SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4240

UPRAVLJANJE SEFOM POMOĆU MIKROKONTROLERA MSP430

Davor Štefok

Zagreb, lipanj 2015.

Sadržaj

Uvod

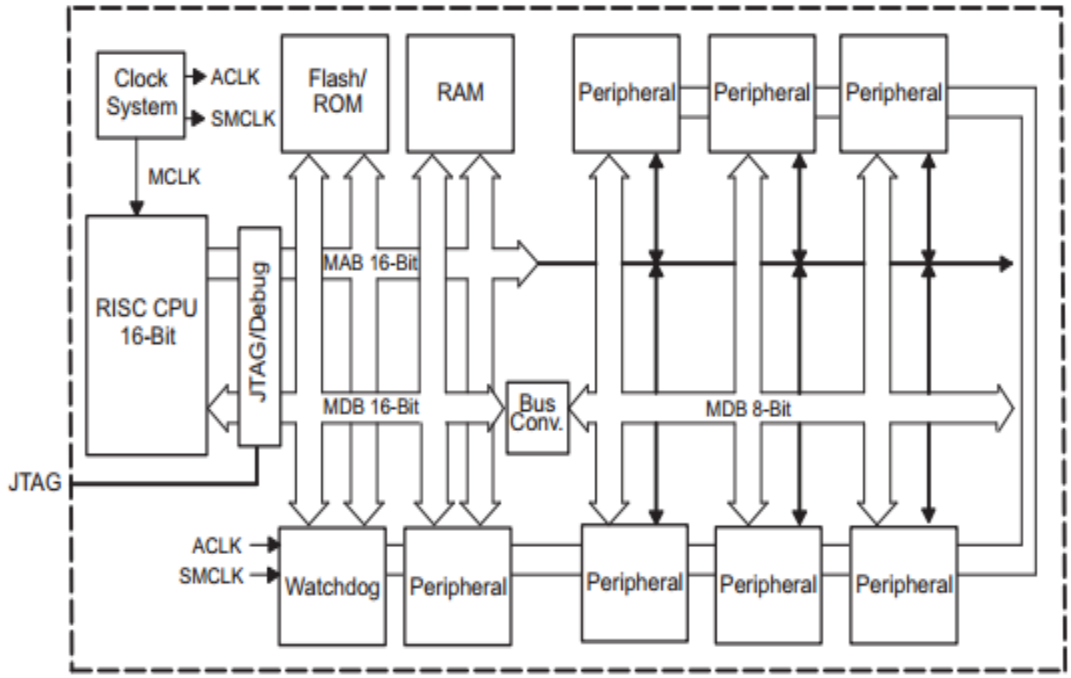
Mikrokontroler je elektronički uređaj koji se koristi u ugradbenim računalnim sustavima. Ugradbeni računalni sustavi imaju široku paletu primjene. Razvitkom tehnologije, ugradbeni računalni sustavi postali su sastavni dio gotovo svakog elektroničkog uređaja.

U ovom radu opisana je programska potpora i izvedba upravljanja sefa pomoću mikrokontrolera iz porodice MSP430G2, te sam mikrokontroler. Također opisane su periferne komponente i njihove funkcije. Opisan je i tijek razvoja, te razlika u korištenju 14-pinskog i 20-pinskog mikrokontrolera.

1. MSP430G2211

1.1 Arhitektura

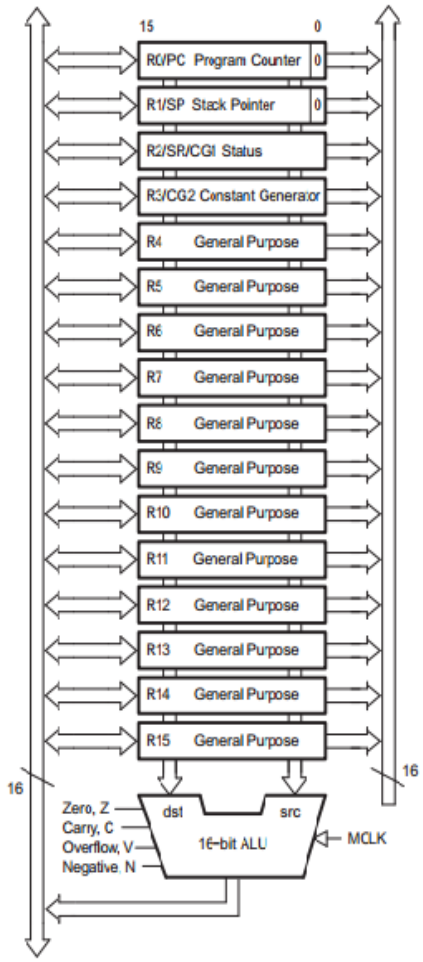
MSP430 sadrži 16-bitni RISC procesor, periferije mikrokontrolera, generator takta te dvije sabirnice [Memory Adress Bus(MAB) i Memory Data Bus(MBD)]. Koristeći von Neumannovu arhitekturu obavlja se povezivanje procesora, generatora takta i periferije. Odlike porodice sklopova MSP430x2xx je mala potrošnja energije i mogućnost dugotrajnog rada na baterijskom napajanju (ovisno o izvedbi). Podatkovna memorija ima kapacitet od 128 bajtova, dok programska memorija ima kapacitet od 2 kilobajta. U mikrokontroleru MSP430G2211 koristi se jedan skup ulazno-izlaznih priključaka koji služi za priključivanje mikrokontrolera na predviđenje periferije i upravljanje ili čitanje iz njih. Razvojni sustav ima ukupno 14 ulazno-izlaznih priključaka te se mogu omogućiti s odgovarajućim mikrokontorlerom.



Slika 1.1 Arhitektura MSP430 porodice sklopova

1.2 Središnja procesorska jedinica

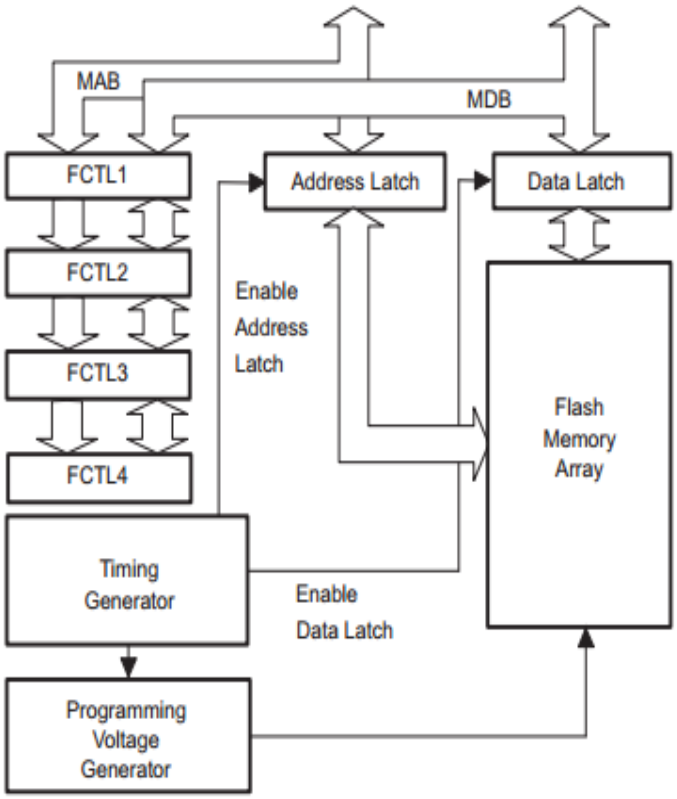
Središnja procesorska jedinica (*CPU*) je dizajnirana u skladu s modernim tehnikama programiranja kao što su naprimjer tablično procesuiranje i korištenje programskih jezika više razine, poput C-a. RISC arhitektura sadrži sedam adresna izvorna operanada i četiri adresna odredišna operanda. Procesor sadrži 16 integriranih registara, od kojih registri R0, R1, R2 i R3 imaju posebne namjene, dok ostali registri služe za opće namjene. Registar R0 služi kao programski brojač, R1 kao pokazivač stoga, R2 kao statusni registar, a R3 kao stalni generator.



Slika 1.2 Blok dijagram *CPU*-a

1.3 Flash memorija

Početna adresa *Flash* memorija ovisi o njegovoj dostupnosti i varira ovisno o uređaju. Krajnja adresa *Flash* memorije je 0x0FFFF za uređaje s manje od 60 kilobajta *Flash* memorije, što uključuje i korišteni mikrokontroler MSP430G2211 koji sadrži 2 kilobajta *Flash*-a. Također je takva vrsta memorije zamjenila EEPROM u brojnim aplikacijama zbog jednostavnosti jer daje korisniku mogućnost višestrukog programiranja i brisanja. Memorija je podjeljenja na više segmenata, te kod brisanja se jedino mogu brisati određeni segmenti, dok kod pisanja se u memoriju mogu unositi podaci reda veličine bita i veći. Minimalno napajanje tokom pisanja ili brisanja memorije je 2.2 V i ako vrijednost napajanja padne ispod navedene razine, rezultat odabarne operacije biti će nepredvidiv.



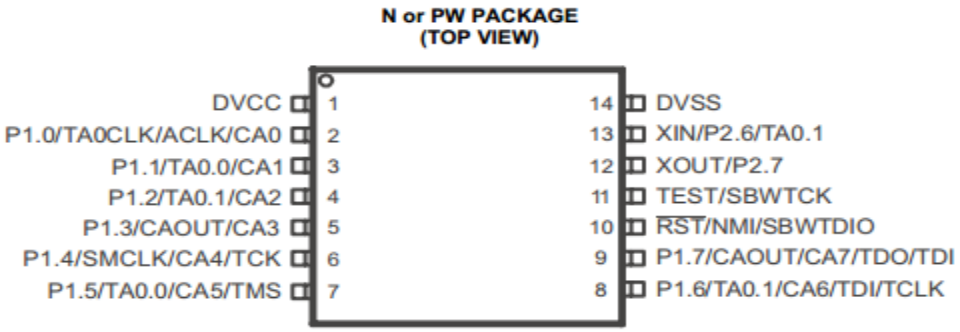
Slika 1.3 Blok dijagram *Flash* memorije

1.4 On-board Emulator

On-board emulator omogućava programiranje i služi za *debugging* podržanih MSP430 uređaja. On nudi više opcija koje su omogućene preko dvožičanog JTAG sučelja nazvanog *Spy-Bi-Wire*. Unutar *datasheeta* MSP-EXP430G2 navedeni su mikrokontorleri koji su podržani sa strane *on-board* emulatora: F20xx, F21x2, F22x2, G2x01, G2x11, G2x21, G2x31 i G2x53. Pošto je ovaj završni rad u jednoj fazi svoga razvoja zamišljen uz izvedbu s 20-pinskim mikrokontrolerom, korišten je i mikrokontroler G2353, no zbog nemogućnosti uspostave veze između razvojnog alata programske podrške (*Code Composer Studio-a)* i navedenog mikrokontrolera odustalo se od takvog rješenja.

1.5 Vanjski priključci

Korisnik vanjske priključke mikrokontrolera može definirati kao priključke posebne funkcije ili kao ulazno-izlazne priključne. Dva priključka služe za napajanje i masu, a jedan priključak služi za reset sustava. Priključci XIN i XOUT mogu biti ulazno-izlazni (P2.6 i P2.7) isključivo u varijanti sa 20-pinskim mikrokontrolerom koji bi bio proširen sa šest ulazno-izlaznih priključaka (P1.0-P1.5), dok u slučaju 14-pinskog mikrokontrolera služe isključivo za generiranje signala takta tako da se izvana priključi kristal kvarca na navedene priključke.



Slika 1.4 Shema priključaka mikrokontrolera porodice MSP430G2x11

1.6 Digitalni ulazno-izlazni priključci

MSP430G2211 ima osam digitalnih ulazno-izlaznih priključaka (P1.0-P1.7). Njima se upravlja preko programske podrške načinima objašnjenim u nastavku. Objašnjeni su samo registri korišteni u programskoj podršci izrađenoj za ovaj rad.

1.6.1 Ulazni registar P1IN

Određuje stanje u kojem se nalazi ulazni signal na određenom pinu.

Bit=0: Ulazni signal je u niskoj razini

Bit=1: Ulazni signal je u visokoj razini

1.6.2 Izlazni registar P1OUT

Definira stanje u koje želimo postaviti izlazni signal na određenom pinu.

Bit=0: Izlazni signal je u niskoj razini

Bit=1: Izlazni signal je u visokoj razini

1.6.3 Registar smjera P1DIR

Služi za postavljanje digitalnog priključka u stanje ulaznog ili izlaznog priključka, odnosno definira smjer priključka.

Bit=0: Postavlja priključak u stanje ulaza

Bit=1: Postavlja priključak u stanje izlaza

1.7 Watchdog timer

Watchdog timer ili sklopovlje za detekciju pogrešnog rada služi kako bi se otkrila određenja smetnja kojoj je izložen elektronički sklop. Posljedica tih smetnji, poput šiljaka u napajanju, je nepravilan rad sklopa.

Registar WDTCTL koristi se kako bi se odredilo što činiti sa *watchdog timer*-om. Sastoji se od 16 bitova, od kojih je samo prvih 8 kontrolnih, dok drugih 8 služe radi sigurnosti, kako naš kod ne bi mijenjao WDT+ (*watchdog timer)*. Registar također sadržava kontrolne bitove za priključak RST. U navedenom radu registar WDTCTL se koristi isključivo na početku glavnog programa kako bi se onemogućio *watchdog* reset tokom izvođenja programa naredbom:

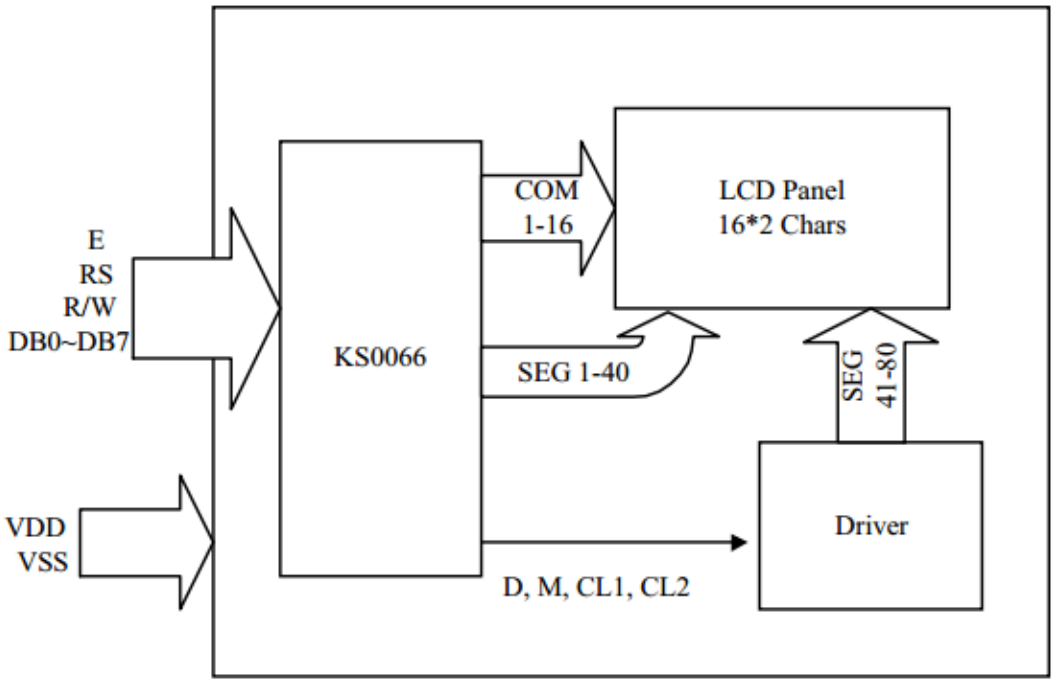
```
WDTCTL = WDTPW + WDTHOLD;
```

WDTPW (0x5A00) služi za onemogućavanje mijenjanja *watchdog timer*-a našim kodom. WDTHOLD služi kako bi isključili *watchdog timer*. Obje vrijednosti definirane su u *header*-u korištenog mikrokontrolera.

2 LCD prikaznik

2.1. Pogramabilni priključci

Dimenzije korištenog LCD prikaznika su 16x2, 16 simbola po redu u 2 reda. LCD prikaznik koristi mikrokontroler KS0066.



Slika 2.1 Blok dijagram LCD prikaznika

Postoje dvije inačice kako možemo prenjeti podatke ili naredbu na navedeni mikrokontroler, pa posljedično i na prikaznik. Prva inačica je 8-bitni prijenos u kojem se koriste svi podatkovni priključci LCD prikaznika (DB0-DB7). Druga inačica je 4-bitni prijenos koji podatak „dijeli“ u *nibble*-ove i šalje podatke u 2 ciklusa na mikrokontroler. Viših 4 bita se šalje u prvom ciklusu, zatim nižih 4 bita u

drugom ciklusu. U tom slučaju koriste se isključivo DB4-DB7 podatkovni priključci LCD-a. Ti priključci spajaju se na podatkovne izlaze mikrokontrolera P1.0-P1.3.

E priključak (*enable*) služi za čitanje podatkovne sabirnice i sam prikaz na LCD-u. RS priključak služi kako bi se odredilo šalje li se na LCD naredba ili podatak kojeg želimo prikazati. R/W priključak služi kako odredio smjer kretanja podataka, odnosno čitamo li sa LCD-a ili pišemo na LCD. U ovom konkretnom radu čitanje s LCD prikaznika nije bilo od važnosti pa je taj ulaz postavljen u nulu, odnosno u konstantno stanje pisanja na LCD.

DIGIT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1 LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
2 LINE	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

Tablica 2.1 DDRAM adrese LCD prikaznika

2.2 Osnovne naredbe LCD prikaznika

Postoji set osnovnih naredbi koji se koristi za upravljanje LCD-om.

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Tablica 2.2 Brisanje svega sa LCD-a

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	1/0

Slika 2.3. postavljanje pokaznika (*cursor*) u početno stanje (prvi red, prvi stupac)

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

Slika 2.4 Određivanje smjera pokaznika

I/D=1: pokaznik se pomakne za jedan u desno i poveća se Display Data RAM

I/D=0: pokaznik se pomakne za jedan u lijevo i smanji se Display Data RAM

S=1: pomak pokaznika u desno

S=0: pomak pokaznika u lijevo

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Tablica 2.5 Uključivanje LCD-a

D=1: uključiti LCD

D=0: isključiti LCD

C=1: uključiti pokaznik

C=0: isključiti pokaznik

B=1: uključiti treptanje pokaznika

B=0: isključiti treptanje pokaznika

RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	1/0	1/0

Tablica 2.6 Aktivacija funkcijskog seta

DL=1: 8-bitno sučelje

DL=0: 4-bitno sučelje

N=1: dvolinijski prikaz

N=0: jednolinijski prikaz

F=1: 5x11 točkasti prikaz

F=0: 5x8 točkasti prikaz

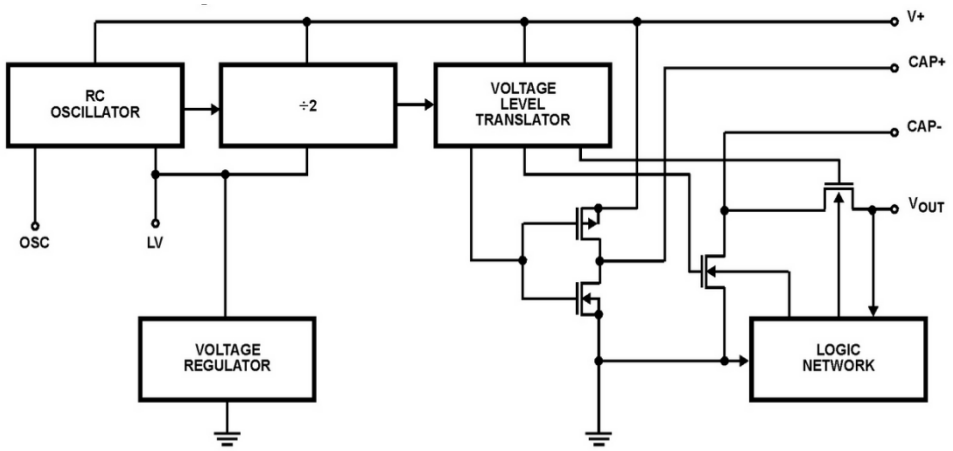
2.3 Napajanje LCD prikaznika

2.3.1 Priključci napajanja

Priključci LCD-a koji se direktno spajaju na mikrokontroler u svrhu napajanja u V_{DD} i V_{SS}. V_{DD} se napaja iz izvora napona MSP430G2211, odnosno u konkretnom slučaju sa USB porta. Iznos napona na priključku V_{DD} iznosi 3.3V. Priključak V_{SS} spojen je na GND priključak (masu) MSP430G2211. Priključak V_O određuje kontrast i ako bi imali napajanje od 5V ne bi bilo potrebno išta spajati na taj priključak jer bi razlika između V_{DD} i V_O bila točno 5V, što je tehnički zahtjev korištenog prikaznika. No kako tome nije slučaj potreban je izvor negativnog napona. Korišten je sklop ICL7660 koje je detaljnije opisan u sljedećem poglavlju. On ima zadaću davati negativan napon iznosa -3.3V. Kako bi taj napon došao do otprilike -1.6V, čime bi se zadovoljio uvjet razlike napajanja prikaznika i kontrasta od 5V, potrebno je preko naponskog dijelila pomoću 2 otpornika smanjiti taj napon. Prvotno su korištena 2 otpornika istog iznosa (10kΩ), što bi zadovoljavalo uvjete idealnom slučaju, no kontrast (odnosno napon na kojem je bio priključak kontrasta) nije bio zadovoljavajuć. Problem je riješen tako da se prvo -3.3V na jednom 10kΩ otporniku poveća na -1.6V, a zatim se spoj zaključi 20kΩ otpornikom koji je spojen na masu. Problem u napajanju LED *backlight*-a je taj što kada se preko 10Ω otpornika anoda spojila na izvor napajanja sklop ICL7660 je izgubio potrebnu snagu. Na anodu nije spojeno ništa, dok je na katodu *backlight*-a spojen izlaz iz sklopa ICL7660, naponske razine -3.3V. Izvedba je vidljiva na slikama 4.1 i 4.2.

2.3.2 ICL7660

ICL7660 je izvor napona koja se zasniva na CMOS spoju, koja pruža neke dodatne mogućnosti naspram prijašnjih sličnih komponenata. Ova komponenta korištena je u svrhu napajanja K (LED-) i V_O ulaza LCD prikaznika. Prikladan je jer generira negativan napon, invertirajući pozitivni ulazni napon, u spoju s 2 elektrolitska kondenzatora prikazana na slici 4.1 i 4.2. Prvotno su korišteni elektrolitski kondenzatori istog iznosa (10µF), ali zbog pada napona od 0.5V (naspram ulaznog) koji generira sam ICL7660 rezultati nisu bili zadovoljavajući.

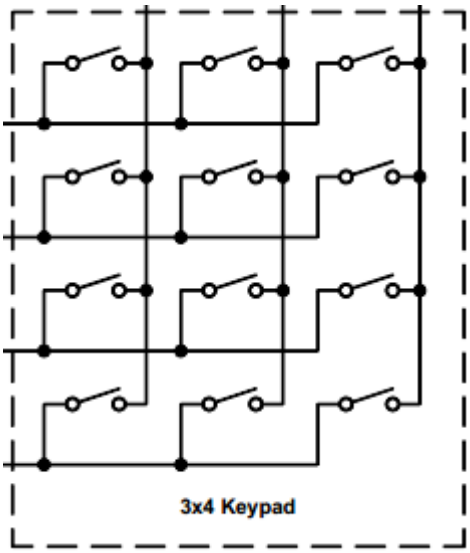


Slika 2.2 Shema ICL7660

3. 3x4 Tipkovnica

U ovom radu početno je zamišljeno kako će se koristiti puna funkcionalnost 3x4 tipkovnice, no zbog ograničenog broja ulazno-izlanih priključaka mikrokontrolera, te zvog nemogućnosti dobavljanja ispravnog 20-pinskog mikrokontrolera implementirana su samo 2 stupca tipkovnice, što pretvara ovu tipkovnicu u oblik 2x4.

Svi principi tipkovnice 3x4 mogu se primjeniti na tipkovnici 2x4.



Slika 3.1 Shema tipkovnice 3x4

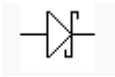
Kako bi tipkovnica bila u potpunosti funkcionalna, potrebno odrediti ulaze i izlaze mikrokontrolera spojene na tipkovnicu, te dodati dvije vrste generičkih elektroničkih komponenata.

3.1 Izvedba tipkovnice

3.1.1 Redovi Tipkovnice

Redovi tipkovnice spojeni su na podatkovne izlaze mikrokontrolera (P1.0-P1.4), ali se za razliku od LCD-a, čiji su ulazi s navedenim izlazima mikrokontrolera spojeni isključivo sa „žicom“, između redova tipkovnice i izlaza iz mikrokontrolera nalaze se diode kako bi struja tekla isključivo u jednom smjeru.

Razne diode se koriste pri izvedbi tipkovnica. Vrlo kompatibilna dioda za navedenu primjenu je Schottky dioda. Schottky dioda temelji svoj rad na ispravljačkom spoju metal-poluvodič. Glavna odlika navedene diode je da za razliku od silicijske diode ima manji napon koljena (oko 0.2 V) te je zbog oblika strujno-naposke karakteristike te kraćeg vremena oporavka izuzetno pogodno za primjenu u brzim sklopovima i u sklopovima koji koriste male razine napona, poput korištenog mikrokontrolera. Iskorištena je u trenutnom projektu kako bi se povećala pouzdanost korištenja tipkovnice.



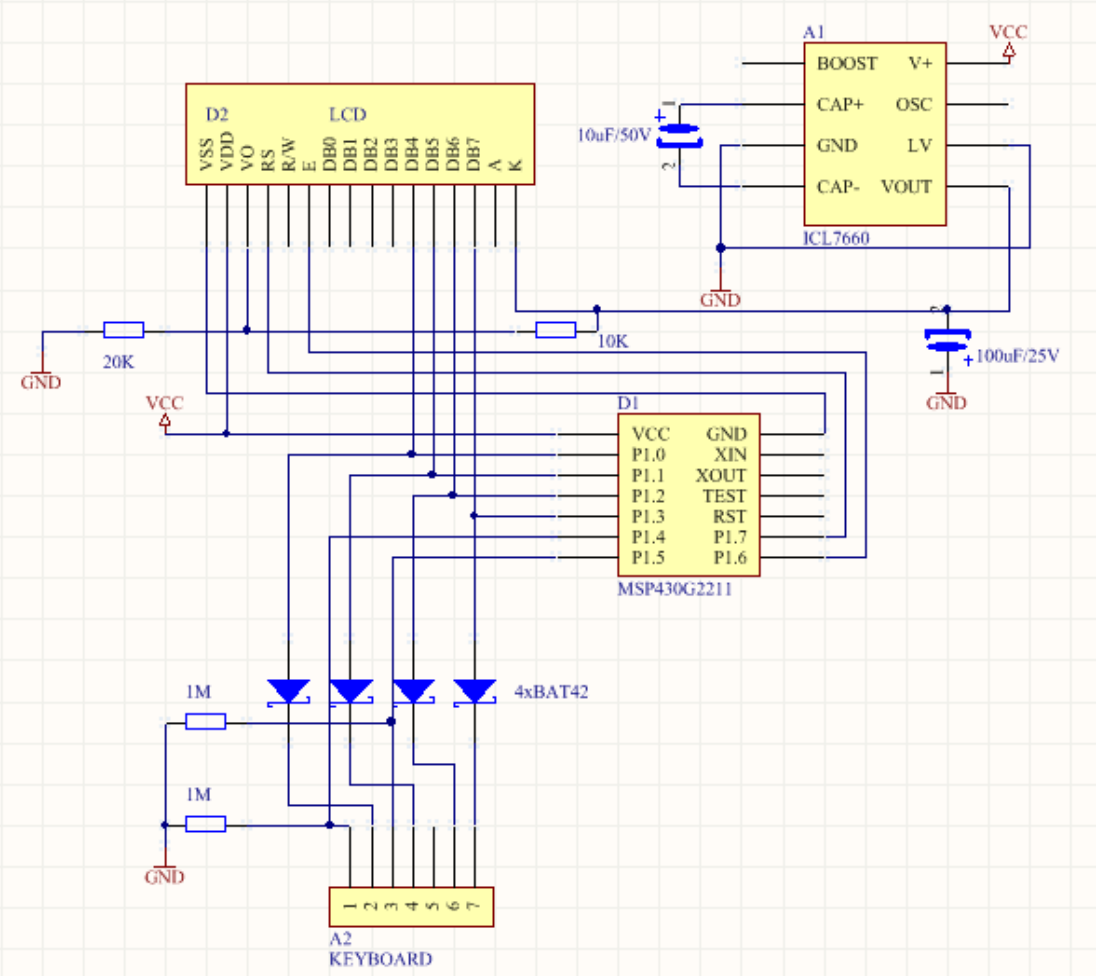
Slika 3.2 Oznaka Schottkyeve diode

3.1.2 Stupci tipkovnice

Stupci tipkovnice spojeni su na ulaze mikrokontrolera (P1.4-P1.5) i na *pull-down* otpornike u iznosu od 1MΩ. Uloga *pull-down* otpornika je da za vrijeme kada odovarajuća tipka na tipkovnici nije stisnuta drže ulaze mikrokontrolera u logičkoj nuli.

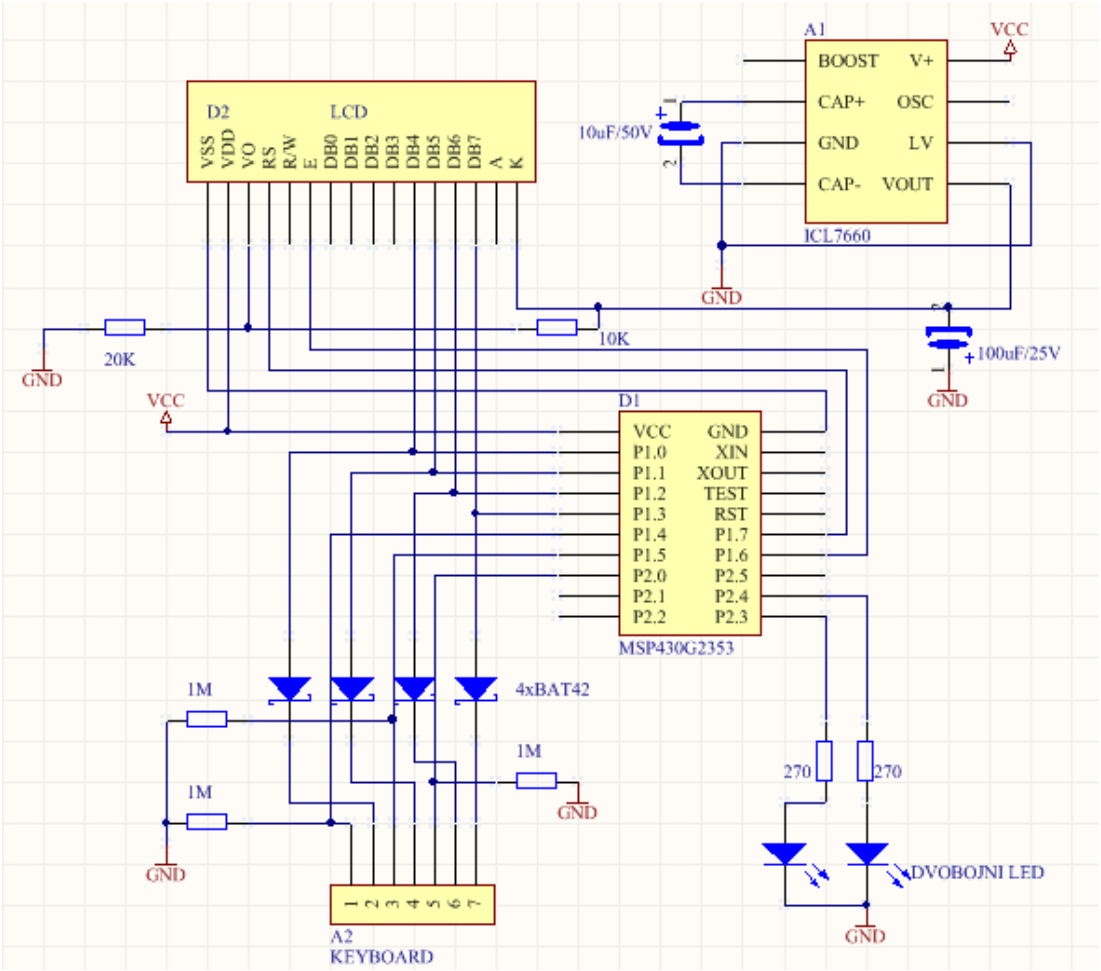
4. Električna shema

4.1 Realizirana shema



Slika 4.1 Shema s 14-pinskim mikrokontrolerom

4.2 Prvotno zamišljena shema



Slika 4.2 Shema s 20-pinskim mikrokontrolerom i dvobojnim LED-om kao izlaznim članom

U prvotno zamišljenoj verziji dvobojni led je služio kako bi označio točnost upisane lozinke.

5. Programska podrška

Programska podrška u potpunosti je izrađena u programu *Code Composer Studio v6,* izdanog od strane *Texas Instruments*-a. *Code Composer Studio* izuzetno je praktičan alat pri *debug*-iranju problema zbog toga što se kod može simulirati korak po korak, liniju po liniju. Istodobno mogu se isčitavati i stanja globalnih varijabli te registara mikrokontrolera, što je vrlo praktično pri analiziranju i traženju greške.

5.1 Programska podrška LCD prikaznika

Programska podrška LCD prikaznika izrađena je kroz više potprograma radi lakšeg korištenja istih. Svi potprogrami vezani uz upravljanje LCD-om uključeni su u *header* datoteku koja je uključena u glavni program.

Kao osnovni potprogram potrebno je istaknuti **lcdcmd**, koji je korišten i u inicijalizaciji LCD prikaznika.

```
void lcdcmd(unsigned char Data)
{
P1OUT &= ~RS;
P1OUT  &=~EN;
P1OUT &= 0xF0;
P1OUT |= ((Data >> 4) & 0x0F);
P1OUT  |=EN;
waitlcd(2);
P1OUT  &=~EN;
}
```

```
void lcdcmd(unsigned char Data)

{

P1OUT &= ~RS;

P1OUT  &=~EN;

P1OUT &= 0xF0;

P1OUT |= ((Data >> 4) & 0x0F);

P1OUT  |=EN;

waitlcd(2);

P1OUT  &=~EN;

}
```

RS je u logičkoj nuli radi toga što se šalje podatak koji je naredba, a ne podatak kojeg treba prikazati na zaslonu. Jedina razlika ovog potprograma od **lcdData** je ta što je u potprogramu u kojem šaljemo podatak RS postavljen u logičku jedinicu.

Princip slanja bitova na prikaznik je ista u oba programa. Bit EN (*enable)* je u logičkoj nuli sve dok se podatkovnoj sabirnici mikrokontrolera ne pojavi podatak kojeg želimo proslijediti na prikaznik. Prvo se donjih 4 bita izlaza P1 postavljaju u nulu kako bi se osiguralo da na njima nema neželjenih podataka. Za prijenos na pokaznik koristi se 4-bitna inačica. Na izlaze P1 se postavlja gornja četiri bita podataka, EN omogućuje čitanje i nakon toga se na isti način šalju i donja četiri bita.

Da bi ovakav način slanja bio moguć potrebno je inicijalizirati LCD prikaznik. Prikazani odsječak potprograma prikazuje baratanje osnovnim naredbama prikaznika prikazanim u tablicama 2.2-2.6. Potprogrami **lcdcmd** i **lcdinit** međusobno se nadopunjuju.

```
P1OUT  |=EN;
P1OUT  &=~EN;
lcdcmd(0x28);
waitlcd(250);
lcdcmd(0x0E);
waitlcd(250);
lcdcmd(0x01);
```

```
waitlcd(250);
lcdcmd(0x06);
waitlcd(250);
lcdcmd(0x80);
waitlcd(250);
```

Potprogrami **lcdcmd** redom rade sljedeće zadaće:

1. Postavlja duljinu podatka na 4 bita i prikazivanje u 2 reda
2. Uključuje prikaznik, pokazivač i treptanje pokazivača
3. Čisti cijeli prikaznik
4. Određuje smjer kretanje pokazivača (u desno)
5. Postavlja adresu RAM-a (nije navedeno u tablicama 2.2-2.6)

Ostali potprogrami iz **lcd16.h** datoteke su **waitlcd** (koji omogućava čekanje u milisekundama), **prints** (koji omogućava ispisivanje *stringova* podataka, praktičan kod ispisivanja predefiniranih stvari na prikaznik, poput „write password“), i **gotoXy**.

Posljednji potprogram koristi se kako bi se pozicionirao pokazivač na prikazniku na željeno mjesto.

```
void gotoXy(unsigned char
x,unsigned char y)
{
 if(x<40)
 {
  if(y) x |= 0x40;
  x |=0x80;
  lcdcmd(x);
  }
}
```

Gornja 4 bita poslana u **lcdcmd** potprogram određuju u kojem će se redu nalaziti, a donja četiri bita na kojem će se mjestu u određenom redu nalaziti pokaznik. Ograničenje gotovo svih LCD prikaznika ovakvog oblika je otprilike 40 znakova po redu, što znači da prikaz ne mora biti fiksan, već se do 40 znakova mogu istodobno kretati po jednom redu LCD prikaznika.

5.2 Programska podrška tipkovnice

Programska podrška tipkovnice izrađena je preko isključivo jednog potprograma koji čita tipke. Od ostalih potprograma korišten je jedino **waitlcd** kako bi se, pri promjeni stanja izlaza P1.0-P1.3, stabilizirao izlaz. Na početku kod je bio napravljen bez **waitlcd** potprograma i isključivo radi mogućnosti CCSv6 da prikaže registre i njihovo stanje korak po korak u kodu uspio sam otkriti zašto iz ponekad potprogram daje točan stupac odabrane brojke, ali ne i redak.

```
unsigned char
get_key(void)

{    k=1;

      for(i=0;i<4;i++)  {

   keyport = (0x01<<i);

    waitlcd(2);

  if(COL2){

           key=k+0;

           while(COL2);

           return key;

           }

  if(COL1){

           key = k+1;

           while(COL1);

           return key;

}
```

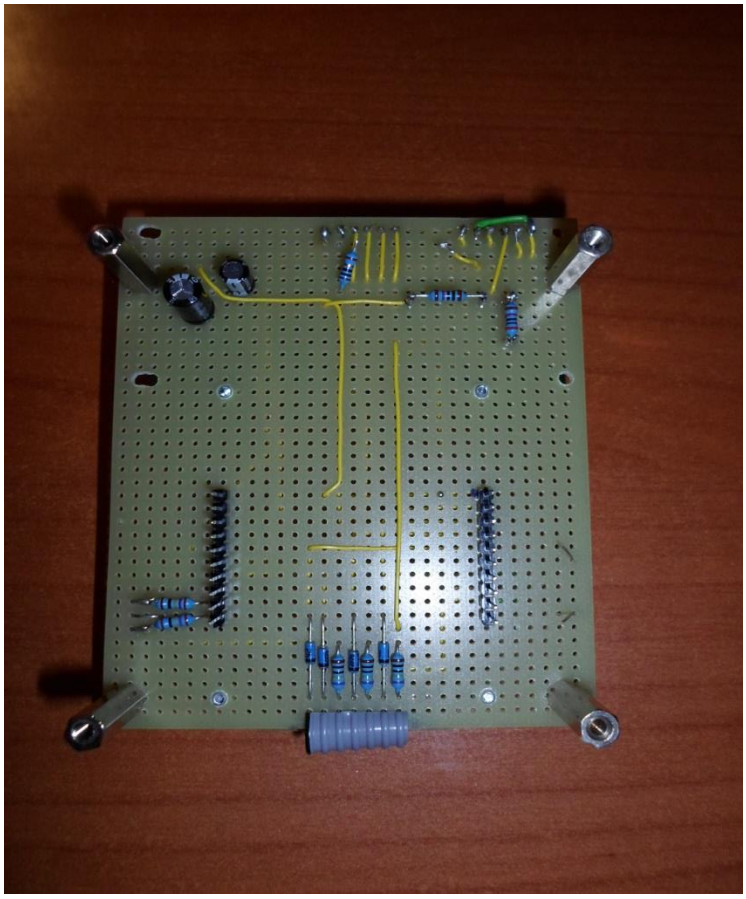
```
/* if(COL3){
            key = k+2;
            while(COL3);
            return key;
}*/
       k+=3;
       }
       return 0;
}
```

COL2 označuje prvi stupac i pin mikrokontrolera P1.4, COL1 drugi stupac i pin P1.3, dok otkomentirani dio (COL3) ostavlja mogućnost za 20 pinski mikrokontroler, predstavljajući pin P2.0. Keyport je globalno definiran P1OUT.

6. Rad sa sustavom

6.1 Vanjski izgled sustava

Kao fizikalna platforma cijelog sustava koristi se *veroboard* radi lakšeg povezivanja komponenata, te radi lakšeg prijenosa cijelokupnog dizajna.

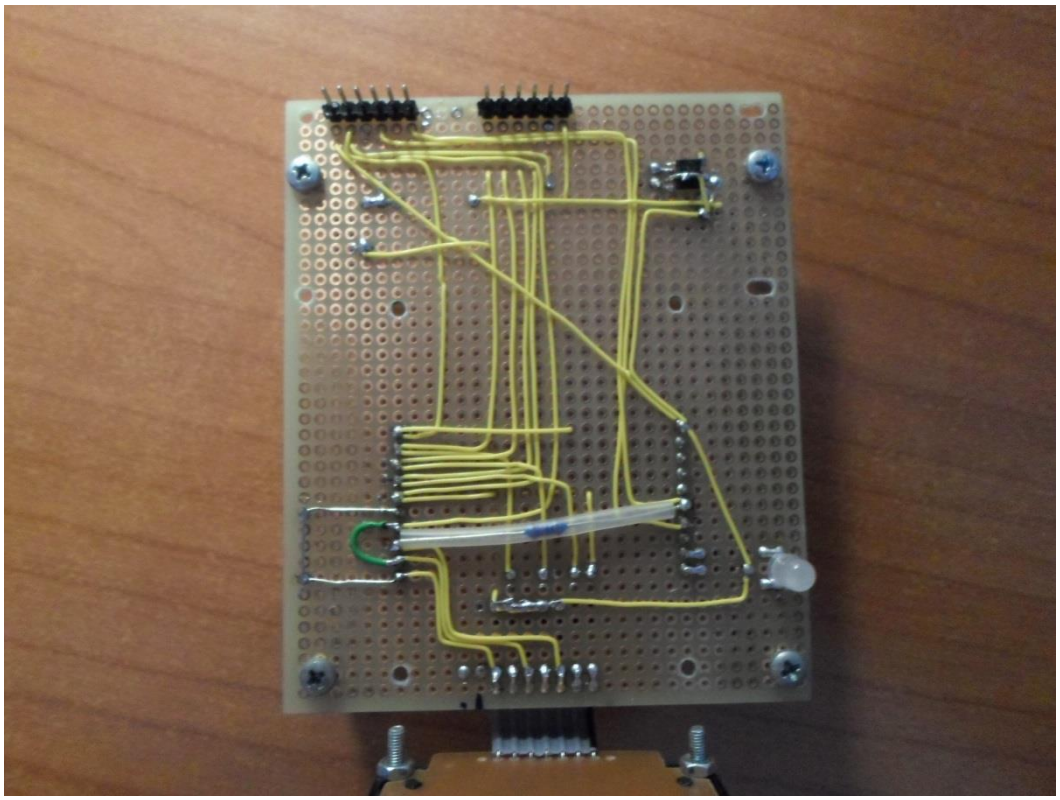


Slika 6.1 donja strana uređaja, bez mikrokontrolera MSP430

Korištena platforma mikrokontrolera ima izlaze izvedene kao utore (*female)*, pa se radi toga na *veroboard* montira 2x10 *male* priključnica kako bi mikrokontroler bolje prijanjao i kako bi se moglo izvršiti slanje signala. S donje strane vidljivi su otpornici i diode korištene za povezivanje tipkovnice.

Pored lijevih pinova pričvršćenih na podlogu nalaze se 2 otpornika i vod uzemljenja koji su potencijalno služili kao vodovi za dvobojnu LED diodu čija je svrha bila pokazivanje točnosti lozinke. S gornje strane nalaze se ulazi za LCD prikaznik,

otpornici korišteni u podešavanju kontrasta, te elektrolitski kondenzatori koji služe za generiranje negativnog napona.



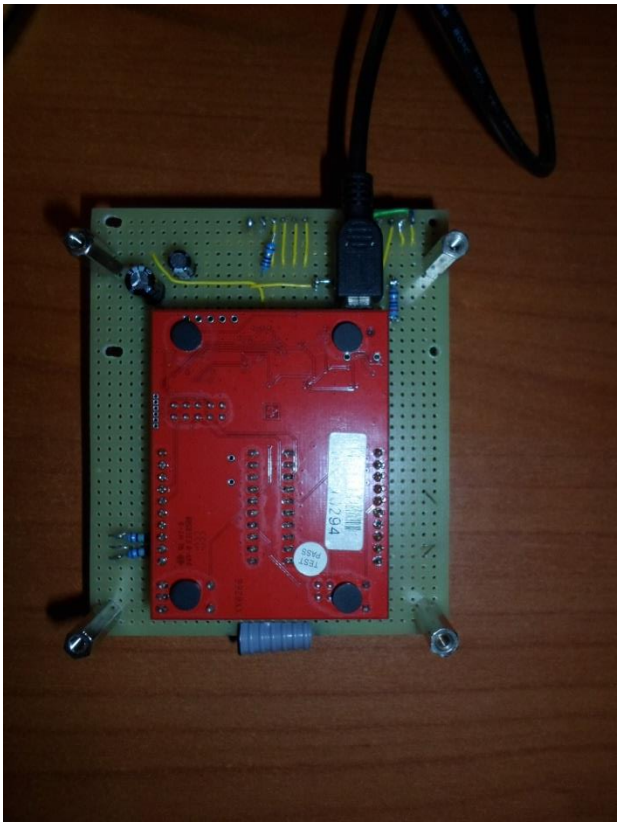
Slika 6.2 Gornja strana uređaja, bez priključenog LCD prikaznika

Na gornjem dijelu slike nalaze se ponovno pinovi koji služe za priključivanje LCD prikaznika, pričvršćeni radi istog razloga kao kod mikrokontrolera (LCD prikaznik ima utore). Također na gornjem dijelu slike s desne strane nalazi se čip ICL7660. Pinovi mikrokontrolera nalaze se u dvije kolone po 10 pinova, a priključci tipkovnice nalaze se pri dnu slike. Srebrna i zelena žica označavaju promjenu u dizajnu, ne koristeći pinove P2.1 i P2.2, nego pinove P1.4 i P1.5 za očitavanje ulaznog signala s tipkovnice.

Izolirani otpornik koji je spojen na pin P2.0 i na pin P1.7 bio je svojevrsni pokušaj da pin P1.7 (RS) osposobim kao ulazni i izlazni. Ulazni u slučaju kada se očitava tipkovnica, a izlazni u slučaju kada se RS koristi za LCD prikaznik. Pošto je pin P2.0 spojen na otpornik iznosa 1MΩ, direktno spajati pin na tako veliki otpor stvarao bi

probleme pri korištenju RS-a. Zato sam kao pokušaj rješenja napravio svojevrsni razmak potencijala otpornikom koji se nalazi u izolatoru, iznosa 10kΩ. Implementacija tako zamišljenog dizajna nije uspjela i takav način aktivacije trećeg stupca tipkovnice nije izvršen, niti igdje zapisan u kodu.

Slika 6.3 Gornja strana sa svim komponentama



6.4 Donja strana sa pričvršćenim mirkokontrolerom

6.2 Upute za korištenje

Prilikom pokretanja sustava na zaslonu se pojavi poruka „Press *“. Kako bi se započeo postupak unošenja lozinke potrebno je stisnust simbol '*' na tipkovnici.

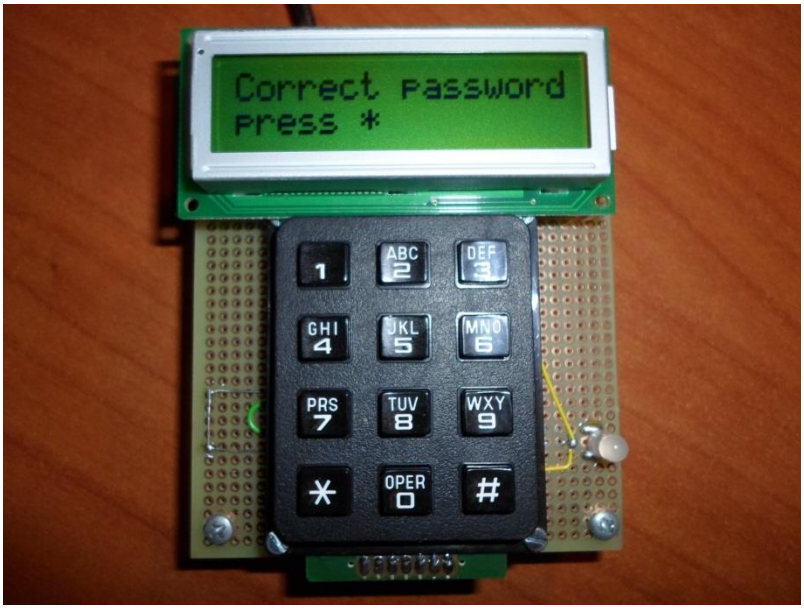
Slika 6.4 početno stanje sustava

Nakon što se stisne potreban simbol tipkovnica ispiše poruku „Write password“. Ovdje kreće unos potrebne lozinke. Trenutna lozinka postavljena u programu je sedmeroznamenkasta, 1282540.

Slika 6.5 Proces unošenja lozinke

Ovisno o točnosti unesene lozinke uređaj izbacuje poruke „Correct password“ (točna lozinka) ili „Wrong password“ (netočna lozinka).

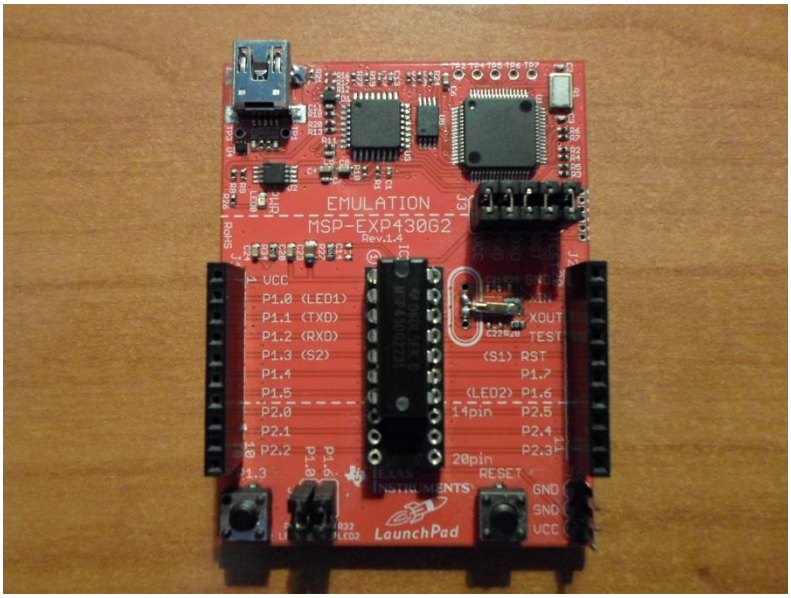
Kako bi se sustav vratio u početno stanje na slici 6.1 potrebno je kao što sugerira sam LCD prikaznik ponovno stisnuti simbol '*'.



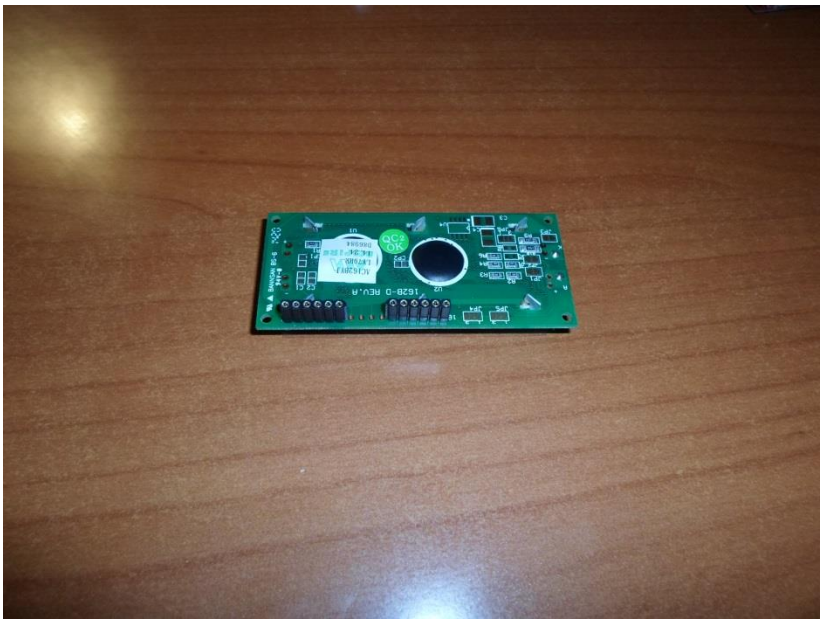
Slika 6.6 Prikaz reakcije LCD prikaznika na točnu lozinku

Slika 6.7 Prikaz reakcije LCD prikaznika na netočnu lozinku

6.3 Prikaz utora komponenata



Slika 6.7 Prikaz utora MSP-EXP430G2



Slika 6.8 Prikaz utora LCD prikaznika

Zaključak

U ovom radu obrađena je komunikacija mikrokontrolera s periferijama tipkovnice i LCD prikaznika. Komunikacija je postignuta kako bi se preko tipkovnice mogla unjeti lozinka za koju program potvrđuje ili ne. Za prijenos podataka na LCD prikaznik korištena je 4-bitna paralelna komunikacija.

Rad nije mogao biti realiziran u prvotno zamišljenom obliku s još jednom izlaznom periferijom, koja bi služila kao simulacija otvarnja sefa, i s potpunom funkcionalnošću tipkovnice, radi manjka vremena koje je potrebno kako bi se pribavio mikrokontroler koji bi zadovoljavao zahtjeve da ima 20 pinova, da je kompatibilan s *launchpadom* i da je moguće emulirati program preko CCSv6 na sam mikrokontroler.

Literatura

[1] Upotreba mikrokontrolera u ugrađenim računalnim sustavima, Mladen Vučić, Zagreb 2007.

[2] Specifications for LCD Module AC-162B, Ampire CO., LTD, 2000

[3] Implementing an Ultralow-Power Keypad Interface With the MSP430, Texas Instruments

[4] http://www.ti.com/lit/ds/symlink/msp430g2211.pdf, Texas Instruments

[5] MSP430x2xx Family, User's Guide, Texas Instruments

[6] MSP-EXP430G2 LaunchPad Evaluation Kit, User's Guide, Texas Instruments

[7] MAX 1044/ICL7660 Switched-Capacitor Voltage Converters

Upravljanje sefom pomoću mikrokontrolera MSP430

Sažetak

U ovom radu realizirana je izvedba mikrokontrolera MSP430G2211 (14-pinskog) u svrhu upravljajna sefom. Korištena je tipkovnica 3x4, ali su realizirana samo njena prva 2 reda. Radi malih razina napona kod izlaznih priključaka mikrokontrolera korištene su Schottkyeve diode. Korišten je 2x16 LCD prikaznik na koji se podaci šalju 4-bitnom paralelnom vezom. Za negativan napon pri kontrastu LCD prikaznika korišten je sklop ICL7660. Aktivacija upisivanja lozinke vrši se preko tipke '*'.

Ključne riječi: sef, MSP430, LCD 2x16, tipkovnica 3x4, ICL 7660, 14-pin, Schottky dioda

Safe Deposit Box Controller Based on MSP430 Microcontroller

Abstract

This paper describes the implementation of a safe deposit box using a microcontroller MS430G2211 (14-pin). The implementation consist of two pheripherals: a 3x4 keyboard and a 2x16 LCD module. Because of the low voltage outputs by the microcontroller the Schottky diode was used while handling the inputs of the keyboard (the rows). Because of the need to have negative voltage for the contrast of the LCD module, ICL 7660 chip was used. The process of writing the password on the LCD module begins by pressing '*' on the keyboard.

Keywords: safe deposit box, MSP430, LCD 2x16, 3x4 keyboard, ICL7660, 14-pin, Schottky diode

Dodatak 1

U ovom dodatku nalazi se kompletni kod, bez prije navedenih potprograma. U komentarima se nalaze rješenja koja bi ostvarila implementaciju izlaznog člana, u ovom slučaju LED diode, u svrhu pokazivanja točnosti lozinke.

lcd16.c

```
void lcdData(unsigned char l)
{
P1OUT |=RS;  //because sending
data
P1OUT  &=~EN;
P1OUT &= 0xF0;
P1OUT |=((l >> 4) & 0x0F);

P1OUT  |=EN;
waitlcd(2);
P1OUT  &=~EN;
P1OUT &=  0xF0;
P1OUT  |= (l & 0x0F);
P1OUT  |=EN;
waitlcd(2);
P1OUT  &=~EN;
}


void waitlcd(volatile unsigned int
x)
{
volatile unsigned int i;
for (x ;x>1;x--)
{
for (i=0;i<=110;i++);
}
}

void prints(char *s)
  {

    while (*s)
      {
       lcdData(*s);
       s++;
      }
  }
```

main.c

```
#include <msp430g2211.h>
#include "lcd16.h"
#include <stdio.h>
#define keyport P1OUT
#define COL1 (0x10 & P1IN)
#define COL2 (0x20 & P1IN)
//#define COL3 (0x01 &
P2IN)
unsigned char Key_Val[] =
{'
','1','2','3','*','0','#',
'7','8','9','4','5','6'};
unsigned char
get_key(void);
unsigned char k;
unsigned char i;
unsigned char key;
unsigned char value
```

Na ovom se mjestu nalazi potprogram **getKey**.

```
void main(void) {
       WDTCTL = WDTPW + WDTHOLD;
       P1DIR=0xCF;
       P1OUT=0x00;
       //P2DIR 0xFE
       //P2OUT=0x00
       unsigned char value;
       unsigned char pass [] =
{'1','2','8','2','5','4','0'};
       unsigned char flag;
    unsigned char g;
    lcdinit();

      while(1){
            flag=0;

            prints("Press *         ");
            do {
                  value=Key_Val[get_key()];

            } while(value!='*');

            gotoXy(0,0);
          prints("Write password  ");
   for(g=0;g<7;g++) {
        value=' ';

        do {
            value=Key_Val[get_key()];

         } while(value==' ');
```

```
gotoXy(g,1);
          lcdData(value);
          waitlcd(100);

          if(value!=pass[g]) flag=1;
     }
   gotoXy(0,0);

   if(flag)    prints("Wrong password  ");
   if(flag==0) prints("Correct password");

   gotoXy(0,1);
   prints("press *         ");



      // if(flag) P2OUT=0x08;  //p2.3
     // if(flag==0) P2OUT=0x10; // p2.4
      do {
        value=Key_Val[get_key()];

         } while(value!='*');

      // P2OUT=0x00;
      gotoXy(0,1);
      prints("                ");
      gotoXy(0,0);
    }

}
```