

## Vmesnik za programiranje ATMEL AVR mikrokrmilnikov

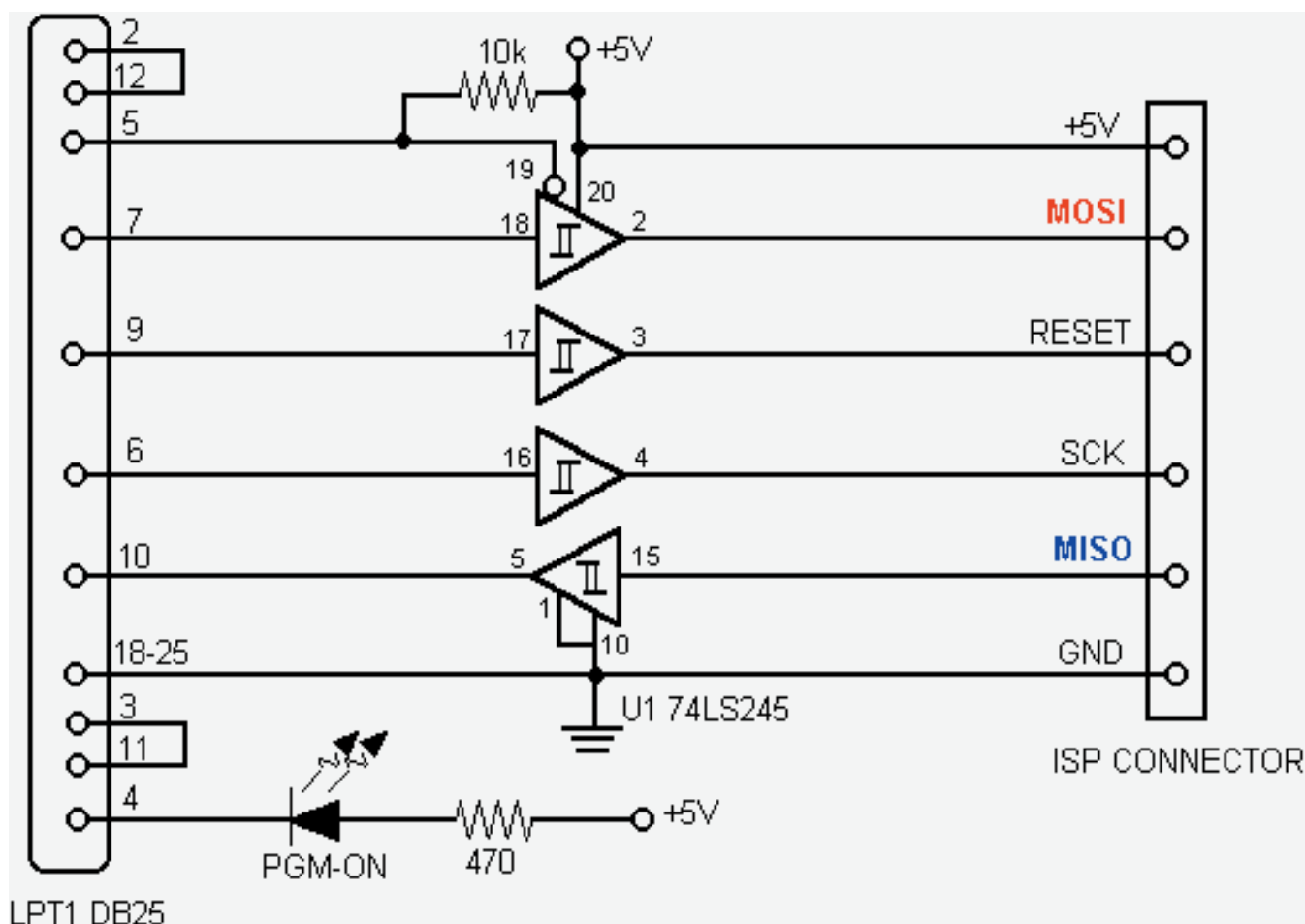
### Uvod

Pred dobrim desetletjem je bila večina elektronskih naprav z mikrokrmilnikom še velikosti Evropa formata kartice. Izjema so bila vezja z mikrokrmilnikom z EPROM-om v samem čipu, ki pa so bila praviloma pregrešno draga in OTP verzije, ki pa so za samogradnjo redko zanimive. Stanje se je bistveno spremenilo z začetkom masovne uporabe flash pomnilnikov. Proizvajalci so namreč začeli flash pomnilnik vgrajevati v čip skupaj z mikrokrmilnikom, dodali so še nekaj RAM-a in nastali so t.i. »single-chip« mikrokrmilniki. V začetku je bilo potrebno zanje nabaviti poseben programator ali pa vsaj programirni adapter. Končno so proizvajalci iznašli način, kako pretočiti kodo v silicij preko majhnega števila povezav, kar omogoča programiranje vezja, ki je že prispajkano na tiskanino, najlepše pri vsem pa je, da lahko naknadno zamenjamo kodo

z novejšo, ne da bi bilo potrebno vzeti vezje iz podnožja ali ga celo odspajkati. Vezjem, ki to omogočajo, ponavadi pravijo da so ISP (In System Programmable). Danes najdemo na trgu vezja z ISP podporo med mikrokrmilniki, mikroprocesorji, DSP-ji, logičnimi programabilnimi vezji itn.

### Opis vezja

V tem članku želim predstaviti vezje za programiranje Atmel AVR mikrokrmilnikov, ki so v zadnjem času postali zelo priljubljeni med samograditelji. ISP vmesnik je namenjen priključitvi na LPT (tiskalniški) vmesnik PC računalnika. Takoj na začetku naj poudarim, da vezje ni zraslo na mojem zelniku, ampak sem samo dodelal vezje, ki sem ga našel na internetu. Vezje je združljivo z izvornim vezjem za STK 200 razvojno AVR ploščo. Zategadelj ga podpira več programov za programiranje (npr. PonyProg) kakor tudi



LPT1 DB25

Slika 1. Shema originalnega vmesnika

nekatera okolja prevajalnikov (npr. Bascom AVR, Code Vision AVR, ...). Sam najraje uporabljam Pony Prog ([www.lancos.com](http://www.lancos.com)), saj poleg AVR podpira še kopico drugih vezij, nekatera iz družine 8051 združljivih celo z istim vmesnikom.

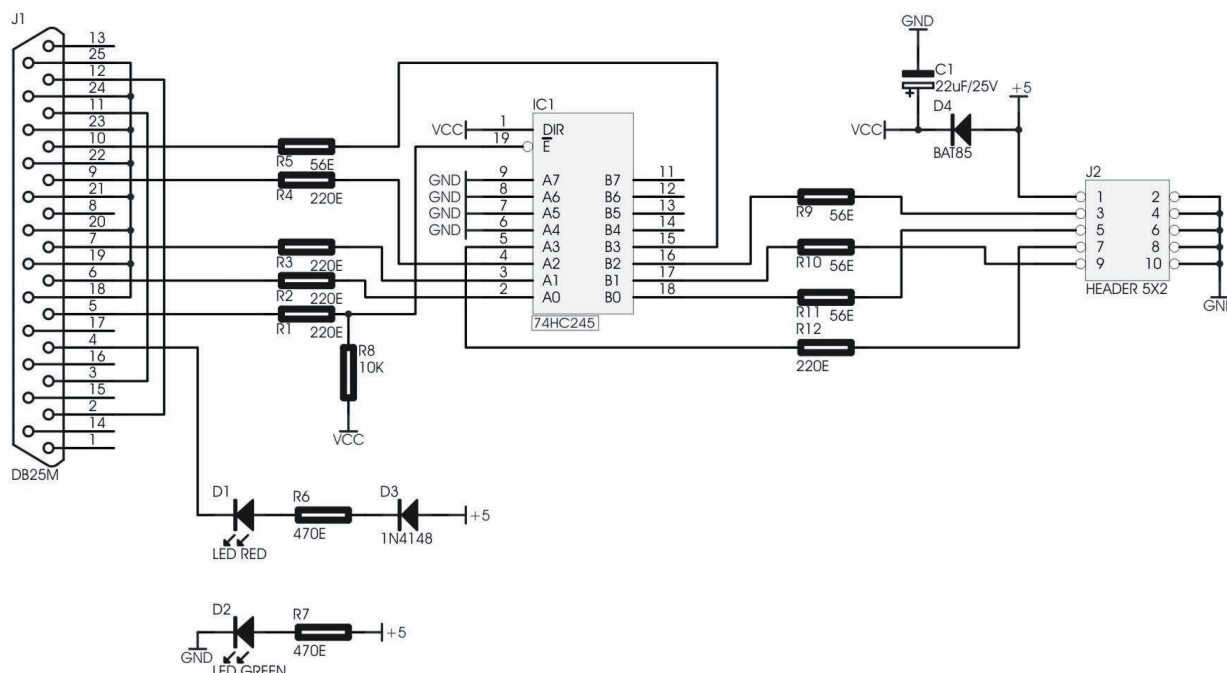
Vezje ni uporabno samo za nalaganje končne verzije kode v mikrokontroler ampak predvsem za razvoj le-te. Atmel za večino svojih mikrokontrolerov z ISP podporo zagotavlja najmanj 1000 ciklov programiranja, kar običajno zadostuje za dokončanje enega projekta. To še posebej velja, če smo pri programiranju skrbni in ne vpisujemo nove verzije, preden smo do neke mere prepričani, da spremenjeni del deluje v skladu s pričakovanji.

Bistvo delovanja vmesnika je, da ob programiranju priključi signale MISO, MOSI in SCK na nogice mikrokontrolera, v normalnem delovanju pa so izhodi v stanju visoke impedance. Potreben je še signal RESET, ki mikrokontroler sploh postavi v način delovanja, ki omogoča programiranje in istočasno postavi vse izhode le-tega v stanje visoke impedance. Na ta način se ne more zgoditi, da bi se signali vmesnika in mikrokontrolera »tepli«. Včasih imamo možnost, da ostanejo nogice MISO, MOSI in SCK nepovezane, v nekaterih primerih, pa to le ni mogoče. Takrat moramo vzeti v

zakup, da se bo elektronika, priključena na V/I linije, »pozibavala« v ritmu programirnih signalov, kar je lahko moteče, včasih pa celo življenjsko nevarno.

Glavni del vezja je IC 74HC245, dvosmerni linijski ojačevalnik. Prav tako bi lahko uporabili 74HC244, še primernejše pa bi bilo 74HC125. Vezje dobi preko sponke 5 na DB25 priključku signal, da preide iz stanja visoke impedance v aktivno stanje. Istočasno mora biti aktiviran signal RESET. Preko signalov MISO, MOSI in SCK se koda po določenem protokolu naloži v vezje. Zanki 2-12 in 3-11 sta namenjeni diagnostiki, da program zazna prisotnost vmesnika na LPT. Vmesnik se napaja iz ciljnega sistema, delno zaščito pred raznimi pozabljivostmi ob priključitvi zagotavlja Schottky dioda BAT 85 z minimalnim padcem napetosti na njej. Od originalnega vezja se moje razlikuje še po tem, da so na vhodu in izhodu zaščitni upori, ki ščitijo vezje pred prehodnimi pojavi ob npr. priključitvi vmesnika v vključeno vezje in podobno. Pozor - razpored signalov na IDC-10 priključku se razlikuje od Atmelovega (Tabela 1). Shemo pa lahko vidimo na sliki 1a.

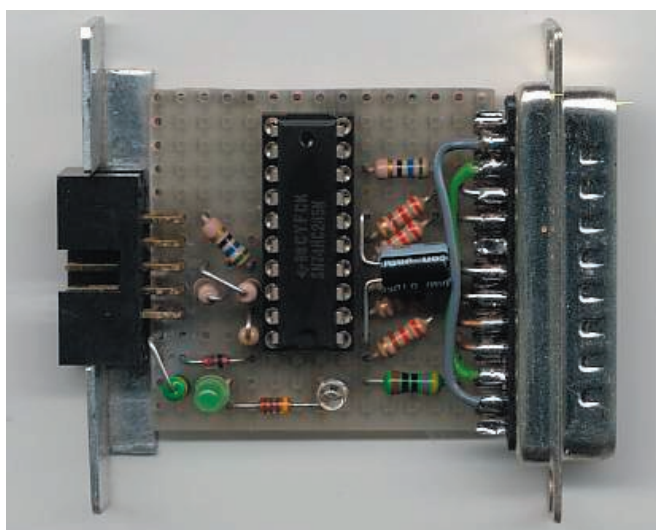
Prvo vezje (ki ga še sedaj uporabljam) sem zgradil na prototipni plošči z luknjicami (slika 2 in 3), kasneje je Jan narisal tiskano vezje zanj..



Slika 1a. Izboljšano vezje z zaščitnimi upori

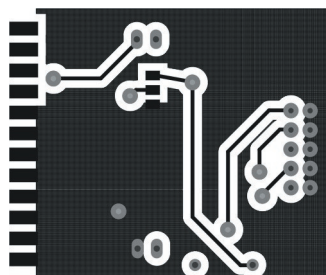
	Opis sponke
1	Napajanje +5 ali 3.3V
3	RESET
5	SCK
7	MISO
9	MOSI
	GND

Tabela 1. Razpored signalov na programirnem priključku.



Slika 2. Prototip

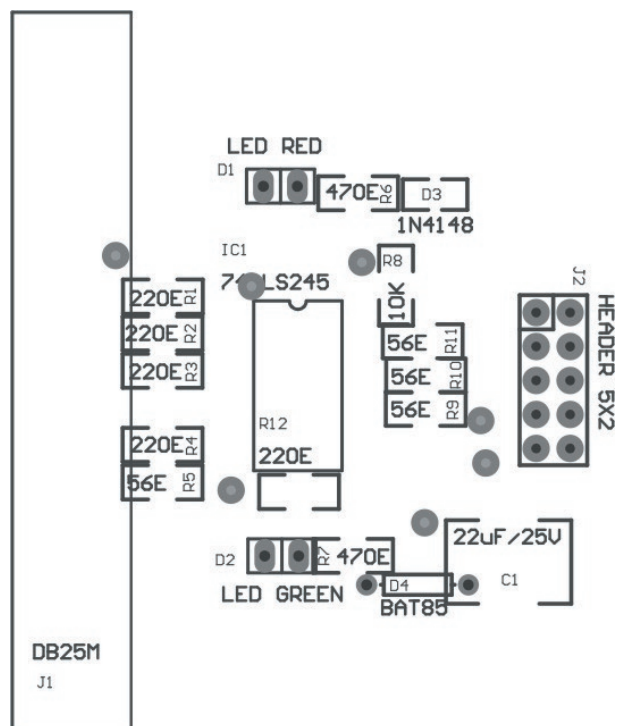
Rdeča LED prikazuje, da je vezje v stanju programiranja, zeleno pa, da ima napajanje. Dioda D3 je vezana v serijo z rdečo LED, da le'ta ne brli, ko ni aktivna.



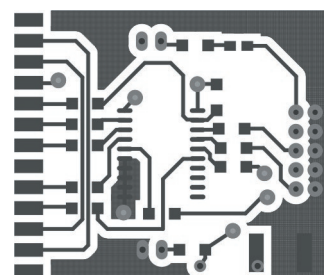
Spodnja stran tiskanine



Slika 3. Prototip v ohišju



Razporeditev elementov.



Zgornja stran tiskanine

**Tiskanina je dobavljiva preko HAM tech-letter**

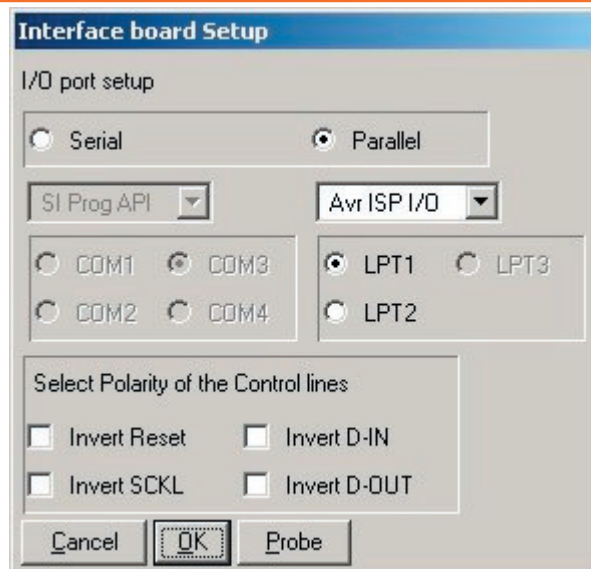
## Uporaba

Vmesnik vstavimo v DB-25 priključek za tiskalnik. S ciljnim sistemom ga povežemo s ploskim kablom. Le-ta naj bi izdelan tako, da je sponka 1 na enem koncu kabla povezana s sponko 1 na drugem koncu kabla, sponka 2 z 2 itn.

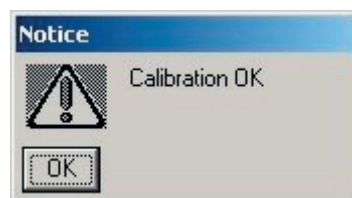
Opisal bom primer uporabe s programom PonyProg. Program mi je všeč že zato, ker deluje v vseh Wintendo operacijskih sistemih: Windows 95, Windows98, Windows NT, Windows 2000. V slednjih dveh je namreč potrebno aktivirati poseben gonilnik za direkten dostop do V/I enot, sam operacijski sistem tega direktno ne dovoljuje. Na srečo to opravi sam PonyProg.

Po zagonu programa najprej v meniju Setup/Inteface setup iberemo nastavitve za naš vmesnik (Slika 4).

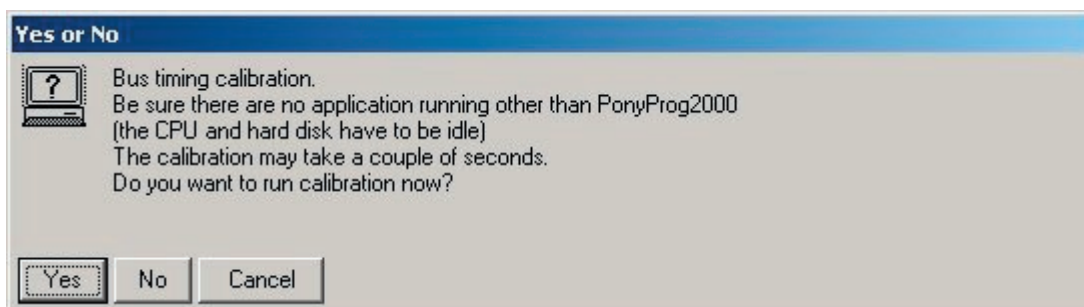
V primeru, da bi programirali vezje, pri katerem je RESET vhod aktiven v visokem nivoju (npr. 89S8252), je potrebno označiti polje Invert Reset.



Slika 4. Izbira vmesnika



Slika 6. Uspešen konec kalibracije.



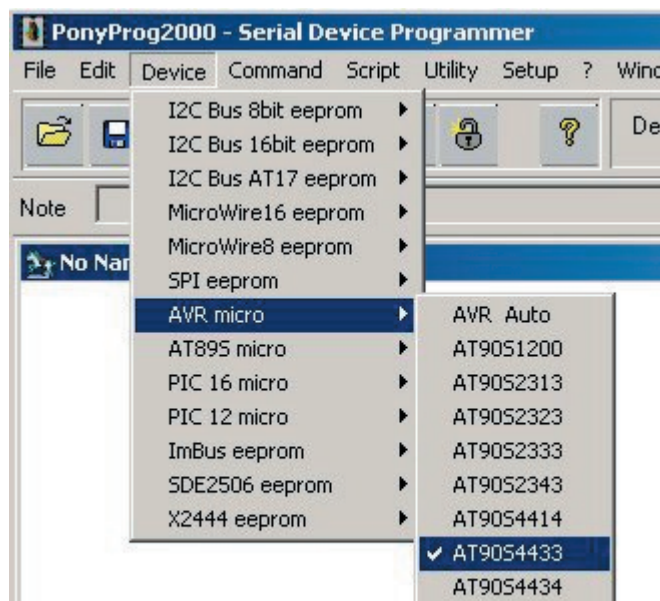
Slika 5. Kalibracija

Zatem moramo program tudi kalibrirati – prilagoditi hitrosti našega računalnika (Slika 5). To postorimo v meniju Setup/Calibration. Pomembno je, da med kalibracijo teče samo PonyProg, vse ostale programe torej začasno zapremo.

Ob koncu kalibracije nas pričaka obvestilo (Slika 6).

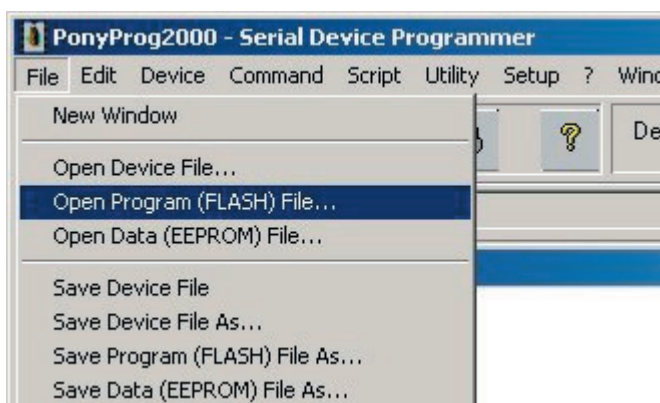
Pred nalaganjem kode v ciljni sistem moramo izbrati tip mikrokrmilnika (Slika 7).

Zatem preberemo datoteko s kodo, ki jo želimo vpisati v mikrokrmilnik. Če gre za vsebino programskega pomnilnika, izberemo File/Open Program (FLASH) File... oziroma File/Open Data (EEPROM) File... v primeru



Slika 7. Izbira ciljne naprave.

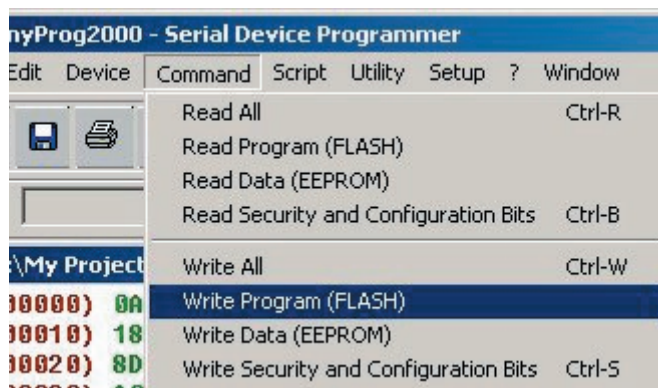
vsebine EEPROM-a (Slika 8). Program pozna HEX, MOT, BIN, ROM in druge zapise kode. Med razvojem je najenostavneje imeti



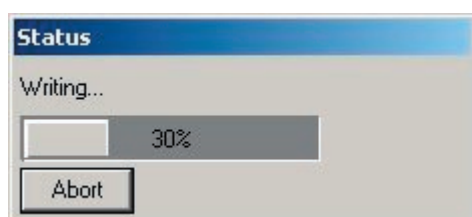
Slika 8. Izbira datoteke s programsko kodo.

PonyProg vedno aktiven, ko enkrat naložimo kodo s kombinacijo tipk Ctrl-L (Reload) naložimo novo verzijo iste datoteke.

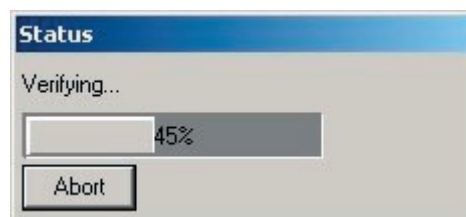
Programiranje poženemo s Command/Write Program (FLASH) če gre samo za vsebino programskega pomnilnika, s Command/Write Data (EEPROM) za vsebino EEPROM-a ali celo s Command/Write All v primeru obojega (Slika 9). Program najprej kodo



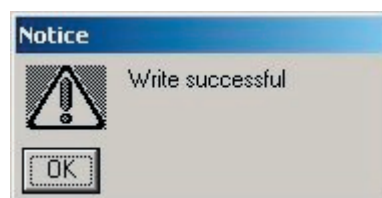
Slika 9. Zagon vpisa v ciljni mikrokrmilnik.



Slika 10. Vpis



Slika 11. Preverjanje vpisa



Slika 12. Vpis uspešen

vpiše (Slika 10), jo preveri (Slika 11) in nas obvesti o uspešnem vpisu (Slika 12).