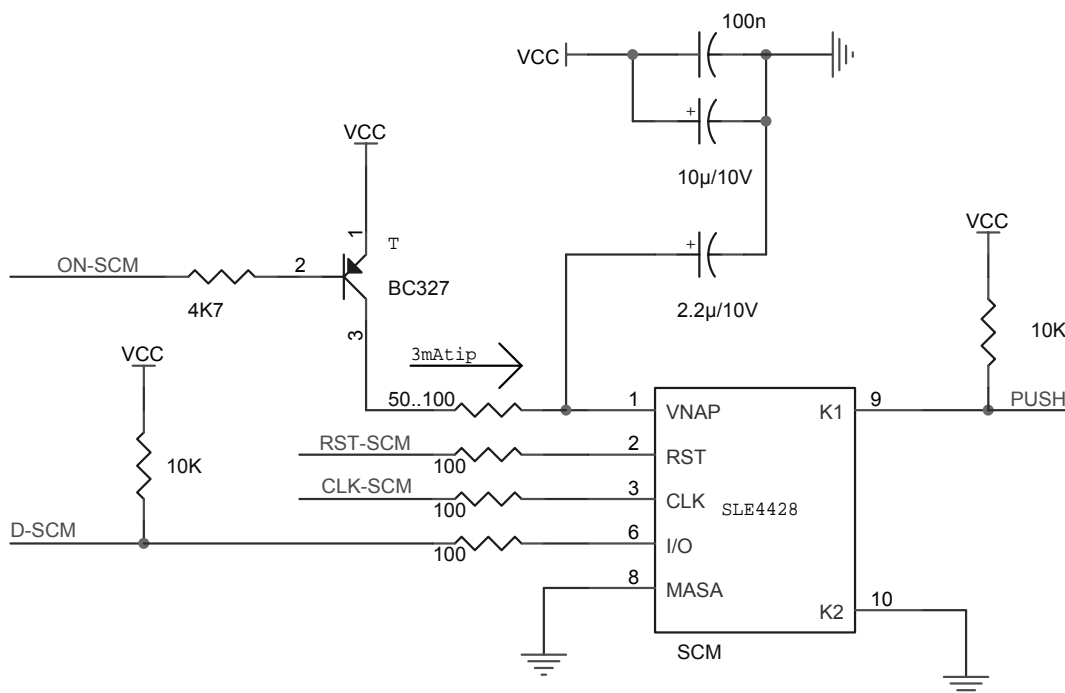


## Povezivanje memorijske kartice na mikrokontroler



SCM je kuciste u koje se ubacuje kartica. Ona ima 8 pina koja kontaktiraju memorijski cip na kartici. Ne koriste se svi pinovi, a njihov raspored moze da se utvrdi uporedjujuci raspored kontaktnih površina cipa na karici (PDF kartice) i raspored pinova na samom kucistu u koje se ubacuje kartica. Dodatna dva pina (9 i 10) su sa prekidača koji se aktivira pritiskom ubacene kartice. Postoje kucista koja imaju zatvoren kontakt kada nema kartice, ali postoje i ona kod kojih je to obrnuto.

Signal PUSH služi za detektovanje prisutnosti kartice u kucistu. Ovaj signal se uvodi u MCU (mikrokontroler) preko nekog pina koji se konfigurise kao ulazni.

Signal sa izlaznog pina MCU ON-SCM svojom log. 0 uključuje napajanje kartice. Napajanje kartice treba da postoji samo dok je detektovano prisustvo kartice u kucistu. Četri otpornika od 100 oma služe za zastitu u slučaju da neko pokuša da u kuciste ubaci neku metalnu pločicu. Obratiti pažnju na otpornik u napajanju, jer se na njemu stvara pad napona proizveden potrošnjom kartice (3mA za napajanje od 5V). Ovo može da smanji napajanje kartice ispod minimalne deklarisanе vrednosti napajanja (4.5V). Tranzistor će za struju napajanja od 3mA biti u dubokom zasícenju, i njegov napon emitor-kolektor će biti oko 0.1V. Ako treba onda se vrednost otpornika preko kojeg teče struja napajanja kartice može smanjiti. Postoji nekoliko cipova preko kojih se dravuju memorijske i mikroprocesorske kartice. Jedan od njih je 73S8024R od firme TDK, ali sa njima se nista posebno ne dobija, osim što uspesno resavaju problem zastite i napajanja. Ja sam koristio gornju semu za karticu SLE4428 i nisam imao

probleme. Za mikrokontrolerske kartice koristim pomenuti cip, jer te kartice trose vise struje, koja uz to i vise varira (potreban je i filterski kondenzator na VNAP).

Komunikacija je sinhrona bidirekionalna po jednoj zici. CLK-SCM je klok komunikacije, a D-SCM je bidirekionalna veza po kojoj se izmenjuju podaci izmedju kartice i MCU. CLK-SCM se generise na izlaznom pinu MCU-a, a D-SCM se vezuje na pin koji u toku komunikacije treba da se jedno vreme definise kao izlazni, a zatim kao ulazni. Komunikacija radi po principu master-slave, gde je MCU master. CLK-SCM frekvencija je tipicno 20KHz.

Slede delovi programa napisani za stari 68HC11. Da bi se bolje snasao, treba da znas da ima dva 8-o bitna akumulatora A i B, koja zajedno cine 16-to bitni akumulator D (B je visi bajt). Takodje ima i 16-to bitne indeksne registre X i Y, kao sto su i stek pointer i programski brojac.

```
{-----}
procedure SCMon; //uklucuje napajanje kartice log. 1 jer je ovaj signal
begin           //kod mene isao kroz invertor
    psha;

    ldaa [PortB];
    oraa OnKard; //OnKard = %00000010 drugi bit porta B koji je knfi-
    staa [PortB]; //gurisan kao izlazni

    pula; //cuva i obnavlja sadrzaj akumulatota A
end;    //na mesto end kompajler stavlja rts naredbu

{-----}
procedure SCMoff; //iskljucuje napajanje kartice
begin
    psha;

    ldaa [PortB];
    anda 255-OnKard; //255-OnKard = %11111101
    staa [PortB];

    pula;
end;

{-----}
procedure TpolA; //Ovo je vremenska zadraska koju sada ne mogu
begin           //da izracunam jer treba da nadjem tablicu sa
    pshb;         //brojem ciklusa instrukcija i da to pomnozim
    ldab 8;       //sa 500ns (2MHz je klok MCU-a. Ako ti treba
    repeat       //potrudicu se da ti nadjem.
        decb;
    until =0;    //posle decb, testira se fleg za nulu (z) i ako
    pulb;        //je on 1 zavrшава se petlja, inace se skace na
end;          //lokaciju prvu iza repeat

{-----}
procedure td; //I ovo je neka kratka zadraska
begin
    nop;
    nop;
end;

{-----}
```

```

procedure RSTon; //Postavlja signal RST-SCM na log 1
begin
    td; //Ovo je poziv procedure td, isto je sto i jsr td
    ldaa [PortA];
    oraa RSTSCM;
    staa [PortA];

    ldaa [DDRD]; oraa DSCM; staa [DDRD]; //konfigurise D-SCM pin kao
                                         //izlazni
    td;
end;

{-----}
procedure RSToff; //Postavlja signal RST-SCM na log 0
begin
    td;

    ldaa [DDRD]; anda 255-DSCM; staa [DDRD]; //Konfigurise D-SCM kao
                                         //ulazni
    ldaa [PortA];
    anda 255-RSTSCM;
    staa [PortA];
    td;
end;

//Zapazio si iz PDF-a za karticu da se njoj salju podaci kada je
//RST-SCM na 0, a od nje se primaju podaci kada je RST-SCM na 1.
//To je razlog sto se u predhodnim dvema procedurama odredjuje smer
//D-SCM pina

{-----}
procedure ClkSCMon; //Postavlja CLK-SCM na 1
begin
    td;
    ldaa [PortA];
    oraa CLKSCM;
    staa [PortA];
    td;
end;

{-----}
procedure ClkSCMoff; //Postavlja CLK-SCM na 0
begin
    td;
    ldaa [PortA];
    anda 255-CLKSCM;
    staa [PortA];
    td;
end;

{-----}
procedure DSCMon; //Postavlja D-SCM na 1
begin
    td;
    ldaa [PortD];
    oraa DSCM;
    staa [PortD];
    td;
end;

{-----}

```

```

procedure DSCMoff; //Postavlja D-SCM na 0
begin
    td;
    ldaa [PortD];
    anda 255-DSCM;
    staa [PortD];
    td;
end;

{-----}
procedure ResetKar; //Resetuje karticu
begin
    ldab 8;
    repeat
        pshb;

        ldaa [PortA];
        anda 255-RSTSCM; //Postavlja RST-SCM na 0
        staa [PortA];

        SCMon; //i ukljucuje napajanje kartice

        Tpola;Tpola; //Pravi potrebnu pauzu
        Tpola;Tpola;
        Tpola;Tpola;

        RstOn; //i podize RST-SCM na 1
        ClkSCMon;
        tPola;
        ClkSCMoff; //Generise impuls CLK-SCM trajanja tPola
        RstOff; //pa obara RST-SCM na 0
        tPola;

        ldab 31;
        repeat //Generise 31 CLK-SCM impulsa
            pshb;

            tPola;

            ClkSCMon; tPola; ClkSCMoff;

            pulb;
            decb;
        until =0;
        pulb;
        decb;
    until =0;
end;

{-----}
procedure SCMpak; {A-podatak;B-kod; X-adresa}
begin //Na osnovu koda, podatka i adrese pravi 3 bajta
    psha; //u 3-o bajtnoj varijabli SCMvar za slanje kartici

    clra;
    lsr; rola; lsr; rola;
    lsr; rola; lsr; rola;
    lsr; rola; lsr; rola;

    psha;

```

```

xgdx;          //Registri D i X zamenjuju medjusobno svoje vrednosti
tsx;          //U indeksni registar X se prebacuje vrednost SP+1
lsla;lsla;lsla;lsla;lsla;lsla;

oraa x[0];     //Kako X sadrzi SP+1 adresirana je vrednost koja
               //je sa psha stavljena na stek
stab [SCMvar+1];
staa [SCMvar+2];
ldaa x[1];
staa [SCMvar];

pulx;         //Skida sa steka bajt koji je predhodno stavljen
end;

{-----}
procedure ToSCM; //Salje 3-o bajtni podatak karici
begin
  RSTon;      //Dize RST-SCM na 1

  ldab 24;
  repeat     //Salje 24 bita preko D-SCM
    pshb;

    lsr [SCMvar]; ror [SCMvar+1]; ror [SCMvar+2];

    if sc then DSCMon //Postavlja bit D-SCM na 1
      else DSCMoff; //ili 0

    ClkSCMon; tPola; ClkSCMoff; tPola; //Generise jedan CLK-SCM

    pulb;
    decb;
  until =0; //testira kraj ponavljanja

  RSToff;    //Obara RST-SCM na 0
  tPola;
end;

{-----}
procedure FromSCM; //citanjne 9-o bitnog podataka
begin             //8 bita su podaci a 9 je bit zastite je
  ldd 0; std [SCMvar]; //zastita od upisa u tu lokaciju

  ldab 9;
  repeat
    pshb;

    ClkSCMon; tPola; ClkSCMoff;
    td;

    ldaa [PortD];
    anda DSCM; //Ako je DS-SCM = 1 onda kada se tome doda
    adda 255; //255 javlja ce bit prenosa, inace ne
    ror [SCMvar];
    ror [SCMvar+1];

    tPola;

    pulb;

```

```

    decb;
until =0;

    ldd [SCMvar];
    lsrđ;lsrd;lsrd;lsrd; //siftanje udesno za 7 bita dobija se
    lsrđ;lsrd;lsrd; //podatak u [SCMvar+1] a u [SCMvar] na
    std [SCMvar]; //mestu najnižeg bita bit zastite

end;

{-----} //Slanje komande za 9-to bitno
procedure InitRead9; {X-adresa} //citanje
begin
    ldaa $55; ldab Read9;
    SCMPak;
    ToSCM;
end;

{-----}
procedure CitaKartu; //Citanje podataka sa kartice
begin
    CopReset; //Resetovanje watch-dog tajmera

    ResetKar; //Resetovanje kartice

{ validacija }
    ldx SCMVld; InitRead9; //Inicira 9-to bitno citanje lokacije
    //kartice na adresi SCMVld = 32;
    FromSCM; //a to je 2-o bajtni validacioni
    ldaa [SCMvar+1]; staa [exVal]; //broj
    FromSCM;
    ldaa [SCMvar+1]; staa [exVal+1];

{ tip }
    ldx SCMTip; InitRead9; //Citanje bajta tipa kartice
    FromSCM;
    ldaa [SCMvar+1]; staa [exTip];

{ kredit }

    ldx SCMKredit; InitRead9; //Citanje kredita na kartici koji je
    FromSCM; //4-o bajtni

    ldaa [SCMvar+1]; staa [exKarta.aKredit];
    FromSCM;
    ldaa [SCMvar+1]; staa [exKarta.aKredit+1];

    FromSCM;
    ldaa [SCMvar+1]; staa [exKarta.aKredit+2];
    FromSCM;
    ldaa [SCMvar+1]; staa [exKarta.aKredit+3];

{ broj kartice }
    ldx SCMbr; InitRead9;
    FromSCM;

    ldaa [SCMvar+1]; staa [exKarta.abrKartice];
    FromSCM;
    ldaa [SCMvar+1]; staa [exKarta.abrKartice+1];

```

```

CopReset;
end;

{-----}
procedure SCMPrg; {B-broj bita} //Programiranje kartice. Zahtevani
begin //broj ciklusa se generise da bi
  repeat //proces u kartici bio završen
    pshb;

    ClkSCMon; tPola; ClkSCMoff;

    tPola;

    pulb;
    decb;
  until =0;
end;
{-----}
procedure OtkljucavaSCM; //Otkljucavanje kartice slanjem PIN
label //koda kako bi moglo da se upisuje u
  Dalje; //njene lokacijama, ako nije postavljen
begin //9-ti bit zastite

  ldx 1021; InitRead9; //Citanje error bajta. Ako je on 0
  FromSCM; //kartica je trajno zakljucana. Svaki
  //poslat netacan PIN kod obara po jednu
  ldx 7; //log 1 na log 0 bita u error bajtu. Dakle
  ldaa %10000000; //najvise moze biti 8 neuspelih pokusaja.
  repeat
    tab; //Pronalazi prvi, sa leva u desno, bit
    andb [SCMvar+1]; //u error bajtukoji je log 1
    if <>0 then jmp Dalje;

    lsra;
    decx;
  until =0;

Dalje:;
  coma; anda [SCMvar+1]; //i obara ga na log 0

{ omogucuje pristup za upis }
  ldab %00010011; ldx 1021; SCMpak; //Pakuje adresu, komandu i novu
  ToSCM; //vrednost error bajta i salje
  //je kartici
  ldab 103; SCMPrg; //omogucuje upis error bajta

  ldaa PINcod div 256; //Pakuje nizi bajt PIN koda, kod naredbe
  ldab %00101100; //i adresu za taj bajt
  ldx 1022;
  SCMpak;
  ToSCM; //Salje na karticu
  ldab 2; SCMPrg; //Obezbedjuje klokove za verifikaciju

  ldaa PINcod mod 256;
  ldab %00101100; //Isto za visi bajt
  ldx 1023;
  SCMpak;
  ToSCM;
  ldab 2; SCMPrg;

```

```
ldaa $ff;           //Postavlje sve bitove error bajta na 1
ldab %00110011;    //sto obezbedjuje novih 8 pokusaja otklucavanja
ldx 1021; SCMpak;
ToSCM;
ldab 203; SCMPrg;
```

**end;**

Ako nesto nije jasno odmah pitaj sa citiranjem onoga sto ti pravi problem.