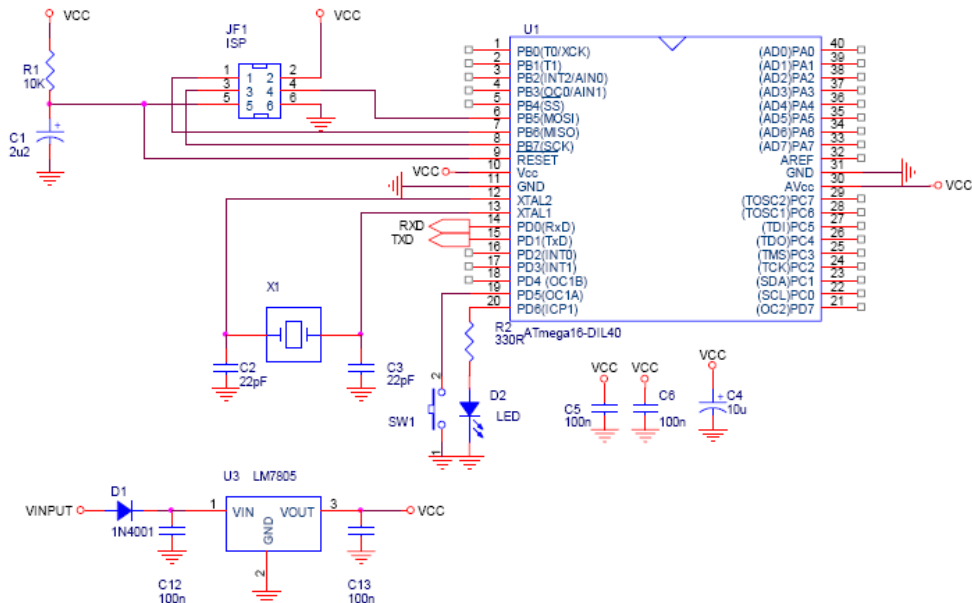


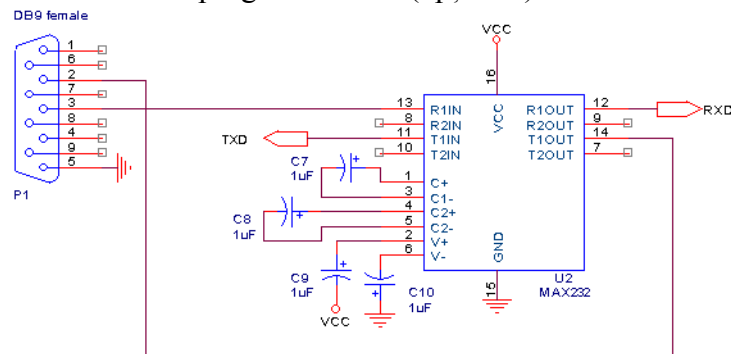
Proiect 2

Schema electrică de bază

Schema de bază minimală, care cuprinde controlerul și componentele esențiale pentru funcționarea și programarea controlerului este data în figura. Aceste componente se vor regasi pe toate machetele de proiect, indiferent de teme. În puls se vor utiliza și componente specifice temei proiectului (conectori suplimentari, switch-uri, LED-uri, potențioetre sau senzori, tranzistoare rele, etc.).



(a) Nucleul circuitului: sursa stabilizată, microcontroler, circuitul de Reset, cuarțul, conectorul de programare ISP (opțional)



(b) conectarea circuitului MAX232/ 202, ST232*

Fig. 1. Schema electrică de bază pentru proiect

Componentele R1, C1 formează circuitul de “power-on reset” (resetare la pornire). În momentul aplicării Vcc, C1 este descărcat, aplicând comanda de reset hardware controlerului. Apoi, condensatorul se încarcă după o lege exponențială prin R1 până la valoarea Vcc. Când tensiunea de pe C1 ajunge la aprox 3V (“1” logic) comanda *Reset* nu mai este aplicată controlerului, astfel încât acesta va începe rularea programului din Flash (sfârșitul încărcării se consideră după un timp aproximativ egal cu $5R1C1$). Acest reset asigură pornirea în bune condiții a controlerului; după

stabilizarea tensiunii de alimentare. (Se evită ca tensiunile tranzitorii care apar în momentul alimentării să ducă la ajungerea procesorului într-o stare incertă).

Condensatoarele C5 și C6 se vor lipi cât mai aproape de controler, între pinii 10-11, respectiv 30-31. Ele sînt condensatoare de decuplare prevenind apariția zgomotului de comutare în traseele de alimentare și sînt specifice alimentării oricărui circuit digital.

Condensatorul electrolitic C4 nu este obligatoriu, dar este recomandat pentru reducerea riscului ca microcontrolerul să se reseteze din cauza unor glitch-uri ale tensiunii de alimentare.

Cristalul de cuarț, împreună cu condensatoarele C2, C3 și cu amplificatorul intern de la bornele XTAL1,2 formează un oscilator cu cuarț. Aceste componente se vor lipi, de asemenea, cât mai aproape de pinii respectivi ai controlerului.

Pentru alimentare se folosește stabilizatorul U3(7805), 2 condensatoare de 100nF necesare pentru stabilitatea funcționării acestuia, și dioda D1 care protejează la alimentarea inversă.

Dioda LED este poziționată astfel încît se aprinde cînd PD6 este pe “1” logic; rezistența R2 asigură limitarea curentului prin LED la cca. $(5-1.6)/330=10\text{mA}$.

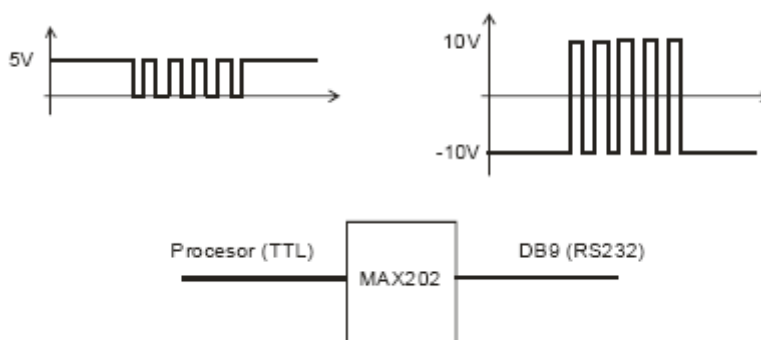
La conecarea unui LED la ieșirea unui circuit digital, fără tranzistor de comandă, trebuie să se verifice că ieșirea circuitului suportă curentul necesar LED-ului. În cazul nostru porturile controlerului Atmel suportă curenti egali pe “0” și pe “1” logic, astfel că LED-ul se poate conecta atât către GND cât și către Vcc.

Butonul SW1 leagă PD5 la masa (0 logic) în momentul apăsării. Întrucît, atunci cînd nu este apăsat, starea pinului PD5 nu este definită, va trebui activată prin software rezistența de *pull-up* internă controlerului pentru pinul respectiv. (Pentru ce alți pini se mai activează rezistențele de *pull-up* ? Vezi datasheet-ul).

Comunicația prin port serial

Controlerul va utiliza comunicația serială pentru a interacționa cu PC-ul (pentru a primi eventuale comenzi și a afișa valori în *hyperterminal*), dar și pentru a încărca programul aplicație (utilizator) în memoria Flash. Controlerul primit este preprogramat astfel încît conține (în Flash) un soft special – bootloader-ul - care permite încărcarea programului aplicație prin interfața USART (serială), fără a mai fi nevoie de un programator hardware pentru controler. Valorile tensiunilor generate sau primite de controler trebuie să fie 0 și 5V pentru semnale digitale și între 0-5V pentru intrările analogice ale controlerului (analogic input, ADC).

Comunicația serială, conform standardul RS232 prevede o tensiune între +6V..+15V față de GND pentru “0” logic și de -6V..-15V față de GND pentru “1” logic. Se observă că logica este “inversată” și sînt necesare tensiuni în afara domeniului furnizat direct de către sursa de alimentare (5V). În figura de mai jos este reprezentată forma de undă TTL (0-5V) și respectiv RS232 pentru octetul “01010101”.



Pentru un număr exprimat în binar în notația obișnuită $b_7b_6b_5b_4b_3b_2b_1b_0$, unde b_7 este MSB și b_0 este LSB, comunicația serială este de tipul LSB-first, astfel că întâi se transmite bitul b_0 și ultimul b_7 . În plus comunicația serială conform standardului RS232 este de tip *asincron* (câte un singur *caracter* format din mai mulți biți – simboluri). Un caracter conține un simbol de *start* “0” logic, cele 8

simboluri (0/1) corespunzătoare celor 8 biți și apoi simbolul de *stop* „1” logic. Așadar, după simbolul de start (0) urmează LSB (b_0), iar după ultimul simbol al caracterului, (b_7), urmează simbolul de stop (1).

Practic, pe figură se observă că bitul “1” de stop durează mult mai mult decât ceilalți biți, dar aceasta se întâmplă pentru că, în absența datelor, linia este ținută în “1” logic, deci după ce se transmite acest bit “1” starea liniei nu se schimbă, pînă la transmiterea unui nou octet, care va începe cu un “0”.

Simbolurile de stop și de start garantează faptul că la începutul fiecărui octet are loc o tranziție “1 → 0”, chiar și în cazul în care octeții transmiși reprezintă lungi șiruri de “1” sau “0”. Dezavantajul este că la fiecare 8 biți de date trebuie adăugați cei 2 biți, adică eficiența transmisiei este de maxim 80%.

Pinii RxD, TxD ai controlerului se conectează la convertorul de nivel realizat cu circuitul MAX232, ST232 sau echivalent. Rolul acestuia este să convertească nivelurile TTL (0..5V) în niveluri RS232, și viceversa; practic, tensiunile +10V și -10V se obțin prin dublarea tensiunii de alimentare a circuitului, folosind un convertor intern cu capacități comutate, care folosește condensatoarele externe C7-C10.

Observație: condensatoarele de 1μF vor fi înlocuite cu 100nF în cazul în care se folosește ST232 în locul circuitului MAX232 sau MAX202 (de generație mai veche). Aceste condensatoare sînt polarizate și se vor monta cu borna (+) respectiv înspre pinii 1,4,2, iar C10 se va monta cu (+) la masa și (-) la pinul 6, ca în figura 1.

Verificarea funcționării acestui circuit în regim static (fără a transmite date) se face măsurînd cu voltmetrul tensiunile de pe pinii 2 și 6, față de masă. Trebuie să se obțină o tensiune de cca. 7..10V cu polaritatea corespunzătoare: (+) pe pinul 2 și (-) pe 6. Practic, dublarea tensiunii la valoarea de 10V se obține numai în cazul ideal și anume în gol; în sarcina această tensiune este mai mică, dar se încadrează în standard. Pe pinii de date, în pauzele dintre transmisii, trebuie să se măsoare “1”, adică +5V pe TTL și circa -10V pe RS232 (logica inversată).

Conectorul DB9 de tip mamă este cablat după standardul DCE (Data Communications Equipment), iar conectorul tată (al PC-ului) este cablat după standardul DTE (Data Terminal Equipment) la care pinii 2 și 3 sînt inversați:

nr pin	DB9 – DCE (mamă)	DB9 – DTE (tată)
2	Tx (transmisie; sens de ieșire)	Rx (recepție; sens de intrare)
3	Rx (recepție; sens de intrare)	Tx (transmisie; sens de ieșire)
5	GND	GND

Aceasta înseamnă că se va folosi un cablu serial direct (1-la-1) pentru conectarea la PC, care este echipat cu o mufa de tip tată, „

Notă : dacă s-ar dori conectarea a 2 echipamente cu același standard (DCE cu DCE sau DTE cu DTE; de exemplu, 2 PC-uri legate între ele pe serială), ar trebui un cablu de tip “crossover” (pinii 2 și 3 inversați la unul din capete). De obicei, toate cablurile disponibile comercial cu mufe mamă-tată sînt de tip direct, iar cele mamă-mamă sînt de tip crossover.

Depanarea portului serial

Dacă nu funcționează comunicația machetă-PC, se poate folosi un osciloscop cu care să se vizualizeze formele de undă în mai multe puncte.

Pentru 9600bps un caracter are 10 biți (din care simbolul *stop* de lungime variabilă în funcție de frecvența de transmisie a caracterelor) și durează minim 1ms.

$$10 \text{ biți} \cdot 1/9600 \text{ sec/bit} \approx 1 \text{ ms}$$

Astfel, dacă se va lucra cu $C_x = 0.1 \text{ms/div}$ (pentru osciloscop) se va vizualiza un caracter pe tot ecranul (10div).

$$T_x = N_x C_x \rightarrow C_x = T_x / N_x = 1 \text{ms} / 10 \text{div} = 0.1 \text{ms/div}$$

Atenție la reglajul de sincronizare pentru a declanșa imaginea la începutul unui caracter! bitul de start începe cu un front negativ de la 5V la 0V pentru nivele TTL (deci slope= falling), și unul pozitiv de la -10V la +10V pentru RS232 (comunicația între machetă și PC). Sincronizarea trebuie deci reglată diferit în funcție de punctul unde se observă forma de undă.

Se va ține apăsată o tastă pe tastatura PC-ului (după ce se pornește programul de terminal). În softul de test (aflat în flash-ul controlerului primit, soft cu funcția de aplicație de test, suplimentar față de bootloader), macheta trebuie să trimită înapoi caracterul următor.

Se urmărește pe osciloscop ordinea evenimentelor (crocodilul de masă al osciloscopului se poate conecta la aripioara metalică a lui LM7805):

- de la PC, folosind terminalul, se emite un caracter (RS232) care
- ajunge pe DB9.2 (nivel RS232), apoi la intrarea în MAX202 (pin 13),
- este convertit de MAX202 în nivel TTL și scos la ieșire (pin 12),
- ajunge la procesor (pin 14),
- acesta trimite caracterul cu codul imediat următor (procesor pin 15), ajunge la intrarea MAX202 (pin 11, nivel TTL),
- este convertit în nivel RS232 (MAX202 pin 14) și
- ajunge pe DB9.3 și de acolo pe cablu direct către PC. Lanțul este întrerupt în locul unde nu se mai observă forma de undă pe osciloscop.

Programarea controlerului folosind *bootloader*-ul

Programarea unui controler nou se poate realiza fie prin scoaterea lui din circuit și scrierea programului cu un circuit de programare dedicat, fie utilizând conectorul ISP (în system programming) la care se conectează un programator hardware (fără a mai scoate controlerul din circuitul aplicației).

O altă variantă, mai ieftină și mai comodă pentru teste este aceea de a se utiliza un soft special numit *bootloader* care se încarcă inițial în *flash*-ul controlerului. Pentru încărcare se utilizează un programator, dar operația se face o singură dată. După aceea, *bootloader*-ul comunică prin interfața serială sau USB (după conversie USB - serial) cu un program de PC numit *PC-loader* pentru a transfera softul de aplicație în memoria flash (de program). În proiectul nostru utilizăm un *bootloader* care comunică prin interfața serială. De aceea, pe placuta noastră de test s-a construit o interfață serială RS232 (cu MAX 232 sau ST232) care permite realizarea electrică a legăturii dintre interfața USART a controlerului și portul serial al PC-ului.

Bootloader-ul se plasează într-o zonă specială a memoriei procesorului, tipic la sfîrșitul flash-ului, și nu este suprascris de aplicație, pe care o încarcă de la începutul flash-ului. Cum, de asemenea, dimensiunea sa este mică, cea mai mare parte a memoriei flash rămâne disponibilă pentru aplicație. Practic, *bootloader*-ul este ca un sistem de operare minimal, care are rolul de încărcare și schimbare a aplicației.

Bootloader-ul este transparent pentru aplicația din microcontroler; acesta trebuie să ruleze aplicația în mod normal, fără interferențe, și să poată folosi portul serial. Există mai multe metode prin care poate fi invocat *bootloader*-ul; metoda cea mai "eleganță" este ca aplicația să accepte o anumită comandă pe serială și în acest caz să predea controlul *bootloader*-ului.

În cazul *bootloader*-ului furnizat, s-a ales o metodă mai simplă, care asigură o transparență totală pentru aplicația încărcată pe controller (aplicația nu trebuie să știe de existența *bootloader*-ului). El se activează la pornirea microcontrolerului și verifică dacă $PIND.5=0$; în acest caz, el rămâne activ și așteaptă să interacționeze cu programul *PC-loader*. În caz contrar, preda imediat

controlul aplicatiei. Deci din punct de vedere al utilizatorului e ca si cind n-ar exista, singura contrângere fiind ca pinul 5 al portului D să fie configurat ca intrare PIND.5 (cu rezistanța *pull-up* activată).

Asadar, pentru a incarca o aplicatie (fisier .HEX) se procedeaza astfel:

- se porneste programul *PC-loader*, se selecteaza cu Browse fisierul .HEX dorit; se conecteaza prin seriala placa de test la PC.
- se tine apasat butonul de pe placa, conectat la pinul D.5, se aplica alimentarea placii, dupa care se elibereaza butonul; se observa ca aplicatia din procesor nu porneste (LED-ul nu clipeste, etc); in acest moment controlul a fost preluat de catre *bootloader*, care asteapta.
- se apasa in *PC-loader* butonul “Start Upload”; in acest moment bara de progres indica incarcarea aplicatiei.
- dupa ce *bootloader*-ul primeste aplicatie, el preda controlul acesteia si isi inceteaza functionarea. O noua invocare presupune repetarea pasilor de mai sus.

Observatie: daca este deschis terminalul din CodeVision, trebuie apasat butonul “Disconnect” al acestuia inainte de a folosi *PC-loader*-ul, caci altfel apare conflict la portul serial. *PC-loader*-ul nu tine linia “conectata” decit cind transmite, deci poate fi lasat pornit tot timpul, fara a produce conflicte.

Dezavantajul acestei metode este ca programul *bootloader* depinde exact de configuratia de pe placa; in cazul nostru, procesorul nu poate fi decit AT Mega 16, trebuie sa existe un buton la pinul D.5, si cuartul trebuie sa aiba exact valoarea specificata. Orice modificare a acestor parametri presupune recompilarea *bootloader*-ului si re-incarcarea acestuia folosind un programator clasic.

Programarea ISP a controlerului cu programatoare (hardware)

Programarea ISP (“in-system programming”) inseamna posibilitatea de a programa procesorul fara a-l demonta din circuit; exista si varianta programarii separate, prin scoaterea din soclu si montarea in soclul unui programator dedicat.

Programarea ISP presupune existenta a 2 componente:

- 1) circuitul de programare (programatorul), care se conecteaza la conectorul ISP a placutei si la portul serial/paralel/USB al PC-ului.
- 2) soft-ul de programare, care preia fisierul utilizator si il transmite programatorului. Evident, trebuie sa fie compatibil cu acesta din urma.

Există mai multe programatoare și soft-uri pentru programare:

a) programarea direct din CodeVision; in acest caz, trebuie folosit un programator suportat de CV, configurat din meniul Settings->Programmer; dintre acestea, posibilitatile cele mai intilnite sint:

- STK500/AVRISP; acesta este un programator “industrial” care contine el insusi un microcontroller care implementeaza toata logica de programare, dar se poate și construi individual (exista scheme pe internet).

- STK200/300; cu acesta este compatibil un programator simplu care se conecteaza pe portul paralel; programatorul nu contine “inteligenta” deci toate semnalele, cu timingurile lor precise, sint generate de catre softul de programare.

b) programarea folosind un alt soft fata de CodeVision. Un exemplu este Pony Prog (www.lancos.com). Suporta atit programatorul paralel, cit si unul serial. Aici, în figura 2, este prezentată pe scurt varianta programatorului serial cu alimentare externă sau alimentare de pe interfața serială a PC-ului.

Conectorul CON10 al programatorului are următoarea mapare între pini și semnalele controlerului de programat:

Pini conector	1	2	4	5	6	10
Semnale controler de programat	Vcc	Reset	SCK	MOSI	MISO	GND

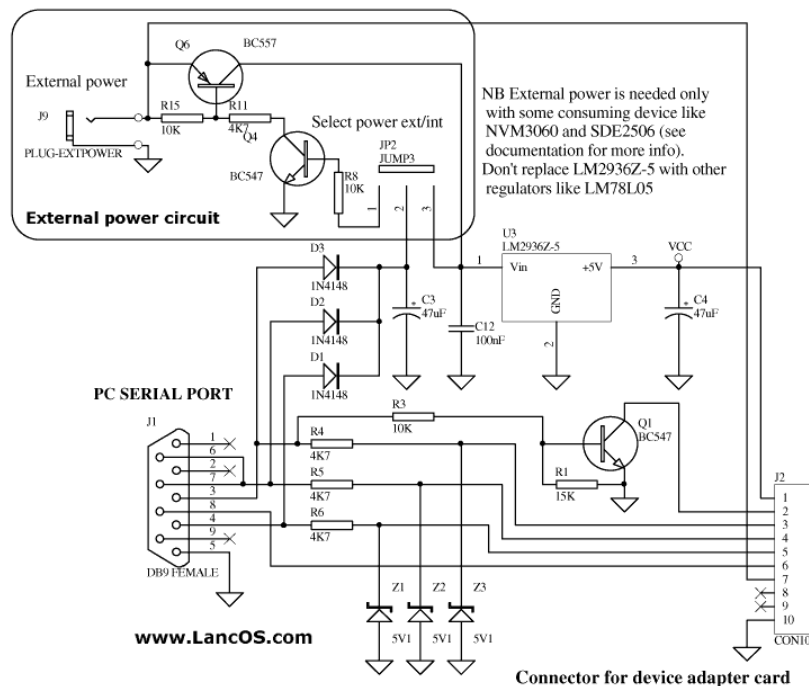


Fig. 2. Programator serial

Explicatii:

Aceasta schemă pe lângă programarea efectivă a controlerului țintă (prin transmisia semnalelor corecte pentru programare: MISO, MOSI, SCK, Reset, GND) permite alimentarea programatorului dintr-o sursă externă sau din portul serial al PC-ului.

Astfel, cele 3 diode sint conectate pe pini nefolositi ai portului serial, D1-D3, servesc la preluarea “parazita” a alimentarii direct din portul serial, care nu e conceput sa alimenteze periferice, asa cum e cazul la USB. Acesti pini, care au funcția de *flow control*, sunt programati de catre soft-ul *ponyprog* să aibă o tensiune de +12V pe ei. Tensiune pozitiva va încărca condensatorul C3, si apoi va fi stabilizată la valoarea de 5V de catre stabilizatorul U3 (similar cu 7805, dar cu o cadere de tensiune mai mica) și condensatoarele C12, C4.

Functionare:

Tranzistorul Q1, primind comandă de deschidere (tensiuni pozitive pe baza), ajunge în saturație generând comanda RESET pentru controler (0 logic) prin asigurarea unei tensiuni de cca 0.2 V apropiată de 0V.

Obs: conform standardului RS232, tensiunile generate de portul serial al PC-ului sunt ori tensiuni pozitive (8..12V), fie negative (-8..-12), niciodata 0V. Rezistenta R1 ține blocat Q1 atunci cind conectorul DB9 nu este conectat.

Rezistentele R5, R6 si diodele Zener corespunzatoare au rolul de a transforma tensiunile de +/-12V in cca. 5V/0V astfel: tensiunile pozitive mai mari de 5V sint limitate la 5V de dioda Zener, diferenta de tensiune este preluată de rezistenta (a carei valoare deci nu e critica), iar tensiunile negative sint limitate la minim -0.6V prin polarizarea directă a diodelor Zener (conducție directă), diferenta de tensiune fiind preluată tot de rezistențe. Astfel se obtine un “0” dar mai puțin “ferm” decit in varianta cu tranzistor, dar care nu afectează programarea corectă a microcontrolerului.

Astfel, dacă nu se dorește alimentarea controlerului prin programator se pot elimina componentele cu care se obținea tensiunea Vcc, rezultând o schemă simplificată (figura 3)

În această schemă, valorile rezistentelor nu sint critice. Q1 poate fi orice tranzistor NPN de uz general de putere mică: BC107-109, BC170-172, 2N2222, etc. Diodele Zener pot fi in gama 4V..5V1, de mica

putere. În acest caz s-a utilizat un conector ISP 2x3 cu următoarea mapare a pinilor: 1 - MISO, 3 - SCK, 4 - MOSI, 5 - RESET, 6 - GND.

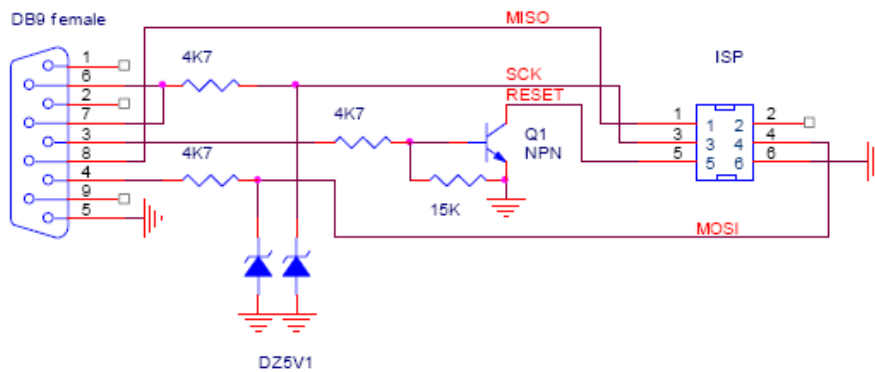


Fig.3. Programator serie simplificat

Există două tipuri de programatoare ISP cu conectare prin interfața USB cu PC-ul: care utilizează un convertor USB-serie, cu circuitul FTDI FT232, completat cu schema de programator serie, sau cu un alt controler (Atmega8, Atmega48). În figura 4 este prezentată schema programatorului prin USB care utilizează un controler Atmega8 sau 48, schema preluată de pe www.fischl.de/usbasp. Tot acolo se găsește și codul care trebuie scris în flash-ul controlerului. Schema funcționează cu AVRdude (<http://mirror.lhnidos.org/GNU/savannah/avrdude/>), Bascom-AVR (<http://www.mcselec.com/>) sau eXtreme Burner – AVR (<http://extremeelectronics.co.in/avr-tutorials/>).

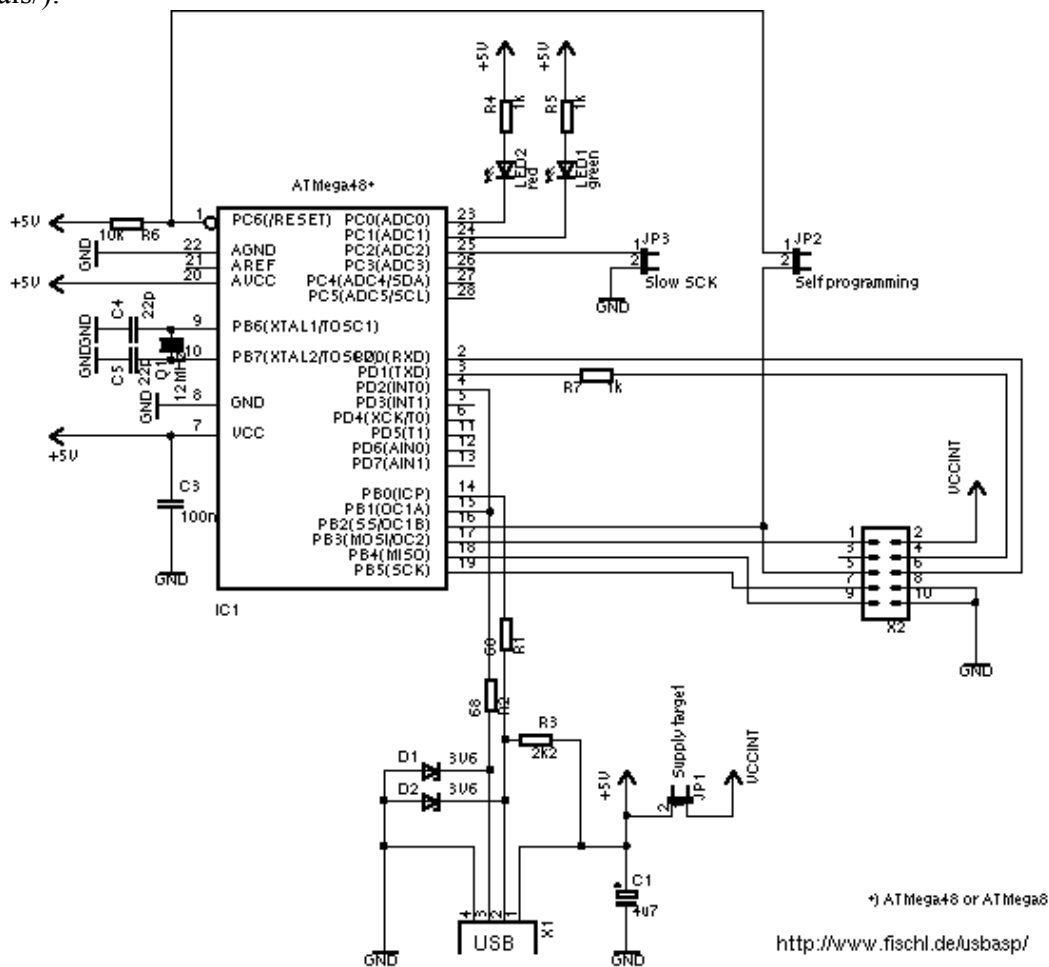


Fig.4. Programator USB cu Atmega8 sau Atmega48