

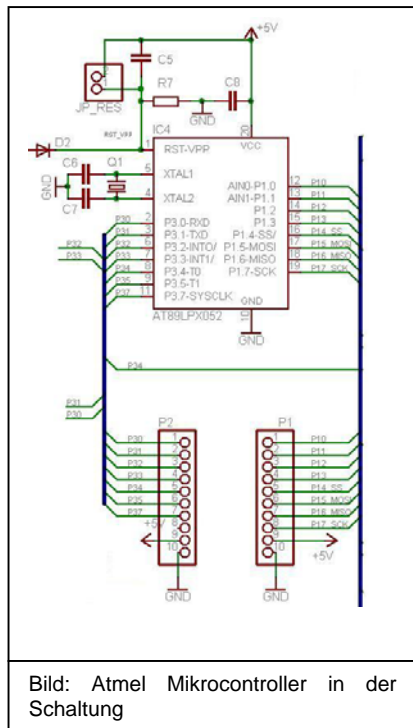
6.0 Der minimale Aufbau der Platine.

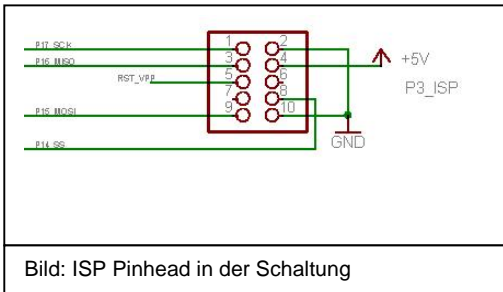
Die wenigen Komponenten für die ersten Versuche lassen sich schnell bestücken. Falls Sie keine Erfahrungen im Umgang mit dem Löten von SMD Bauelementen besitzen, können Sie im Internet genügend Informationen finden. Hilfreich sind dabei eine Pinzette, ein Lötcolben mit 0,4 mm Spitze und Entlötlitze. Nur beim IRF FET im SO-8 Gehäuse und den 5 Pins des Maxim D/A Wandlers ist die Bestückung ein wenig kniffliger. Bei der Entwicklung des Layouts wurde daher für die einfache Bestückung reichlich Platz um die einzelnen SMD Bauelemente gelassen.

Nach jedem Schritt kann die Funktionalität mit Testprogrammen überprüft werden.

6.1 Aufbau mit Atmel Mikrocontroller und ISP

Die Schaltung des Controllers und der ISP Anschluss sind einfach. Es werden ein Quarz und 2 Kapazitäten für den Schwingkreis benötigt. Der Reset wird über C5 erzeugt. Im Layout wurde der Widerstand R7 nach Masse berücksichtigt, der nicht bestückt werden muss. Die Ports werden über die Anschlüsse P1 und P2 herausgeführt.





Der ISP Pinheader braucht nur bestückt zu werden. Atmel AT89S-Fachleute erkennen vielleicht die notwendige Erweiterung mit dem SS Signal.

Eine ISP Beschreibung finden Sie auch auf den Atmel Internetseiten (www.atmel.com. Suche Sie nach AT89ISP).

Nachbau:

Benötigen Sie im ersten Schritt nur den Mikrocontroller mit dem ISP Programmierstecker, dann müssen Sie auch die Bauelemente für die Spannungsstabilisierung bestücken.

Mit einem Atmel AT89ISP- oder dem GMS PPS Programmieradapter können Sie das erste Programm zum Mikrocontroller senden.

Folgende Bauteile müssen auf der Platine bestückt werden:

Die 2pol Schraubklemme für conn_PWR, C15-C18, IC_IS 7802, R22 und die LED_PWR für das Netzteil. Eine (gedrehte) 20 pol Fassung, C5-C8, Q1 (11.059MHz) und, ggf. für einen externen Reset, einen 2 poligen Pinheader, und den 2*5Pinhead P3_ISP für eine ISP Programmierung des Atmel Mikrocontrollers.

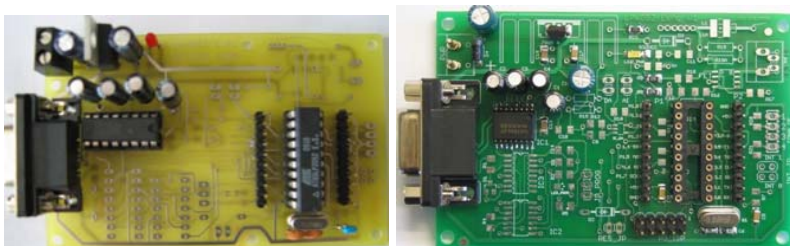


Bild: Eine minimale Bestückung der ersten Prototypen ohne Ser-ISP Programmierlogik

Die notwendige Software für die ISP Programmierung können Sie im Internet bei www.atmel.com (suche dort nach At89LP2052) kostenlos erhalten.

6.2 SPI – ISP - IDE

Die SPI (**S**erial **P**ort **I**nterface) Schnittstelle dient gleichzeitig als Programmierschnittstelle für die serielle ISP (**I**n **S**ystem **P**rogrammierung) des AT89LPx052. Dieses, mittlerweile übliche Verfahren, gibt einige Vorteile für die Entwicklungsumgebung IDE (**I**ntegrated **D**evelopment **E**nvironment), da auf ein teures Programmiergerät verzichtet werden kann und der Mikrocontroller nicht mehr umgesteckt werden muss. Nun ja, es ist nicht ganz richtig. Wurde bei diesem Atmel AT89LPx052 Mikrocontroller ISP deaktiviert, dann hilft nur noch die parallele Programmierung weiter, die sich im Übrigen auch etwas gegenüber den Vorgängern geändert hat.

Der Mikrocontroller kann also für die Programmierung im laufenden Betrieb angehalten, neu programmiert und wieder gestartet werden.

Figure 22-27. ISP Command Sequence

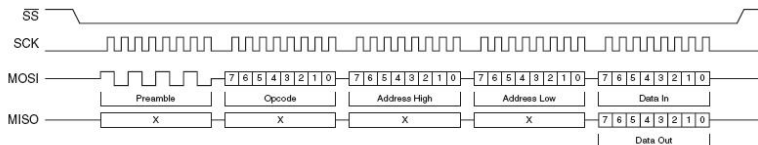


Bild: Ein Auszug aus den Atmel Spezifikationen: In dieser Signaldarstellung, aus dem vorläufigen Atmel AT89LPx052 Handbuch, fehlt das RST-Vpp Programmiersignal (Pin1) das während der ISP Start Sequenz aktiviert wird.

Dazu ist eine vorgegebene Reihenfolge einzuhalten. Nach der ISP Start Sequenz, die zur Aktivierung der seriellen ISP Programmierung im laufenden Betrieb dient, kann die Programmierung mit der Sequenz AAH, ACH, 53H eingeleitet werden. Dann können bis zu 32 Datenbytes in einem Zyklus (Page), beginnend mit einer Präambel (AAH), dem entsprechenden Opcode, den Adressen und den Daten übermittelt werden. In einer Page wird nach jedem Datenbyte intern der Adresszähler automatisch um eins erhöht. Die ‚MOSI‘ Daten müssen beim Schreiben bei der steigenden Flanke des SCK Signals anliegen. Zu lesende ‚MISO‘ Daten liegen unmittelbar nach der fallenden Flanke

des letzten SCK Signals an. Diese Daten liegen am CTS Eingangssignal der seriellen Schnittstelle an.

Im Gegensatz zu bestehenden Atmel ISP Anbindungen ist die Unterstützung des SS Signals (Pin 8 des erweiterten ISP Pfostensteckers) für die ISP Programmierung erforderlich, da der Mikrocontroller als Slave agiert. Das Signal muss über einen gesamten Zyklus aktiv ("L") sein. Liegt das SS Signal auf "H" dann ist der SPI Port des Slaves deaktiviert. In diesem Fall kann P1.5/MOSI als Eingang genutzt werden.

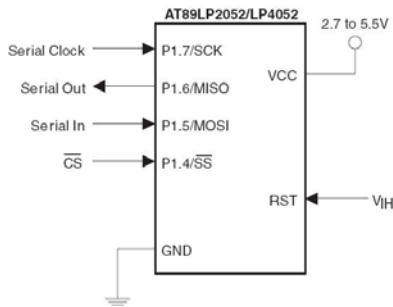
Ist der Atmel Mikrocontroller ein Master dann kann P1.4 mit dem SS Signal als Ein- oder Ausgang verwendet werden. Als Master können SPI Taktfrequenzen im SPCR Register (D5H) eingestellt werden. Die Bits SPRO und SPR1 erlauben dazu die Teilung der Quarzfrequenz durch 2, 8, 32 und 64.

Serielle und parallele Programmierung unterstützen die gleichen Kommandos. Lediglich die Hardware ist unterschiedlich.

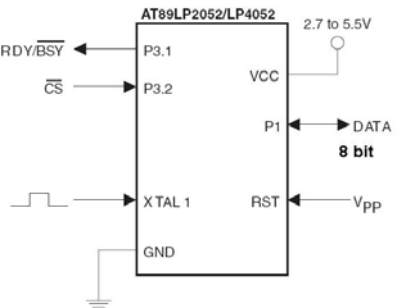
Die parallele Programmier benötigt 10 Signale (P1, XTAL1 und CS) die serielle dagegen lediglich die 4 SPI Signale.

Auszüge aus dem vorläufigen Atmel Datenblatt:

ISP/Serial Programming Device Connections



Flash Parallel Programming Device Connections



Auch die erforderlichen Programmierspannungen an Pin 1 (RST) des Mikrocontrollers sind unterschiedlich. Für die parallele Programmierung sind 12 V (V_{pp}) erforderlich.

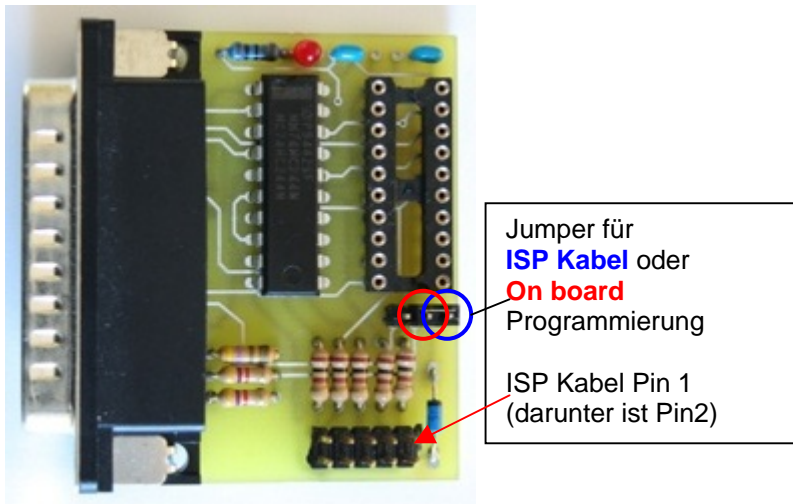
6.3 Beschreibung Parallel Port - ISP Programmieradapter

Zur 'In System Programmierung' des AT89LPx052 Flash über die SPI Schnittstelle kann ein Adapter für den Parallel Port verwendet werden, welcher die Signale des Parallel Ports für die SPI Schnittstelle entsprechend umsetzt.

Ein (möglichst kurzes) 10adriges Kabel mit Pfostensteckern verbindet den Programmieradapter mit einem vorhandenen Board (mit ISP Steckverbindung). Achten Sie bei der Verbindung auf die richtige Polung, da der Adapter über das Kabel mit Spannung versorgt wird und bei vertauschten Anschlüssen der Adapter wahrscheinlich zerstört wird.

Für die Atmel LP Familie sind Adapter von Atmel und GMS Bentheimer Softwarehaus GmbH verfügbar (Stand Nov 2005). Vorsicht bei anderen Adaptern. Nicht alle am Markt befindlichen, 'Atmel kompatiblen', Adapter unterstützen das erweiterte SS Signal des AT89LPx052.

Die Datenübertragung erfolgt mit der Original Atmel Software (AT89ISP) oder auch mit GMS Flash Software.



Der GMS PPS Adapter kann, alternativ zur Datenübermittlung über ISP zu einem vorhandenen Board, einen AT89LPx052 im DIP Format auch direkt auf dem Adapter programmieren. Für diesen Fall braucht nur ein Jumper auf dem Board umgesteckt zu werden. Diese 'On Board' Programmierung wird nur von der GMS Flash Software unterstützt.

Bei der Atmel LP Familie besteht die Möglichkeit die Programmierfunktion über ISP generell zu deaktivieren. In diesem Fall hilft nur noch eine 8bit parallele Programmierung!

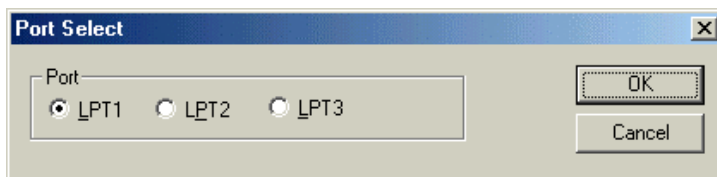
Selbstverständlich können auch am Markt befindliche EPROM Programmiergeräte eingesetzt werden.

6.4 Beschreibung der Flash Programmiersoftware

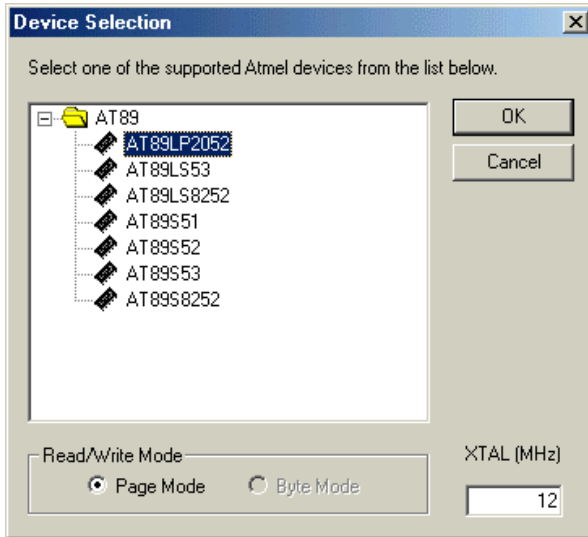
Nach der Erstellung einer Mikrocontrollersoftware werden die Daten im Hex oder Bin Format erzeugt und mit Flash Software in den nicht flüchtigen Speicher des Atmel AT89LPx052 übertragen.

Die Atmel ISP Software können Sie im Internet bei www.atmel.com (suche dort nach At89LP2052) kostenlos erhalten. Diese Software unterstützt auch den GMS PPS Programmieradapter.

Wenn Sie die Atmel Software und das letzte Update richtig installiert, einen Adapter angeschlossen und die Kabelverbindung richtig herum gesteckt haben, müssen Sie in der Software als erstes den richtigen Parallel Port wählen:



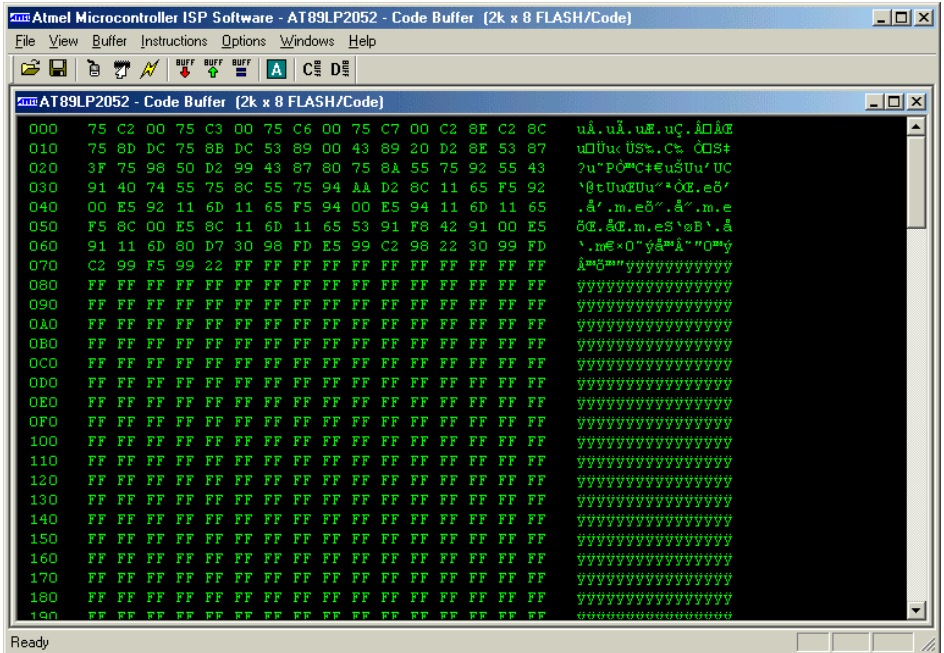
Als nächstes müssen Sie den richtigen Baustein wählen:



Danach können Sie eine erfolgreiche Verbindung testen indem Sie den Adapter initialisieren:

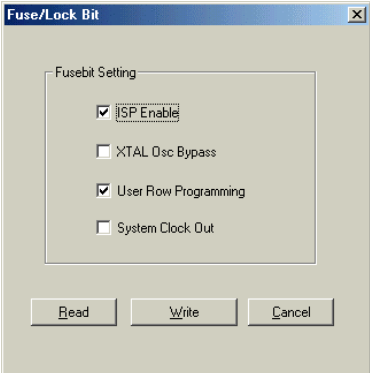


Mit ‚Datei Öffnen‘ können Sie Ihre Hex Datei wählen, laden und das Ergebnis wird Ihnen im Code Buffer Fenster angezeigt:



Mit dem roten BUFF Pfeil können Sie dann den Hex Code zum Mikrocontroller übermitteln.

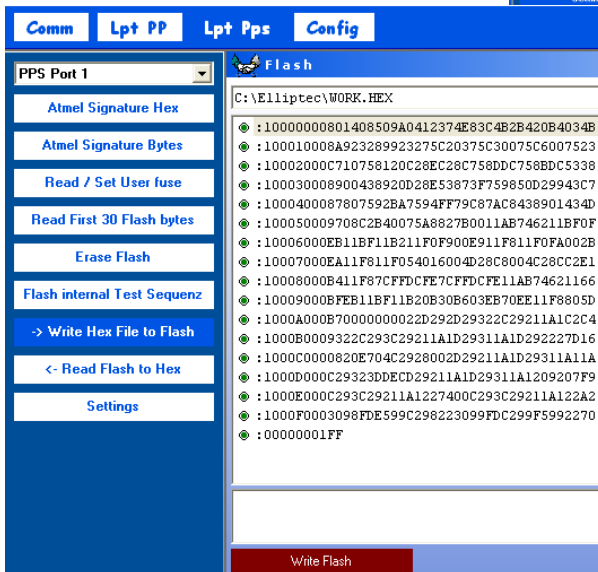
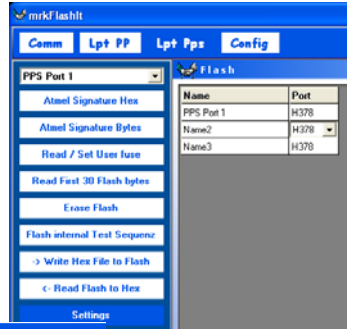
Geht nicht? Dann gibt es vielleicht ein Problem mit den Einstellungen. Kontrollieren Sie ob ISP auch aktiviert ist. Nein? In diesem Fall hilft nur noch eine 8bit parallele Programmierung.



Möchten Sie einen Atmel AT89LPx052 im Dual Inline Package auf dem PPS Adapter programmieren, dann benötigen Sie die GMS Flash Software. Die GMS Flash Software unterstützt ebenfalls die serielle-ISP Programmierung des Ghost Boards.

Hier einige Auszüge zur ersten GMS Flash Version:
 Für den GMS Parallel Port PPS Adapter wählen Sie zuerst *Lpt Pps*, da diese Software auch für eine serielle –ISP Programmierung verwendet werden kann.

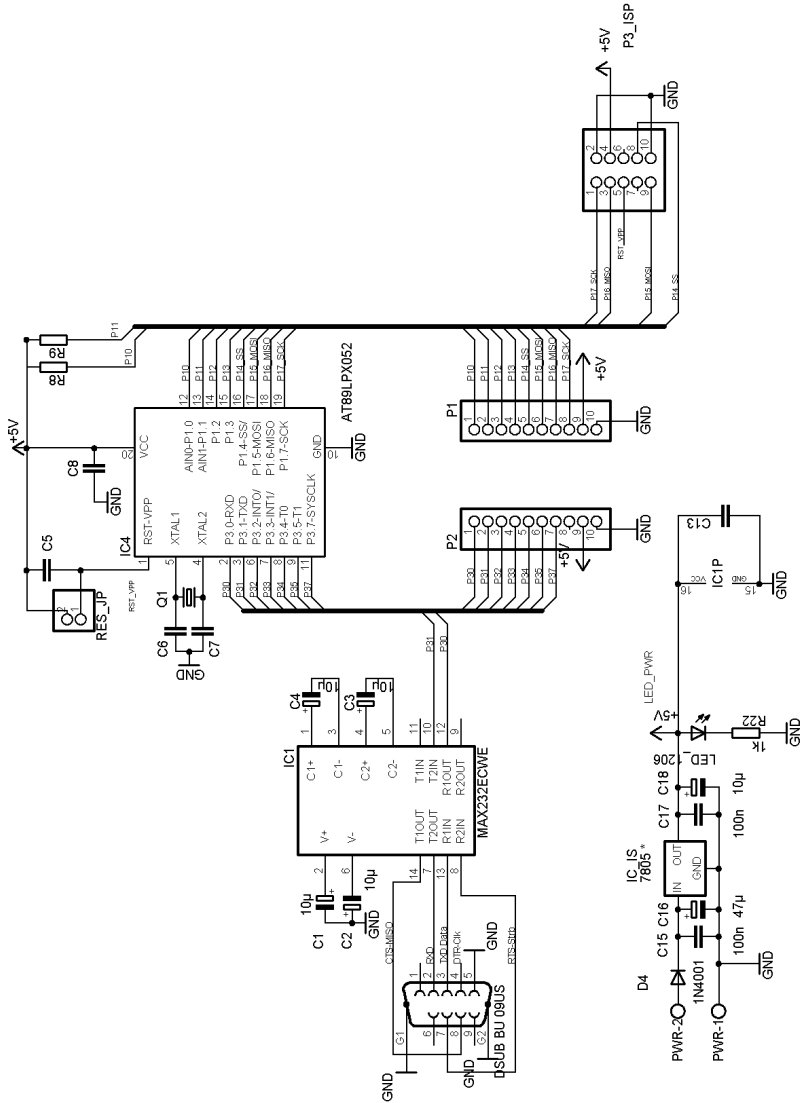
In *Settings* können Sie für den Port einen Namen vergeben und die Adresse des Ports einstellen.



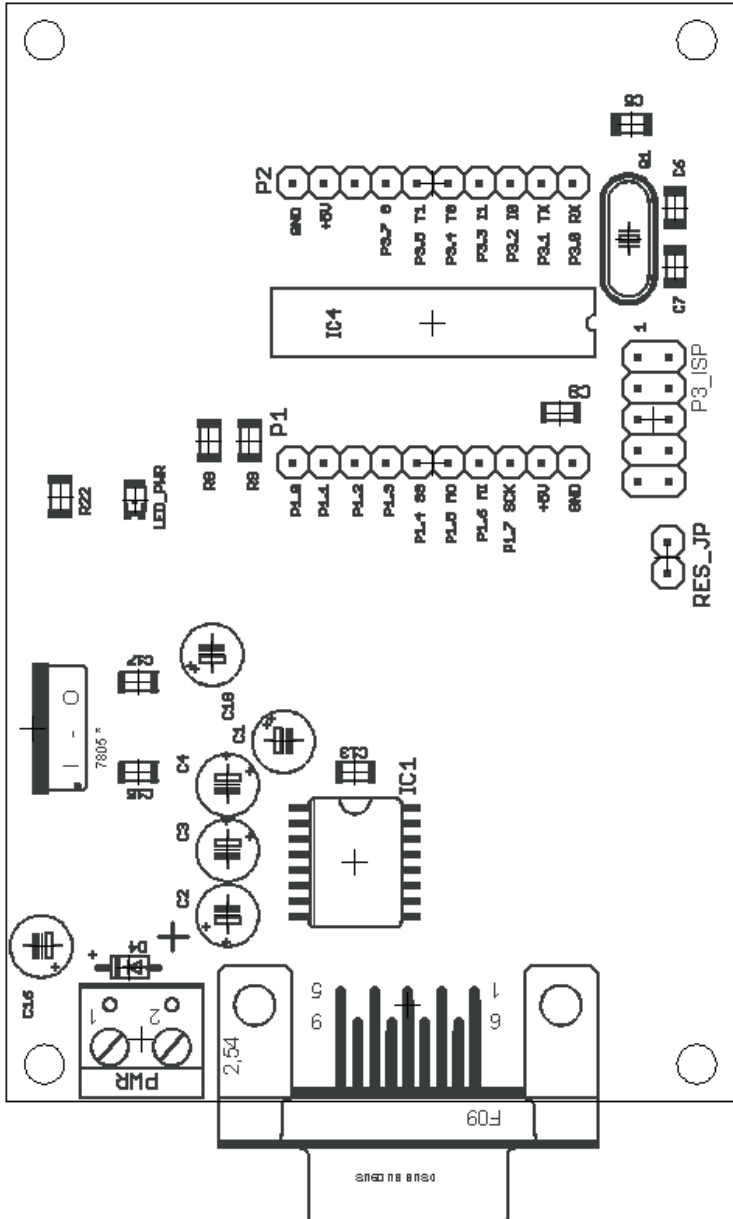
Eine Hexdatei können Sie im Bereich *Write HexFile to Flash* auswählen und danach in einen Buffer laden. Wie bei der Atmel Software wird Ihnen der Hexcode (und die Speicheradressen) in einem gesonderten Fenster angezeigt.

Mit 'Write Flash' wird der Buffer in den Flash des AT89LPx052 übermittelt.

6.5 Schaltbild der minimalen Bestückung



6.6 Bestückung der minimalen Ausführung



6.7 V24 Schnittstellenkabel und SPI Programmierung

Ein dreiadriges Kabel für die serielle Schnittstelle ist für ein Software Handshake aber nicht mehr die serielle-ISP Programmierung ausreichend, da die Signale DTR, RTS und CTS der seriellen Schnittstelle verwendet werden.

Sie müssen in diesem Fall ein 9-poliges, nicht-gekreuztes Verbindungskabel verwenden.

SENDER PC	Signal	EMPFÄNGER Board	Beschreibung
1	CD	1	Data Channel Received Line Signal Detector
2	RxD	2	Empfangsdaten Receive Data
3	TxD	3	Sendedaten Transmit Data
4	DTR	4	Data Terminal Ready
5	GND	5	
6	DSR	6	Sendedaten vorh. Data Set Ready
7	RTS	7	Sendeaufforderung Request to send
8	CTS	8	Sendebereitschaft Clear to send
9	RI	9	Ring indicator

Die Pins 1,6, und 9 werden nicht verwendet.

6.7.1 Die serielle - SPI Programmierung.

Die Erweiterung mit den Komponenten für die serielle Programmierung ist ebenfalls sehr einfach.

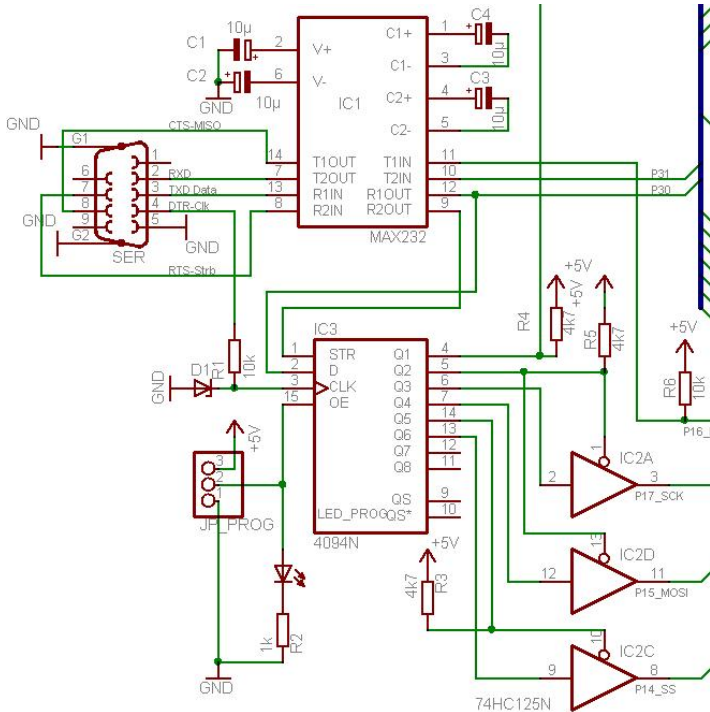


Bild: serieller Anschluss und SER-ISP Programmierlogik

Neben den Signalen RxD und TxD werden die Signale DTR, RTS und CTS der seriellen Schnittstelle verwendet.

Die letzteren Signale werden für die serielle ISP Programmierung über die SPI Schnittstelle des Mikrocontrollers benötigt.

Am seriellen Anschluss stehen nur 3 Ausgänge zur Verfügung, für die Programmierung werden aber 4 benötigt (Vpp, SCK, MOSI und SS). Daher werden die Daten und Informationen zur Treiberaktivierung zunächst in ein HC4094 Schieberegister abgelegt und dann über den HC125 Treiber (mit Tri-State Funktionen) dem Prozessor zugeführt.

R3 bis R5 sorgen für einen definierten Zustand des HC125 wenn der HC4094 deaktiviert ist.

Für das CLK Signal stand am MAX 232 kein Ausgang mehr zur Verfügung und wurde daher mit einer Z-Diode realisiert.

Der Jumper JP_Prog muss für die Programmierung auf die Pins 2-3 gesetzt werden. Dadurch werden die Ausgänge des Schieberegisters freigeschaltet und die Programmiermöglichkeit freigegeben. Dieser Zustand wird mit der LED angezeigt.

Nur wenn nach der Programmierung eine serielle Kommunikation das RTS Signal wechselt ist der Jumper wieder zurück zu setzen, da sonst vielleicht Daten übernommen werden, die den HC125 Treiber wieder aktivieren.

Die Schaltung hätte ggf. auch nur mit einem Treiber realisiert werden können, jedoch verlangen derzeit die vorläufigen AT89LP2052 Spezifikationen nach der Programmierung eine besondere ISP Exit Sequenz, die mit dieser Schaltung berücksichtigt wurde.

Der Nachteil dieser Schaltung ist die ca. 8 fache Verringerung der Programmiergeschwindigkeit gegenüber der Standard ISP Programmierung, da jedes Bit zunächst in das Schieberegister getaktet werden muss.

Folgende Bauelemente sollten für die serielle Kommunikation und Programmierung bestückt sein:

D-Sub F09HP, Sockel für IC1-IC3, C1-C4, R1-R6, D1, D2, LED_PROG und den Jumper JP_PROG und, natürlich, die IC's 1-3.

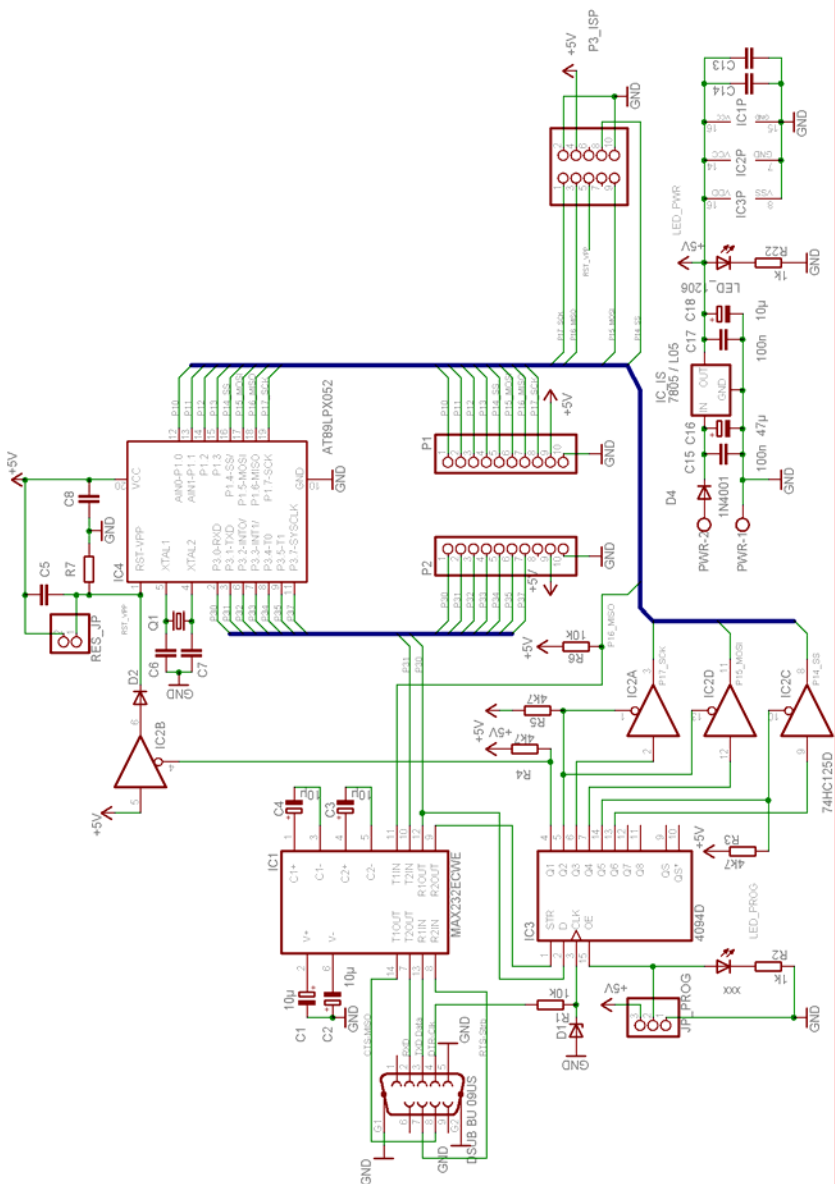
Sind die Bauelemente bestückt können Sie die GMS Programmier Software aus dem Internet bei www.gms2000.de (Bereich HW Projekte) downloaden.

Verbinden Sie Ihren Rechner über die serielle Schnittstelle mit dem AT89LPx052 Entwicklungsboard und starten Sie die Software. Die Software benötigt die beigefügten dll's für die serielle Schnittstelle um direkt auf die serielle Hardware zugreifen zu können. Eventuell müssen die dll's registriert werden.

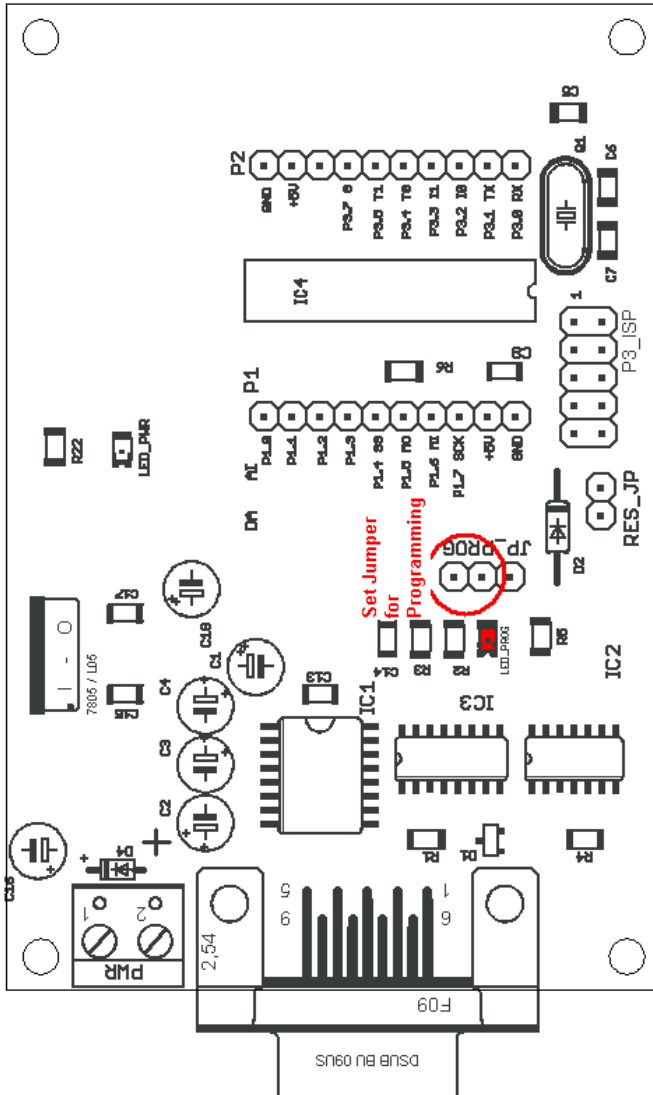
In der Software können Sie zunächst zu Testzecken die internen Atmel Bezeichnungen des Mikrocontrollers auslesen.

Im Erfolgsfall können Sie als nächstes probieren den Mikrocontroller mit dem internen Testprogramm zu programmieren.

6.8 Schaltbild, Bestückung und Jumper mit SER-ISP Logik



Set Jumper JP_Prog to enable / disable Programming



Aus Produktionsgründen (Herstellung für die Ausführung mit der Elliptecmotor Steuerung) **können auf dieser Platine weitere, passive Bauteile vorhanden sein.**

6.9 Beschreibung Testprogramm für die SER-ISP Flash Software

Das interne Testprogramm der Flash Software dient zur schnellen Überprüfung funktionsfähiger Hardware. Das Testprogramm erzeugt in einer Schleife Ausgaben am Port1 des Mikrocontrollers.

Nach einem Reset kann man mit einem Oszilloskop die Frequenzen der Frequenzteilung an Port 1 des Mikrocontrollers messen.

Der Assembler Quelltext, der einfach in einem Texteditor eingegeben wird, ist in Fettschrift auf der rechten Seite abgebildet. Die Angaben auf der linken Seite zeigen die Ergebnisse nach der Übersetzung mit dem Assembler TASM in der vom TASM erzeugten *.lst Datei.

Am Anfang jeder Zeile ist die Speicheradresse angegeben. Die Zahlen in Kursivschrift zeigen Maschinencodes des jeweiligen Befehls.

Mit **.org 0000H** wird die Anfangsadresse angegeben. Die nächsten 4 Zeilen setzen die Ports des Mikrocontrollers auf quasi bidirektional, eine Angewohnheit gleich alle Ports zu setzen, denn Port 3 zu setzen wäre hier eigentlich nicht notwendig da dieser Port nicht verwendet wird. Ohne die Angaben zum Port 1 würde man allerdings an Port 1 nichts messen können, da der Controller nach dem Reset auf Input steht.

Und dann geht es in drei verschachtelte Schleifen, die mittleren zwei davon, um die Ausgabe zu verzögern. Der Inhalt des Accumulators, der vor jedem **sjmp next** um eins erhöht wird, wird dabei zyklisch an Port 1

ausgegeben.

```

                                .org 0000H
0000  75 C2 00      main  mov   0c2H,#00H
0003  75 C3 00      mov   0c3H,#00H
0006  75 C6 00      mov   0c6H,#00H
0009  75 C7 00      mov   0c7H,#00H
000C  74 00         mov   a,#00
000E  F5 90      next  mov   P1,a
0010  7A FF         mov   r2,#255
0012  7B 14      loop1 mov   r3,#20
0014  DB FE      loop2 djnz  r3,loop2
0016  DA FA         djnz  r2,loop1
0018  04         inc   a
0019  80 F3         sjmp next
001B         .end

```

Das GMS Flash Programm wurde in Visual Basic entwickelt. Die Programmierung für die Testsequenz sieht vereinfacht wie folgt aus. Der Maschinencode wird nach der Initialisierung der Startsequenz byteweise mit der internen Funktion *datasend* übermittelt. Die notwendigen Sequenzen für die Programmierung können den Atmel AT89LPx052 Spezifikationen entnommen werden.

datasend &HAA	' Startsequenz
datasend &HAC	
datasend &H53	
toggle_SS	'SS erneut setzen
datasend (&HAA)	' send preamble
datasend (&H50)	' send opcode
datasend (&H0)	' send AdressHigh
datasend (&H0)	' send Adress
datasend (&H75)	' set Port outputs
datasend (&HC2)	
datasend (&H0)	
datasend (&H75)	
datasend (&HC3)	
datasend (&H0)	
datasend (&H75)	
datasend (&HC6)	
datasend (&H0)	
datasend (&H75)	
datasend (&HC7)	
datasend (&H0)	
datasend (&H74)	' testprogram
datasend (&H0)	
datasend (&HF5)	
datasend (&H90)	
datasend (&H7A)	
datasend (&HFF)	
datasend (&H7B)	
datasend (&H14)	
datasend (&HDB)	
datasend (&HFE)	
datasend (&HDA)	
datasend (&HFA)	
datasend (&H4)	
datasend (&H80)	
datasend (&HF3)	
datasend (&H0)	