

## Projektni zadatak iz Ugradbenih Sistema

Tema:

**5x (8x8) Matrični displej**

12.05.2016

Sarajevo

Grupa 2

Tim 003

Edin Begić

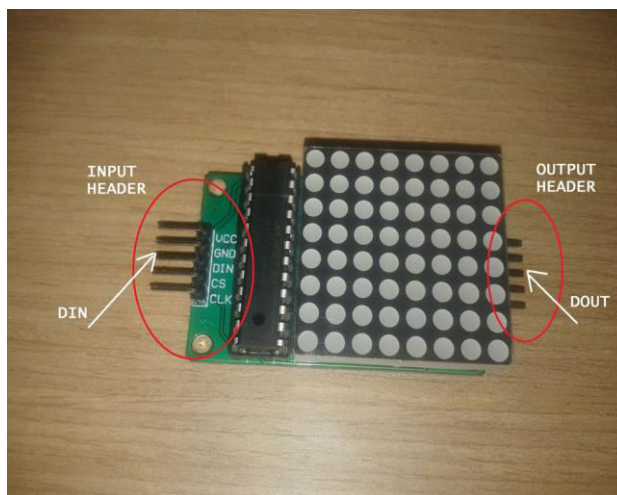
Emir Bećirović

## Sadržaj:

Šema spajanja:.....	3
Specifikacija: .....	5
Razrada:.....	6
Dnevnik rada:.....	10

## Šema spajanja:

Za uspješno spajanje svih komponenti našeg projekta potreban nam je matador(po mogućnosti više njih kako bi mogli serijski spojiti sve displeje), 5 LED matrix displeja s ugrađenim MAX7219 driverom, potencijometar, te LPC1114ETF mikrokontroler.



Serijski spajamo 5 displeja, tako da strana na kojoj se nalazi DIN pin bude uključena direktno na matador. Pošto je podržano Daisy chain spajanje nad MAX7219 driverom, VCC 1. displeja spajamo sa VCC 2. displeja posredstvom žica (male/male), te VCC 2. sa VCC 3. displeja itd. To isto ćemo uraditi i za sve ostale pinove na displeju (GND,DIN,CLK).

Slijedeći korak jeste spojiti svaki CS od displeja direktno na mikrokontroler, na DigitalOut pinove.

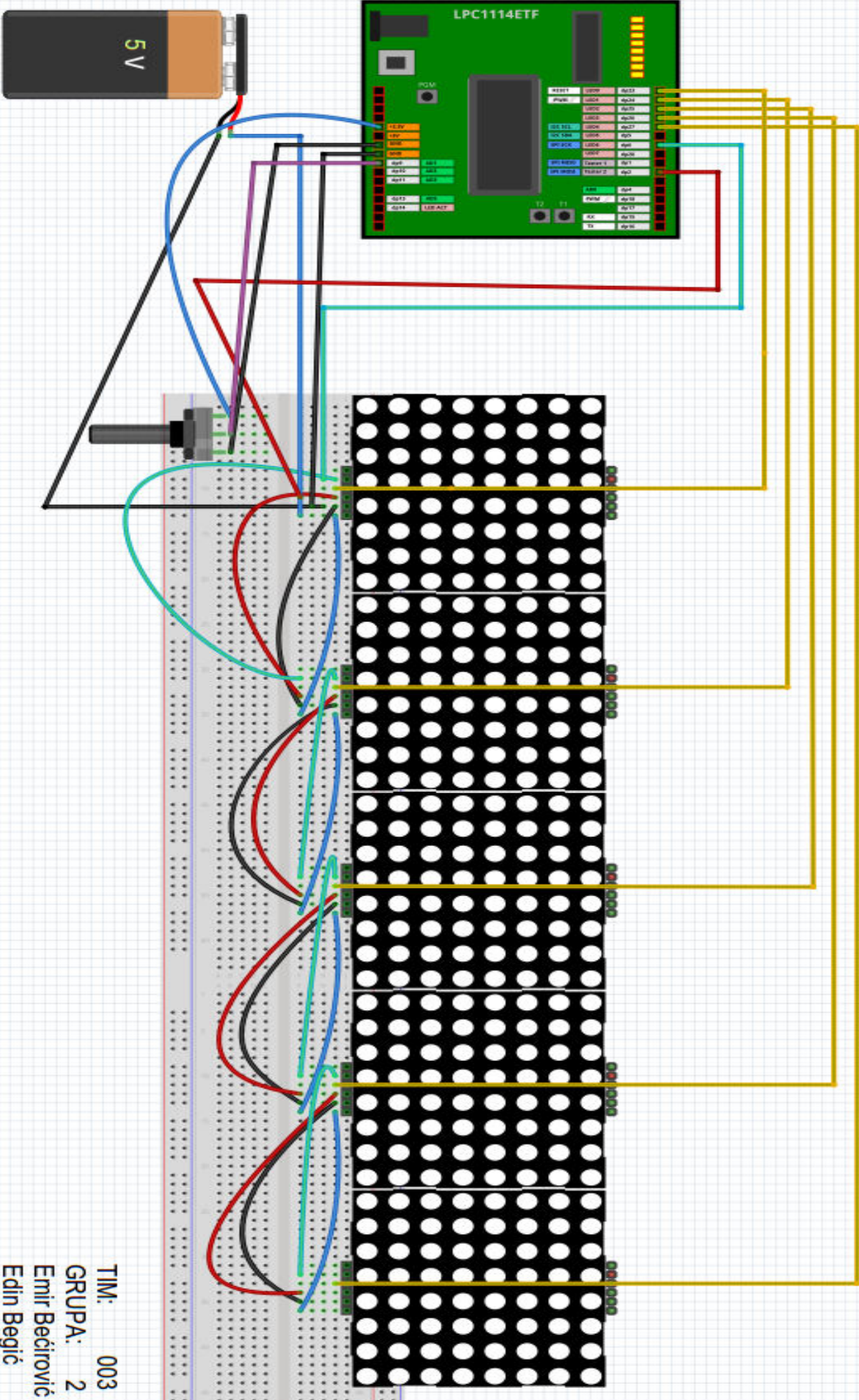
Sada je potrebno spojiti displeje sa našim mikrokontrolerom, a to ćemo postići na slijedeći način: VCC pin prvog displeja ćemo spojiti na 5V, koju dobijamo od naponske jedinice.

GND pin prvog displeja ćemo serijski spojiti sa ground-om mikrokontrolera, te zajednicki ground spojiti na naponsku jedinicu

DIN pin prvog displeja spajamo sa SPI MOSI pinom mikrokontrolera, što je u našem slučaju dp2

CLK pin prvog displeja spajamo sa SPI SCK pinom mikrokontrolera, što je u našem slučaju dp6

CS pin prvog displeja spajamo sa dp23 pinom mikrokontrolera, dp24 sa CS pinom drugog displeja, dp25 sa CS pinom trećeg displeja, dp26 sa CS pinom četvrtog displeja, dp27 sa CS pinom petog displeja. Preostalo nam je još spojiti potencijometar s LPC114ETF mikrokontrolerom; to ćemo postići tako što VCC pin potencijometra spojimo na 3.3 volta mikrokontrolera, GND pin na odgovarajući GND mikrokontrolera, te izlazni napon istog spajamo na dp9 pin. Kako bi to trebalo izgledati na kraju, prikazano je na slijedećoj šemi koja je izvedena u Fritzing okruženju:



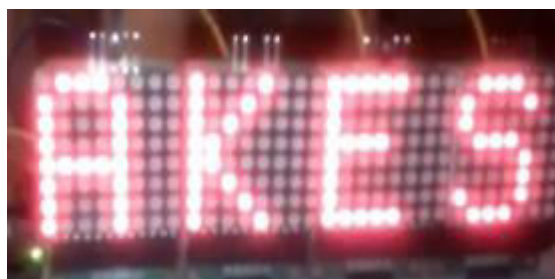
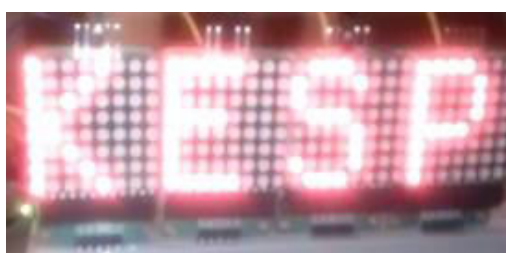
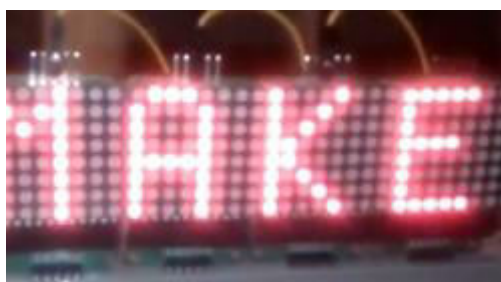


## Specifikacija:

Svrha ovog sistema je da omogući ispis teksta unijet preko tastature na 5 matričnih 8x8 displeja. Ispis teksta je moguć na dva načina : karakter po karakter I scroll mode (s desna na lijevo, pri čemu je brzina pomjeranja određena potenciometrom koji je spojen na sistem), te prikaz sata.

Što se tiče moda karakter po karakter na matričnom displeju na svakom od displeja se prikazuje slovo po slovo (prikaže se jedno slovo, pa se očisti slovo, pa se prikaze slijedeće slovo) dok se ne ispiše cijela rečenica, a potenciometrom povećavamo, odnosno smanjujemo, brzinu mijenjanja slova.

Kod scroll moda karakteri treba da se postepeno pomjeraju s desna na lijevo po displeju jedan za drugim dok se ne ispiše rečenica, s tim da potenciometrom povećavamo brzinu pomjeranja teksta.



Kako je sve ovo spojeno na računar, svime se upravlja pomoću konzole, tako unosom komande u terminal omogućen je: unos teksta, promjena moda, uključivanje prikaza teksta, isključivanje prikaza teksta, te prikaz sata, tj. trenutno vrijeme sistema, promjena jačine osvjetljenja displeja, mod kućne mašine.

## Razrada projektnog zadatka sa zaduženjima članova tima

Za ovaj projekat razvijene su glavne funkcije za regulaciju displeja, ispis stringova na displeje i prikaz vremena; one su smještene u MAX7219.h fajlu koji će se uključivati u main.cpp fajlu. main.cpp fajl će sadržavati main funkciju i funkcije za pozivanje menija, za unos stringa koji će se ispisivati na displeju.

### Main.cpp: [Termin 1 – Emir Bećirović]

void **postaviKasnjenje()** - korištenjem potencimetra povećavamo/smanjujemo kašnjenje prilikom promjene znaka na displeju (posebno je kašnjenje za scroll prikaza, a poseban za 'znak po znak');

void **unesiZnak()** – služi za unos i prikaz znaka, ako smo izabrali da želimo da ispišemo znak u meniju;

void **unesiString(char unos)** – služi za unos stringa koji će se ispisivati na displeje; Parametar 'unos' određuje u kojem modu će biti prikazan string;

void **prikazVremena()** - poziva metodu prikazVremena nad instancom klase Max7219;

void **promijeniOsvjetljenje(int pojacaj)** – u zavisnosti od vrijednosti parametra 'povacaj', poziva se metoda pojacajOsvjetljenjeDispleja ili smanjiOsvjetljenjeDispleja nad instancom klase Max7219;

void **konstantniUnosZnaka()** - direktno ispisuje znak na svim displejima, te očekuje novi unos

int **main()** – služi za ispis menija kojim se upravlja sistem;

### MAX7219.h: [Termin 2 – Edin Begić]

**Max7219**(PinName Pin\_mosi, PinName Pin\_miso, PinName Pin\_sclk, PinName Pin\_cs0, PinName Pin\_cs1, PinName Pin\_cs2, PinName Pin\_cs3, PinName Pin\_cs4) – konstruktor sa pinovima za displeje;

**~Max7219()** - destruktor klase (briše attribute spi, cs0, cs1, cs2, cs3, cs4);

void **upisiURegistar(int registar, int vrijednost, int redni\_broj\_uredjaja)** - Direktan upis u registre MAX7219 drivera, kako bi mogli gasiti/paliti određeni broj dioda. redni\_broj\_uredjaja je cijeli broj koji predstavlja displej koji manipuliramo;

void **ocistiDisplej(int redni\_broj\_uredjaja)** – funkcija koja čisti jedan displej (gasi sve diode sa jednog displeja);

void **ocistiDispleje()** – funkcija koja čisti sve displeje (gasi sve upaljene diode sa svih displeja);

void **upisiZnakURegistre(const uint8\_t\* pok, int redni\_broj\_uredjaja)** - pok je pokazivac na proširenu ascii tabelu (255 znakova) predstavljenu kao niz, gdje svaki znak zauzima 8 polja u nizu; upisuje cijeli znak na jedan od displeja;

void **prikaziVektor**(vector<uint8\_t>&v, int redni\_broj\_uredjaja) – Vektor v predstavlja buffer sa podacima koji se ispisuju na jednom displeju; ova metoda je korištena prilikom prikaza stringova scroll opcijom;

void **pomocnaShiftKolone**(int shift\_poz, int& vrijednost) – pomoćna funkcija prilikom određivanja bita koji se pomjera sa desna na lijevo u displeju (scroll mod);

void **popuniSveVektorePoKolonama**(vector<uint8\_t>&v4, vector<uint8\_t>&v3, vector<uint8\_t>&v2, vector<uint8\_t>&v1, vector<uint8\_t>&v0, int shift\_poz, char znak) - vektori su bufferi sa podacima koji se ispisuju scroll modom zdesna na displejima, shift\_poz je cijeli broj koji govori koji bit od 8-bitnog znaka se u tom trenutku prenosi;

void **displejInicijalizacija**() - Prilikom paljenja displeja, potrebno je upisati određene vrijednosti u pojedine registre kako bi mogli uspješno manipulirati driverom;

void **prikaziZnak**(char c, int redni\_broj\_uredjaja) – funkcija za prikaz jednog znaka na displeju;

void **prikaziString**(const string& s) – funkcija za prikaz stringa u modu karakter po karakter na svim displejima;

void **prikaziStringScroll**(string& s) – funkcija za prikaz stringa u scrollable modu preko svih displeja;

void **postaviKasnjenjeZnakPoZnak**(float t) – Postavlja vrijednost kašnjenja prilikom ispisa znakova pojedinačno na displejima;

void **postaviKasnjenjeScroll**(float t) – Postavlja vrijednost kašnjenja prilikom shiftanja znakova u scrollable modu;

void **prikazVremena**() - služi za ispis trenutnog vremena (računara) na displeje;

void **smanjiOsvjetljenjeDispleja**() - smanjuje vrijednost atributa 'osvjetljenje' te istu vrijednost upisuje u intensity mode registar, koji je zadužen za jačinu osvjetljenja dioda

void **pojacakOsvjetljenjeDispleja**() - analogno prethodnoj funkciji, samo se sada vrijednost 'osvjetljenje'-a povećava

### MAX7219.h: [Termin 3 – Emir Bećirović]

void **dodajIntUString**(int broj, string &s) – konvertuje određeni cijeli broj u string formatiran za upotrebu kod prikaza digitalnog sata

void **ocisti**() -javna metoda koja poziva privatnu metodu ocistiDispleje(). Potrebna prilikom ispisa znakova u modu kućace mašine.

### MAX7219.h: [Termin 3 – Edin Begić]

void **popuniSveVektorePoRedovima**(vector<uint8\_t>&v4,  
vector<uint8\_t>&v3,vector<uint8\_t>&v2,  
vector<uint8\_t>&v1,vector<uint8\_t>&v0, int shift\_poz, char znak) -  
Dodatna varijanta shiftanja podatadaka na displejima; ovaj put po redovima

void **alocirajSPI**(int redni\_broj) – da bi uspješno kontrolisali sve displeje tokom rada, potrebno je pri svakoj promjeni znaka ponovo alocirati pokazivac na SPI, što je implementirano u ovoj funkciji

void **deallocirajSPI**(int redni\_broj) – deallocira trenutni pokazivac na SPI



Main
display : Max7219 ticker : Ticker pc : Serial unos : String tekst : String kasnjenjeScroll : float kasnjenjeZnakPoznak : float potenciometar : AnalogIn displayZaKonstantniUnos : int postavkiKasnjenje() : void konstantniUnosZnak() : void unesiZnak() : void unesiString(unos : char) : void prikazVremena() : void promijeniOsjetljenje(pojacaj : int) : void meni() : void

<<enumeration>>
Max7219
-dp2 -dp1 -dp6 -dp23 -dp24 -dp25 -dp26 -dp27

<<enumeration>>
Serial
-USBTX -USB RX

<<enumeration>>
AnalogIn
-dp9

Max7219
-spi : SPI* -cs0 : DigitalOut* -cs1 : DigitalOut* -cs2 : DigitalOut* -cs3 : DigitalOut* -cs4 : DigitalOut* -pin_mosi : PinName -pin_miso : PinName -pin_sclk : PinName -kasnjenjeZnakPoznak : float -kasnjenjeScroll : float -osjetljenje : int +ocistiDisplay(redni_broj_displaya : int) : void +ocistiDisplay() : void +upisiURegistar(registar : int, vrijednost : int, redni_broj_uredjaja : int) : void +upisiZnakURegistre(pok : const uint_8 *, redni_broj_uredjaja : int) : void +prikaziVektor(v : vector<uint8_t> &, redni_broj_uredjaja : int) : void +popuniSveVektorePoKolonomama(v4 : vector<uint8_t> &, v3 : vector<uint8_t> &, v2 : vector<uint8_t> &, v1 : vector<uint8_t> &, v0 : vector<uint8_t> &, shift_poz : int, znak : char) : void +popuniSveVektorePoRedovimal(v4 : vector<uint8_t> &, v3 : vector<uint8_t> &, v2 : vector<uint8_t> &, v1 : vector<uint8_t> &, v0 : vector<uint8_t> &, shift_poz : int, znak : char) : void +pomocnaShiftKolone(maska : int, shift_poz : int &) : void +dealocirajSPI(redni_broj : int) : void +alocirajSPI(redni_broj : int) : void +dodajIntUString(broj : int, s : String &) : void +Max7219(Pin_mosi : PinName, Pin_miso : PinName, Pin_sclk : PinName, Pin_cs0 : PinName, Pin_cs1 : PinName, Pin_cs2 : PinName, Pin_cs3 : PinName, Pin_cs4 : PinName) +~Max7219() +displejInicijalizacija() : void +prikaziZnak(c : char, redni_broj_uredjaja : int) : void +prikaziString(s : const String &) : void +prikaziStringScroll(s : String &) : void +postavkiKasnjenjeZnakPoznak(t : float) : void +postavkiKasnjenjeScroll(t : float) : void +prikazVremena() : void +smanjiOsjetljenjeDisplay() : void +pojacajOsjetljenjeDisplay() : void +ocisti() : void

TIM: 003  
GRUPA: 2  
Emir Bećirović  
Edin Begić

## Dnevnik rada – Edin Begić

11.05.2016. 20:00 – 23:00	Čitao specifikaciju drivera MAX7219, te popravio do sad nacrtanu šemu spajanja, napravio max7219.cpp biblioteku po uzoru na biblioteku drivera MAX7219, te funkcije za ispis znaka prikaziZnak(char c, int redni_broj_uredjaja) i funkciju za mod ispisa znak po znak void prikaziString(const string& s), te pomoćne funkcije kao što su konstruktor i destructor za driver, displejInicijalizacija(), ocistiDispleje(), upisiURegistar(int registar, int vrijednost), upisiZnakURegistre(const uint8_t* pok), displejInicijalizacija() , funkcije za modifikaciju kašnjenja promjene znaka na displeju (postaviKasnjenjeZnakPoZnak I postaviKasnjenjeScroll).
12.05.2016. 17:00 – 19:00	Uspješno spojili šemu i testirali određene funkcionalnosti na jednom matričnom displeju.
15.05.2016. 13:00 – 16:00	Ispravljaao kod za rad sa više matričnih displeja, doradio potrebne funkcije koje nisu bile implementirane do sada. Implementirao funkciju PrikaziStringScroll I to koristeći binarno shifatanje podataka po kolonama.
19.05.2016. 17:00 – 19:00	Testirali do sada napisani kod za rad na dva matrična displeja, pri tome dodali funkciju prikazVremena(). Javio se problem prekoračenja dostupne flash memorije zbog "time.h" biblioteke.
19.05.2016. 21:00 – 23:00	Izvršio optimizaciju koda, kako bi mogao iskoristiti metode iz "time.h" biblioteke; skratio ASCII tabelu za prikaz znakova(na 127 znakova)
20.05.2016. 14:00 – 16:00	Dodao funkciju AlocirajSPI(int redni_broj), koja inicijalizira displej pri svakom novom ispisu znaka.
21.05.2016. 13:45 – 15:45	Uspješno testirali većinu funkcija na dva matrična displeja, a zatim i na svih pet displeja.
25.05.2016. 20:00 – 1:00(26.05.2016.)	Implementirao funkciju PopuniVektorePoRedovima kako bi omogućio horizontalan prikaz stringa u scroll modu (do sada ispis bio vertikalni).
26.05.2016. 17:00 – 19:00	Testirali funkcionalnost projekta, ispravljali sitne greške, demonstrirali asistentu funkcionalnost projekta, te snimili funkcionalnost istog kao video.
Ukupno: 23 sati	

## Dnevnik rada – Emir Bećirović

11.05.2016. 20:00 – 23:00	Nacrtao šemu spajanja u fritzing okruženju, napisao detaljnu specifikaciju zadatka, našao pomoćne materijale objavljene od strane Maxim firme za driver MAX7219, te primjere kako aktivirati cs signal na datasheetu, napisao main funkciju sa glavnim menijem.
12.05.2016. 17:00 – 19:00	Uspješno spojili šemu i testirali određene funkcionalnosti na jednom matričnom displeju.
15.05.2016. 20:00 – 23:00	Napisao razradu projektnog zadatka.
19.05.2016. 17:00 – 19:00	Testirali do sada napisani kod za rad na dva matrična displeja, pri tome dodali funkciju prikazVremena(). Javio se problem prekoračenja dostupne flash memorije zbog "time.h" biblioteke.
20.05.2016. 14:00 – 16:00	Sa asistentom ispravljali greske u pojedinim funkcijama
21.05.2016. 13:45 – 15:45	Uspješno testirali većinu funkcija na dva matrična displeja, a zatim i na svih pet displeja.
25.05.2016. 20:00 – 22:00	Doradio main funkciju za ispis glavnog menija
26.05.2016. 17:00 – 19:00	Testirali funkcionalnost projekta, ispravljali sitne greške, demonstrirali asistentu funkcionalnost projekta, te snimili funkcionalnost istog kao video.
Ukupno: 20 sati	