

Einfache AVR Programmier-Interfaces und -Programme

Viele AVR-Mikroprozessoren besitzen eine eingebaute Schnittstelle zur Programmierung im eingebauten Zustand (ISP In-circuit Serial Programming). Dazu werden 4 Anschlüsse des Prozessors über einfache - meist nur passive - Verbindungen mit der parallelen oder seriellen Schnittstelle verbunden. Ein Nachbau ist also problemlos möglich.

Diese Ausarbeitung betrachtet nur zwei Interfaces mit der seriellen Schnittstelle. Die bekannteste stammt von Claudio Lanconelli und eine noch einfachere von Burkhard Kainka.

Die Anschlüsse der Prozessoren tragen folgende Bezeichnungen

Bezeichnung	Bedeutung	Anschluss beim ATtiny13	
RST oder /RST	Reset bzw. 'Not Reset' (aktiv low)	/RST	Pin 1
SCK	S erial C lock vom PC → Prozessor	PB2	Pin 7
MISO	M aster I n S lave O ut: Daten vom Prozessor → PC	PB1	Pin 6
MOSI	M aster O ut S lave I n: Daten vom PC → Prozessor	PB0	Pin 5

Beiden Interfaces gemeinsam ist die Gewinnung der zur Programmierung erforderlichen Betriebsspannung aus der seriellen Schnittstelle. Nur wenn diese nicht ausreichend hoch und niederohmig genug ist, muss eine externe Spannung - gesteuert von der intern erzeugten Spannung - bereit gestellt werden. Das kann bei Notebooks oder USB-Seriell-Interfaces der Fall sein, wobei letztere oft auch noch wegen mangelnder Geschwindigkeit nicht nutzbar sind. Da die serielle Schnittstelle von den Programmen regelrecht 'vergewaltigt' wird - es handelt sich nicht um eine übliche serielle Datenübertragung - gibt es immer dann Probleme, wenn das jeweilige Betriebssystem keinen ungehinderten Zugang zu den Steuerleitungen DTR, DSR, RTS und CTS ermöglicht oder diese Anschlüsse bei einfachen USB-seriell Wandlern gar nicht implementiert sind. Alle Anschlussbezeichnungen und Pinnummern der seriellen Schnittstelle beziehen sich auf die Stifte des DB9-Anschlusses am PC. Die Interfaces haben demzufolge eine DB9-Buchse, lassen sich also direkt am PC anstecken oder über ein 1:1-Verlängerungskabel mit Buchse und Stecker betreiben.

Claudio Lanconelli (www.lancos.com) veröffentlichte seine Interfaceschaltung **SI Prog (Serial Interface for PonyProg)** zusammen mit seiner GNU Software **PonyProg**. Sie wird von PonyProg unter den Bezeichnungen 'SI Prog API' und 'SI Prog I/O' verwendet. Auch die GNU Software **avrdude** akzeptiert diese Schaltung unter der Bezeichnung **Ponyser**. Beide Programme werden für LINUX und Windows angeboten. Erprobt habe ich nur PonyProg2000 (Version 2.07c) unter Windows 2000. Die Schaltung **AVR-PG1B** von www.olimex.com/dev ist der **SI Prog** Schaltung zum Verwechseln ähnlich. Sie verzichtet aber ganz auf die Erzeugung der Betriebsspannung aus der seriellen Schnittstelle und besitzt eine Schutzdiode vor der Basis des Reset-Transistors in Serie zu R3. R1 ist dafür 33k statt 15k Ohm (vgl. Seite 3).

Burkhard Kainka veröffentlichte seine sehr einfache Schaltung für den ATtiny13 (nur 3 Widerstände, zwei Dioden, ein low-drop 5-Volt-Regler und zwei keramische Kondensatoren) mit dem "Lernpaket Mikrocontroller" im Franzis Verlag. Es ist auf die Programmierung des ATtiny13 beschränkt. Seine dazu passende Software LPmikroISP.exe gibt es leider nur für Windows. Sie soll aber auch unter Wine auf Linux Rechnern laufen (nicht erprobt).

Signal / Attiny13-Pin	Serieller Pin bei SI Prog	Serieller Pin bei LPmikro
/RST 1 RES	3 TXD	Anschluss fehlt. Hardware-Brücke!
SCK 7 PB0	7 RTS (und 6 DSR)	3 TXD
MISO 6 PB1	8 CTS	8 CTS (und 2 RXD)
MOSI 5 PB0	4 DTR	4 DTR

Bei genauerem Hinsehen sind die Unterschiede nicht besonders groß, so dass man mit 3 Steckbrücken ein gemeinsames Interface bauen kann. Einzelheiten siehe die folgende Aufstellung und den Stromlauf auf Seite 3.

Gemeinsame serielle Schnittstellenschaltung

- SI Prog besitzt eine Verbindung von Pin 7 nach Pin 6. Damit prüft das Programm, ob das Interface vorhanden ist. Diese Verbindung stört bei LPmikro nicht, da PIN 7 nur zur Betriebsspannungserzeugung dient (wie auch bei SI Prog) und Pin 6 nicht verwendet wird. Sie kann also bleiben.
- LPmikro empfängt Signale vom PC an den parallel geschalteten Anschlüssen 8 und 2, SI Prog nur am Anschluss 8. Da der Anschluss 2 (RXD) bei SI Prog nicht verwendet wird, kann die Verbindung zwischen Pin 8 und Pin 2 auch bei SI Prog bleiben, so dass es damit keinen Unterschied (mehr) gibt.
- Der Anschluss 4 DTR wird von beiden Interfaces verwendet, stimmt also überein.
- Das Signal SCK muss durch eine Steckbrücke zwischen Pin 3 (LPmikro) und Pin 7+6 (SI Prog) umgeschaltet werden können.
- Das Reset Signal stammt bei SI Prog vom Pin 3 und wird mit einem Transistor invertiert. Bei LPmikro fehlt eine Steuerung des RES Pins des ATtiny13. Es wird statt dessen eine Steckbrücke verwendet, die den Anschluss RES des ATtiny13 zum Programmieren auf Masse legt und bei Betrieb offen lässt. Da LPmikroISP die zur Programmierung nötigen Resets de facto durch Aus- und Einschalten der Betriebsspannung mit Pin7 generiert und Pin 7 in der kombinierten Hardware den Resettransistor steuert, kann auch unter LPmikroISP.exe der Resettransistor benutzt werden. Dann darf auch die externe Betriebsspannung ständig eingeschaltet sein (Brücke JP2 in Stellung 3-4). Die Steckbrücke muss nur zum normalen Betrieb geöffnet werden, wenn an Pin 7 (RTS) weiter eine positive Spannung steht.

Da die Signale der seriellen Schnittstelle Standardpegel von +3..+10V und -3..-10V haben, der ATtiny13 aber nur 0..+5V verträgt, müssen die Spannungen begrenzt werden.

SI Prog benutzt dazu Widerstände von 4,7k und 5,1-Volt-Z-Dioden, die gleichzeitig beide Pegel begrenzen. LPmikro verwendet 100k-Widerstände und verlässt sich auf die in den ICs vorhandenen Schutzdioden. In der Signalrichtung vom IC zum PC verwendet LPmikro einen Schutzwiderstand von 1k, während SI Prog auf eine Direktverbindung setzt, um den Spannungshub möglichst nicht zu verkleinern, denn der Eingangswiderstand der Seriellen Schnittstelle liegt bei rund 3k Ohm (sowohl am Pin8 als auch am Pin 2). Streng nach den IEC und IEEE RS232-Normen müsste der Low-Pegel < -3V betragen. Fast alle Interfaces sind aber mit 0V zufrieden - verlassen kann man sich aber nicht (immer) darauf!

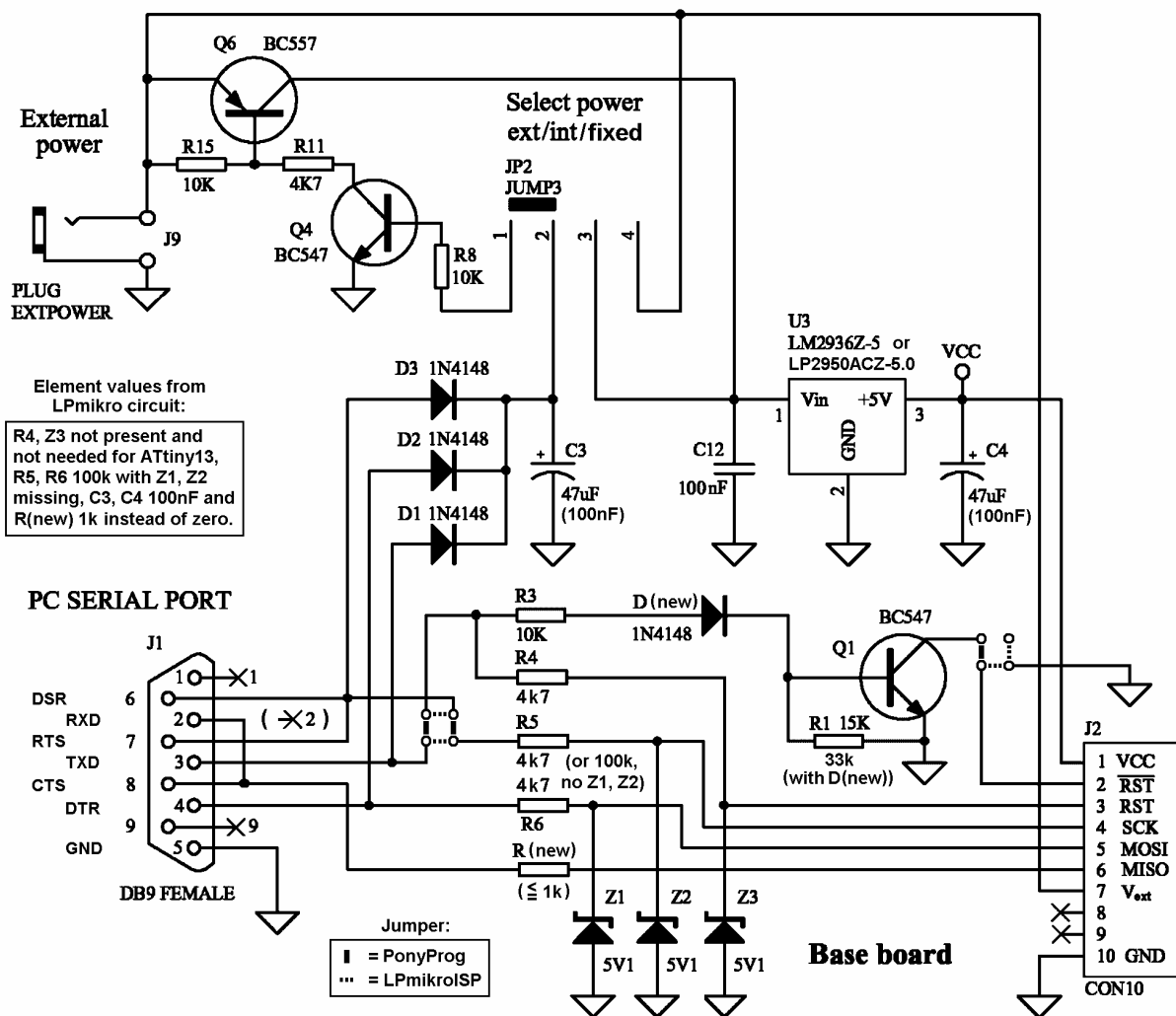
Hinweise zur Interfaceschaltung (vgl. Seite 3)

Die Schaltung basiert auf SI Prog und enthält folgende Erweiterungen:

- JP2 mit zusätzlicher Stellung 3-4 zur Nachbildung der AVR-PG1B-Lösung. Bei PonyProg kann die Betriebsspannung in dieser Stellung ständig anliegen, bei LPmikro dagegen nicht - es sei denn, man verwendet das Resetsignal von Q1 (siehe oben).
- Diode in Serie zu R3 (R1=33k) nach AVR-PG1B (Sicherheit gegen Überschreitung der Basis-Emitter-Sperrspannung von Transistor Q1)
- Verbindung 2-8 am seriellen Anschluss nach LPmikro Vorlage
- R(neu)=1k in der MISO-Leitungsverbindung nach LPmikro Vorlage
- Hinweis auf alternativen Baustein zu U3
- Steckbrücken zur Auswahl des Brennprogrammes PonyProg / LPmikroISP

Hinweis:

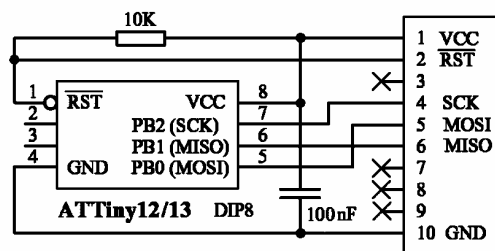
Die Originalschaltungen von LPmikro, AVR-PG1B und PonyProg finden sich im Anhang.



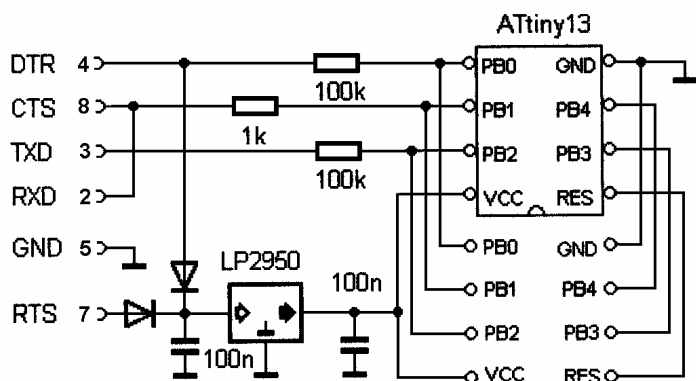
Extracted from

Title	SI Prog - Serial Interface for PonyProg	
Number	http://www.LancOS.com	Revision 2.2
Date:	29-Jun-2000	
File:	C:\Document1\Protel\PonyProg2.2\SI_Prog2-2c.ddb	

Edited and expanded by DH2MIC 23-Mar-2011 with:
External power fixed, Diode in series to R3 with R1=33k,
alternate Part for U3 and for LPmikroISP compatibility:
additional connection 8 - 2 on J1 (pin 2 previously unused)
and Jumpers between SI Prog circuit and LPmikro circuit.



Auszug aus AVR AT90Sxx adapter

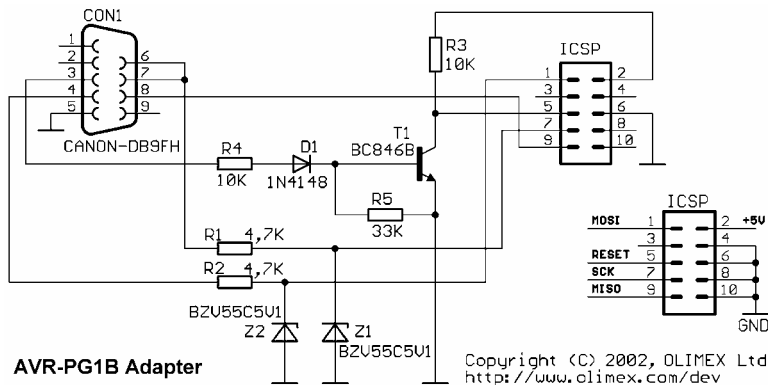


Oben:

Das umschaltbare Interface für den ATtiny13

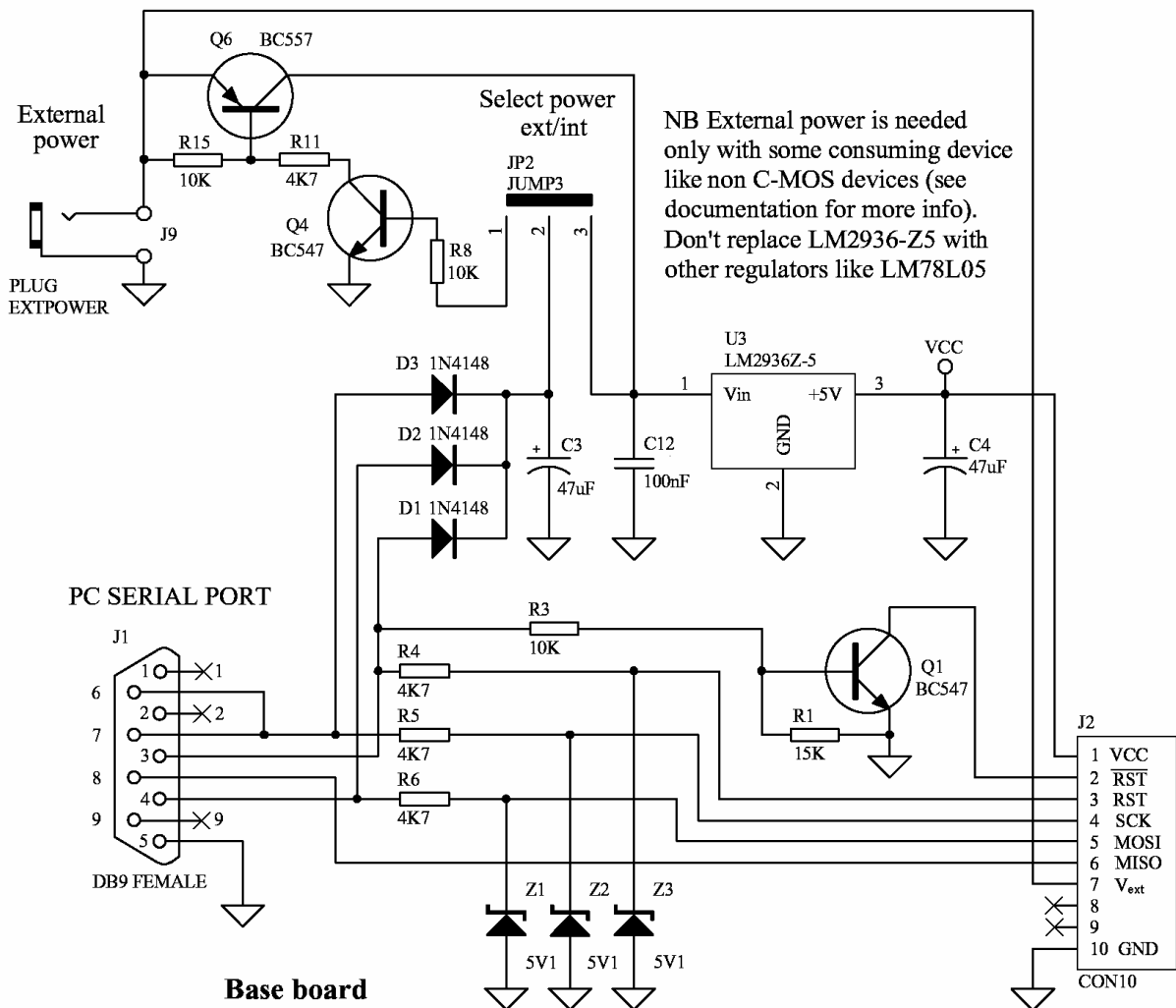
Links:

Schaltung des LPmikro Interfaces



Links:
AVR-PG1B Adapter

Unten:
Auszug aus SI_Prog2-2.pdf



Auszug aus

Title		
SI Prog - Serial Interface for PonyProg		
Number	http://www.LancOS.com	Revision
		2.2
Date:	29-Jun-2000	
File:	C:\Documenti\Protel\PonyProg2.2\SI_Prog2-2c.ddb	

