnom

nogometu ne igra na vrijeme, izjednačeni rezultat nije moguć te je tako dodijeljeno da pobjeda donosi jedan bod, dok poraz ne donosi niti jedan.

Sa desne strane nalazi se ispis utakmica podijeljen u dvije tablice. Prva tablica pokazuje odigrane utakmice. Kao što je vidljivo ispisuje se datum odigravanja utakmice, koja ekipa je bila crvena, a koja plava te konačan rezultat. Pobjednički rezultat obojan je blago zelenom bojom kako bi se pobjednik naglasio i bio uočljiviji. Druga tablica prikazuje utakmice koje se još nisu odigrale, koja ekipa igra sa crvenim igračima, a koja sa plavim i datum odigravanja utakmice.

Ispod samog naslova stranice nalazi se tipka pomoću koje se pristupa administratorskom dijelu stranice. Kako bi se spriječio ne ovlašteni pristup tom dijelu stranice postavljen sustav za prijavljivanje. Na slici 4.12. prikazan je sustav za prijavljivanje.



Slika 4.12. Sustav za prijavljivanje na administratorski dio stranice

Nakon što se u određena polja unese odgovarajuće korisničko ime i lozinka, pritiskom na gumb „Prijavi se“ otvara se administrativni dio stranice. Slika 4.13. prikazuje izgled gornjeg dijela administrativne stranice.



Slika 4.13. Izgled administrativnog dijela stranice (gornji dio)

Slika 4.13. prikazuje obrazac za kreiranje novih ekipa koje sudjeluju na turniru. Potrebno je unijeti ime ekipe, ali i imena igrača pazeći na to koji igrač igra napad, a koji obranu. Nakon što je odrađen unos potrebnih podataka pritiskom na gumb Nova ekipa, podaci se spremaju u bazu podataka.

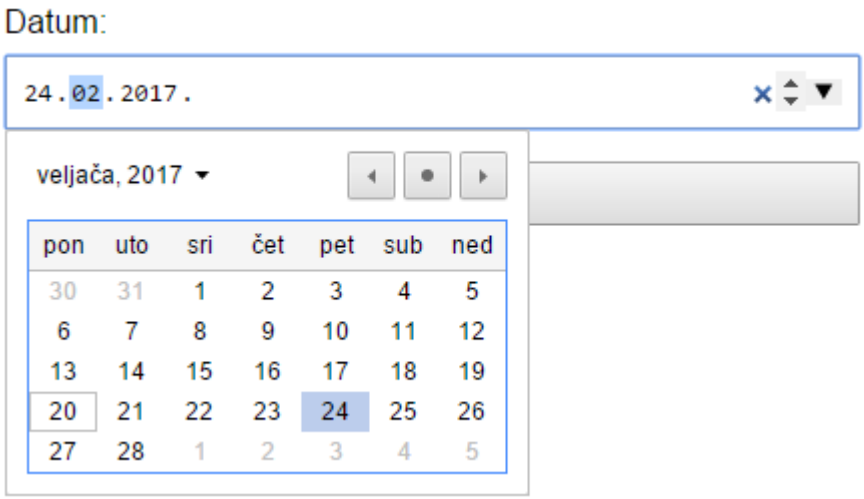
S druge strane se vidi obrazac za kreiranje nove utakmice. Pomoću padajućeg izbornika sa slike 4.14. odabire se ekipa koja će prilikom odigravanje utakmice biti plavi ili crveni tim.



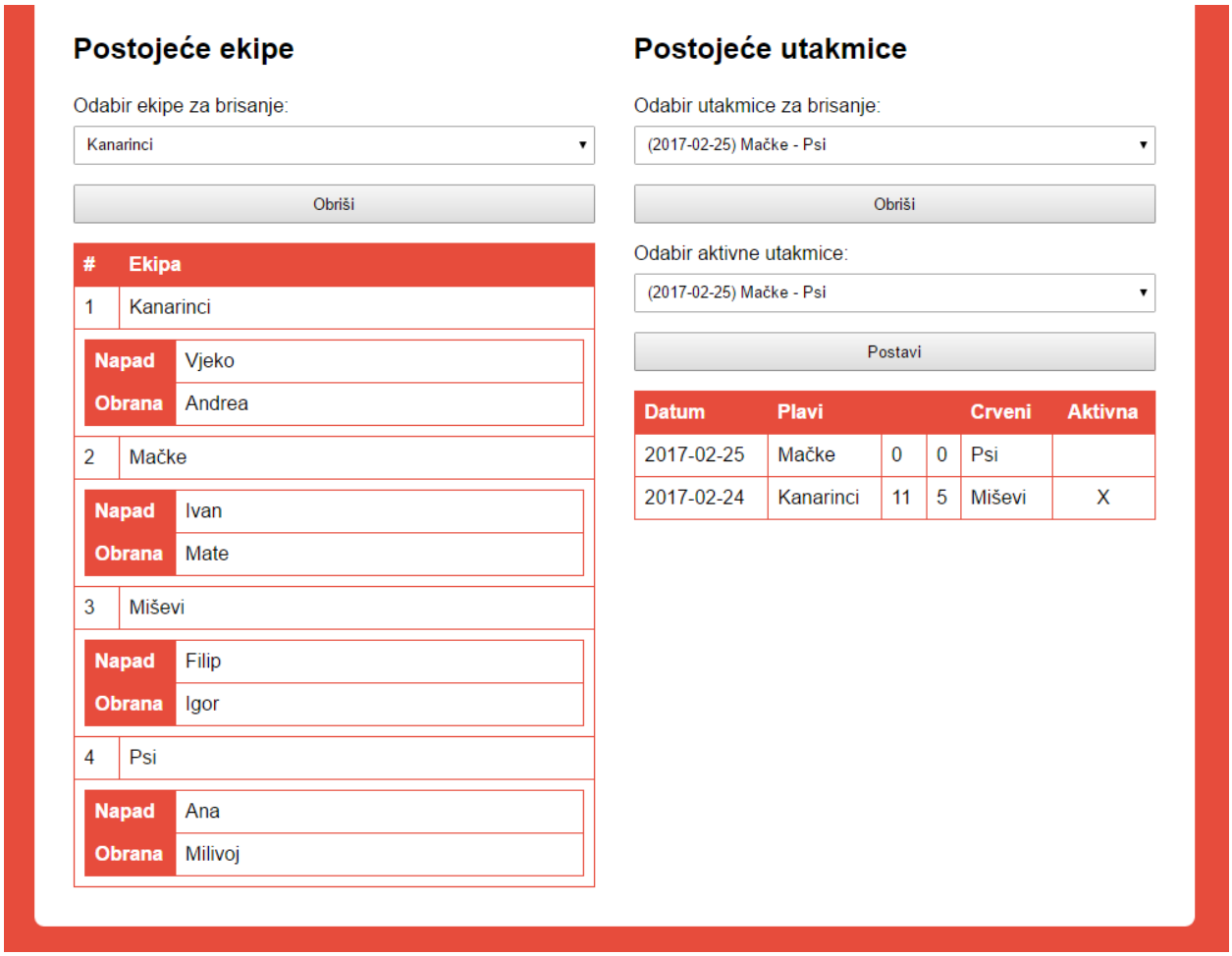
Slika 4.14. Padajući izbornik za odabir ekipa

Nakon što su odabrane ekipe potrebno je odabrati datum kada će se ta utakmica odigrati. To je moguće odraditi na tri načina. Prvi je da se izravno upiše brojka za dan, mjesec i godinu, drugi je da se pomoću strjelica za gore ili dolje podesi željena vrijednost za sve tri grupe znamenaka i

treći način je da se otvori padajući izbornik s kalendarom te se kliko miša odabere željeni datum kao što je prikazano na slici 4.15.



Slika 4.15. Padajući izbornik za odabir datuma



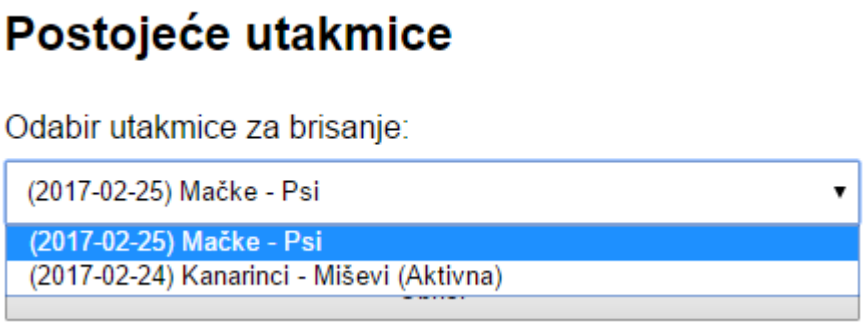
Slika 4.16. Izgled administrativnog dijela stranice (donji dio)

Na slici 4.16. prikazan je donji dio stranice za administraciju. S lijeve strane nalazi se popis postojećih ekipa skupa s njihovim članovima i pozicijom koju igraju. U ovom slučaju ekipe se ispisuju abecednim redom. Također tu je i padajući izbornik za brisanje postojeće ekipe prikazan na slici 4.17. Brisanje se vrši tako da se u padajućem izborniku odabere ekipa koju se želi obrisati te se pritiskom na gumb Obriši ekipa briše iz baze podataka.



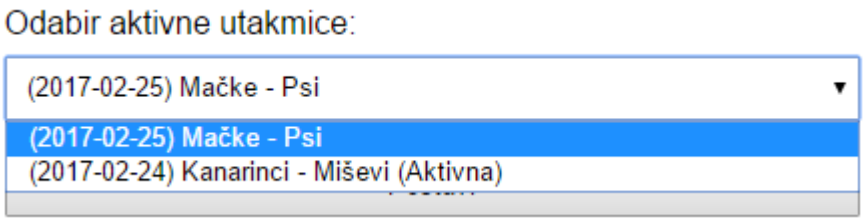
Slika 4.17. Padajući izbornik za brisanje ekipa

Iz slike 4.16. je također vidljivo da je na stranici omogućen rad s kreiranim utakmicama. Pomoću padajućeg izbornika sa slike 4.18. moguće je obrisati kreiranu utakmicu ukoliko je kod kreiranja došlo do nekakve pogreške ili je utakmica jednostavno otkazana.



Slika 4.18. Padajući izbornik za brisanje utakmice

I kao posljednja funkcija administratorske stranice je odabir aktivne utakmice. Postavljanje aktivne utakmice odrađuje se pomoću padajućeg izbornika sa slike 4.19. Postavljanje aktivne utakmice je izrazito važno zato što se rezultat koji se šalje sa Arduino razvojnog sustava sprema u onu utakmicu koja je aktivna. Slovom X označena je aktivna utakmica.



Slika 4.19. Padajući izbornik za odabir aktivne utakmice

5. ZAKLJUČAK

Primarni cilj ovog diplomskog rada koji je bio napraviti sustav pomoću mikroupravljača koji će automatski brojati golove te ih ispisivati na semaforu kao i na internet stranici. Zadatak je uspješno izvršen budući da svi dijelovi sustava rade kao jedna cjelina.

Prilikom izrade projektnog zadatka došlo je do nekoliko problema i zastoja budući da same komponente nisu visoke kvalitete, a može ih se oštetiti i vlastitom nepažnjom ako se dovede preveliki napon ili pokidaju nožice pinova koje su izrazito mekane.

Radom na ovom projektnom zadatku stekle su se vještine rada u Arduino razvojnom okruženju te izradi internet stranice. Izradu zadataka znatno je olakšalo to što su sve korištene tehnologije otvorenog tipa, pogotovo Arduino koji ima jako veliku zajednicu koja je svakim danom sve veća.

Kao i svugdje, tako i u ovom radu ima mjesta za poboljšanja. Neke od mogućnosti poboljšanja su da se ispis na LED matričnom pokazivaču ne odrađuje preko objekta, kreiranje animacije nakon zabijenog pogodtka na LED matričnom pokazivaču, praćenje rezultata u živo (*engl. LiveScore*) i druge.

LITERATURA

[1] Stranica o stolnom nogometu, veljača 2017.
http://www.foosballsoccer.com/

[2] P.Zenzerović, „Arduino kroz jednostavne primjere“, Hrvatska zajednica tehničke kulture, II izdanje- Zagreb 2014.
www.paolozenzerovic.info/hr/arduino-knjiga/

[3] Službena stranica Arduino platforme, veljača 2017.
www.arduino.cc

[4] ESP8266 forum zajednica, veljača 2017.
www.esp8266.com

[5] I. Holjevac, Klijentske Web tehnologije, diplomski rad-Varaždin 2010.
http://www.irenaholjevac.com/dipl/, pristup ostvaren 15.06.2016.

[6] J.Duckett, „HTML & CSS Design and Build Websiter“, John Wiley & Sons, Inc., 1st edition- SAD 2011.

[7] K. Yank, "Build Your Own Database Driven Web Site Using PHP & MySQL" , SitePoint Pty. Ltd., 4th edition- SAD, 2009.

[8] M.Varga, „Baze podataka – konceptualno, logičko fizičko modeliranje podataka“, Društvo za razvoj informacijske pismenosti (DRIP), I izdanje, Zagreb 1994.

[9] K.Cardwall, „Building Virtual Pentesting Labs for Advanced Penetration Testing“, Packt Publishing, 1st edition, UK 2014.

[10] Philips Semiconductirs I^2C bus specification, Version 2.1, veljača 2017.
www.nxp.com/acrobat_download/literature/9398/39340011.pdf

[11] Arduino Playground Arduino forum zajednica, veljača 2017.
www.playground.arduino.cc

[12] Instructables Arduino forum zajednica, veljača 2017.
www.instructables.com

SAŽETAK

Naslov: Sustav za unapređenje stolnog nogometa primjenom mikorupravljačke i Internet tehnologije

Sažetak: U ovom radu predstavljen je rad sa Arduino razvojnim okruženjem i Internet tehnologijama kroz izradu sustava za praćenje rezultata kod stolnog nogometa. Prilikom razrade rada potrebno je bilo uskladiti različite tehnologije i podsustave kako bi radili kao jedna cjelina. U radu se koriste dvije različite tehnologije, te su detaljno opisane njihove tehničke specifikacije i način na koji su prilagođeni za rad. Arduino razvojno okruženje ima izrazito veliki broj perifernih uređaja, a samo neki od njih su korišteni za izradu ovog rada. Nakon opisa korištenih tehnologija, mikro upravljača i njihovih perifernih uređaja, kroz nekoliko odlomaka i slika prikazan je način rada pojedinog dijela sustava za praćenje rezultata, kao i njega cjelokupna shema. Kroz izradu samog rada stečeno je osnovno poznavanje rada u Arduino razvojnom okruženju i tehnologijama za izradu Internet stranice.

Ključne riječi: Arduino razvojni sustav, Internet tehnologije, mikro upravljači, komunikacija, povezivanje.

ABSTRACT

Title: System improvement for Table Foosball application of microcontrollers and Web technology.

Abstract: This paper presents work with Arduino development environment and Web technology thrue making a system for score results in a game Table Foosball. The main problem in making this project was working with different technologies and make them work together as one system. The first part of the paper contains a detailed description of the technical specification and method of their linkage. Arduino development environment contains a great number of peripherals, but only a few of them are used in making of this project. After this, next paragraphs with pictures contains detail presentation of how the system works. While working on this project I acquired the knowledge of Arduino development environment and Web technology.

Keywords: Arduino development environment, Web technology, microcontroller, communication, connection.

ŽIVOTOPIS

Mate Delaš je rođen 11.01.1993. godine u Vinkovcima. Osnovnoškolsko obrazovanje stekao je u Osnovnoj školi Matije Gubec u Jarmini. Nakon završene osnovne škole 2007. godine upisuje Tehničku školu Ruđera Boškovića u Vinkovcima gdje stječe zvanje arhitektonskog tehničara. Po završetku srednje škole upisuje Elektrotehnički fakultet u Osijeku, preddiplomski studij, smjer računarstvo. 2014. godine završava preddiplomski studij i stječe zvanje sveučilišnog prvostupnika inženjera računarstva. Iste godine upisuje diplomski studij na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer procesno računarstvo. Od listopada 2016. godine radi u Hrvatskom Telekomu kao Bussines Service Customer Support.

Potpis:

_____