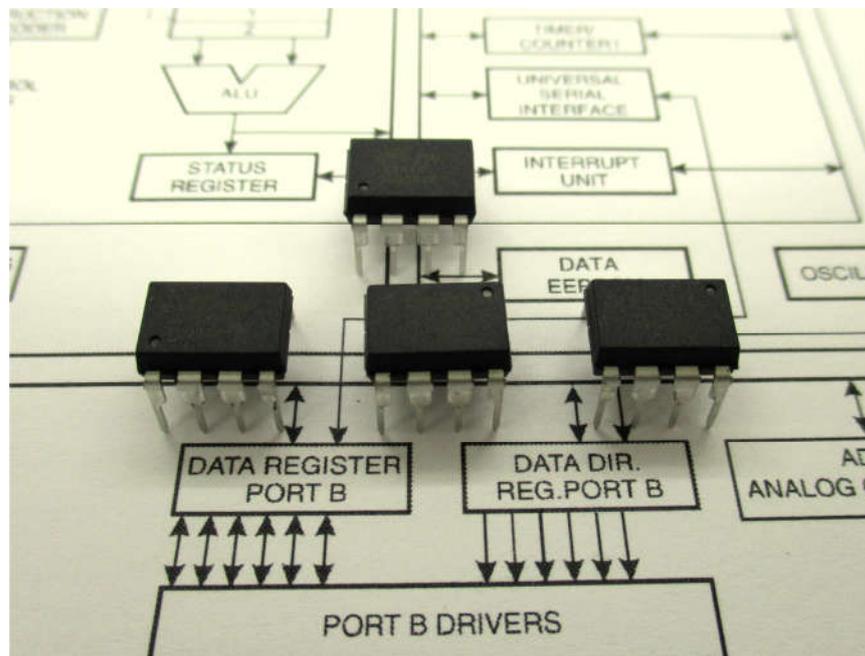


# Mikrocontroller-Praxis

## ATtiny85



Burkhard Kainka

Copyright © 2021 Burkhard Kainka  
Alle Rechte vorbehalten.  
ISBN: 97xxxxxxxxxxxxx  
Independently published

## Vorwort

Dieses Buch basiert im Kern auf dem Franzis Lernpaket Mikrocontroller. Es ermöglicht Ihnen einen einfachen und schnellen Einstieg in die Welt der Mikrocontroller am konkreten Beispiel des ATtiny85. Sie benötigen keine Vorkenntnisse und können gleich durchstarten. Die verwendete Platine mit USB-Anschluss kann bei AK Modul-Bus bezogen werden. Sie können wahlweise mit dieser Platine arbeiten oder alles auf einer Steckplatine aufbauen. Beide Alternativen werden im Detail erläutert.

Mikrocontroller sind nichts anderes als vollständige kleine Computer mit Recheneinheit, Speicher, Schnittstellen und allem was sonst noch dazu gehört. Die neuere Entwicklung hat dazu geführt, dass immer mehr in einen kleinen Chip gepackt wurde. Ein achtbeiniges IC wie der im Lernpaket enthaltene ATtiny85 bietet bereits so viele Möglichkeiten, dass es praktisch unmöglich ist, sie alle zu nutzen.

Das Lernpaket ist eine Weiterentwicklung des Vorgänger-Lernpakets, das den ATtiny13 verwendete. Statt einer seriellen Schnittstelle gibt es nun einen USB-Anschluss. Und statt des Assembler-Schwerpunkts wird nun vorwiegend mit Bascom gearbeitet. Alles ist etwas einfacher und komfortabler geworden. Falls Sie bereits mit der Vorgänger-Version gearbeitet haben, werden Sie das Lernpaket mit dem Tiny85 als Upgrade empfinden. Achtfach mehr Speicher, neue Möglichkeiten, schnelleres Entwickeln, aber weiterhin ein kompakter kleiner Controller mit nur acht Anschlüssen. Durch die Möglichkeit der ISP-Programmierung können Sie jederzeit einen fabrikneuen Tiny85 in die Fassung setzen.

Entwickeln Sie Ihre eigenen Anwendungen und damit praktisch Ihr eigenes Spezial-IC. Sei es eine spezielle Alarmanlage, ein Messgerät oder eine Robotersteuerung, mit den vorgestellten Grundlagen können Sie Ihre Ideen umsetzen. Die im Lernpaket enthaltene Hardware ist

zugleich Entwicklungsplattform und Programmiergerät. Sie können beliebig viele Mikrocontroller programmieren und dann in Ihre Schaltungen einbauen  
Bleiben Sie neugierig!

Ihr Burkhard Kainka

Software und weitere Informationen zum Buch:  
[www.elektronik-labor.de/AVR/Tiny85.html](http://www.elektronik-labor.de/AVR/Tiny85.html)

Platine zum Bauteile: [www.ak-modul-bus.de/](http://www.ak-modul-bus.de/)

## INHALT

1	Einleitung .....	1
1.1	Bauteile .....	1
1.2	Die Controller-Platine .....	2
1.3	Verwendung einer Steckplatine .....	5
1.4	Entwicklungs-Software .....	6
2	Interface-Experimente .....	13
2.1	Portausgänge .....	13
2.2	Porteingänge .....	15
2.3	Spannungsmessung .....	17
2.4	Pullup-Widerstände .....	20
2.5	Der Fototransistor .....	21
2.6	LED als Lichtsensor .....	22
2.7	Ladungsmessung .....	23
2.8	Messungen an einem Elko .....	25
2.9	Das Oszilloskop .....	27
2.10	Der PWM-Ausgang .....	29
2.11	Schaltsschwellen .....	33
2.12	Programm-Upload .....	34
3	Bascom-Grundlagen und Portzugriffe .....	37
3.1	BASCOM-AVR .....	37
3.2	Der Bootloader .....	43
3.3	Ein Wechselblinker .....	47
3.4	Geschwindigkeitstest .....	49
3.5	Digitale Eingänge .....	50
3.6	Eingang mit Pullup .....	51
3.7	Die UND-Funktion .....	52
3.8	Das RS-Flipflop .....	54
3.9	Das D-Flipflop .....	55
3.10	Das Toggle-Flipflop .....	57
4	Die serielle Schnittstelle .....	59
4.1	Print-Ausgaben .....	59
4.3	Daten empfangen .....	64
4.5	Texteingabe .....	65
4.5	Byte-Empfang .....	66
5	Timer/Counter und Interrupts .....	68
5.1	Zeitmessung .....	68
5.2	Impulse zählen .....	70
5.3	Timer-Interrupt .....	71

5.4 Sekunden-Timer .....	73
5.5 PWM-Ausgang .....	75
5.6 Der weiche Blinker.....	76
5.7 Frequenzmessung .....	77
5.8 Interrupt-Eingang 0 .....	81
5.9 Pin-Change-Interrupt.....	82
5.10 Watchdog und Power-Down .....	84
6 Der AD-Wandler .....	86
6.1 10-Bit-Messung.....	86
6.2 Messung an vier Kanälen .....	89
6.3 Interne Referenz .....	91
6.4 Differenzmessung.....	93
6.5 Temperaturmessung .....	95
6.6 Zweipunktregler .....	97
7 Interfaces und Datenlogger .....	101
7.1 Das universelle Interface .....	101
7.2 Das Oszilloskop.....	104
7.3 Ein Transientenrecorder .....	107
7.4 Langzeit-Datenlogger .....	110
7.5 Kennlinienschreiber.....	112
7.6 Der MCS-Bootloader .....	115
8 Messtechnik-Anwendungen .....	119
8.1 DC-Millivoltmeter 0.1 mV ... 1100.0 mV .....	119
8.2 RMS-Millivoltmeter 0.1 mV bis 250.0 mV .....	121
8.3 Widerstandsmessung 100 $\Omega$ bis 1 M $\Omega$ .....	124
8.4 Kapazitätsmessung 1 nF bis 1000 $\mu$ F.....	126
8.7 Kapazitätsmessung 1 pF ... 1000 pF .....	128
8.6 Sinusgenerator 0...5 kHz .....	131
9 Arduino-Anwendungen.....	135
9.1 Der ATtiny in der Arduino-IDE .....	135
9.2 Software Serial .....	140
9.3 Bascom auf dem Arduino UNO .....	141
9.4 SIOS-Emulation .....	146
9.5 Analoge Plotter.....	150
9.6 SIOS für den Tiny85 .....	152
9.7 Programmieren mit CompactDefinition .....	156

# 1 Einleitung

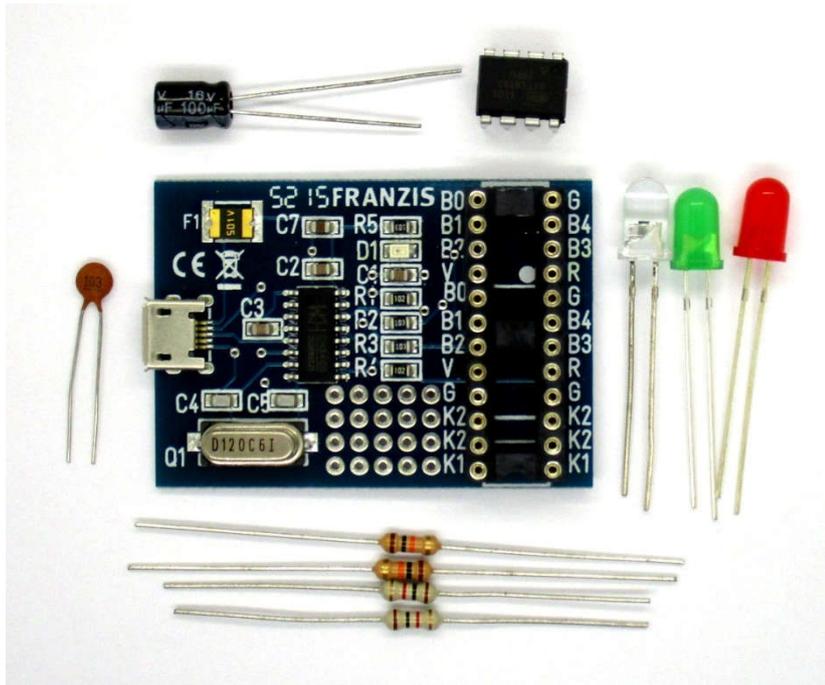
Das Buch geht von der bestückten Platine des Lernpaktes Mikrocontroller mit einer USB-Schnittstelle auf der Basis des CH340 aus. Die fertig aufgebaute Platine ist bei AK Modul-Bus erhältlich. Alternativ können Sie auch die unbestückte Platine bestellen oder selbst herstellen. Die Gerber-Dateien der Platine finden Sie unter [www.elektronik-labor.de](http://www.elektronik-labor.de). Eine weitere Alternative unter Verwendung eines USB-Seriell-Wandlers und einer Steckplatine wird weiter unten vorgestellt.

Vor den eigentlichen Experimenten müssen Sie nur den Mikrocontroller in die Fassung stecken, den USB-Treiber installieren und die Software installieren. Diese Vorbereitungen sind jedoch schnell erledigt.

## 1.1 Bauteile

Das Buch verwendet eine fertig bestückte Platine mit USB-Schnittstelle und die folgenden Bauteile:

- 1 Bestücke Platine mit USB-Interface
- 1 USB-Kabel
- 1 Mikrocontroller ATtiny85
- 1 Elektrolytkondensator 100  $\mu$ F
- 1 Kondensator 10 nF
- 1 Fototransistor (im klaren LED-Gehäuse)
- 1 grüne LED
- 1 rote LED
- 2 Widerstände 10 k $\Omega$  (Braun, Schwarz, Orange)
- 2 Widerstände 1 k $\Omega$  (Braun, Schwarz, Rot)

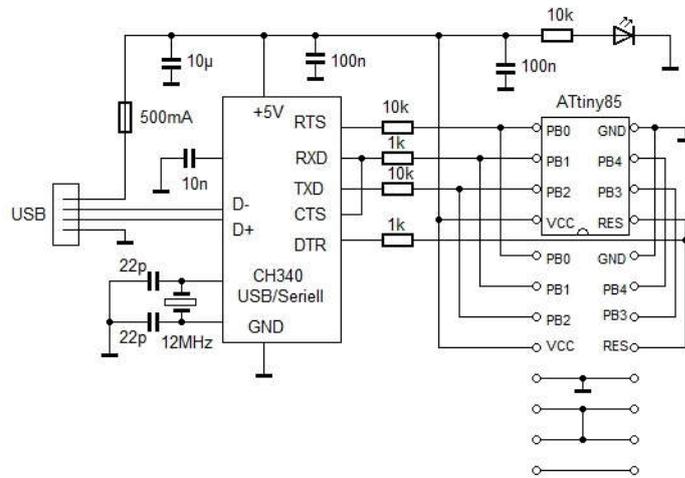


Setzen Sie den Mikrocontroller in den IC-Sockel, wie es das Foto zeigt. Richten Sie dazu die Anschlussbeinchen parallel aus, damit sie genau in den Sockel passen. Bei einem neuen IC sind die Beinchen meist etwas nach außen gespreizt. Sie lassen sich leicht zurechtbiegen, indem man das IC seitlich auf eine Fläche drückt. Beachten Sie beim Einsetzen in den Sockel die korrekte Polung. Pin 1 ist der Reset-Pin auf der Außenseite der Platine, Pin 4 ist der GND-Pin ganz an der Platinen-Ecke.

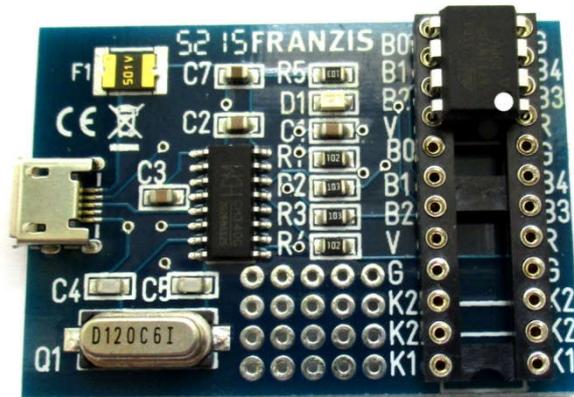
### **1.2 Die Controller-Platine**

Der Mikrocontroller ATtiny85 ist ein IC im achtbeinigen DIP-Gehäuse. Alle Anschlüsse sind an zusätzliche Sockelkontakte geführt und können in den Versuchen mit anderen Bauteilen verbunden werden. Acht weitere Anschlüsse des Sockels stellen ein kleines Experimentierfeld dar. Hier können die zusätzlichen losen Bauteile eingesteckt werden. Der

insgesamt 24-polige IC-Sockel dient also sowohl zu Aufnahme des Controllers als auch als Stecksystem.



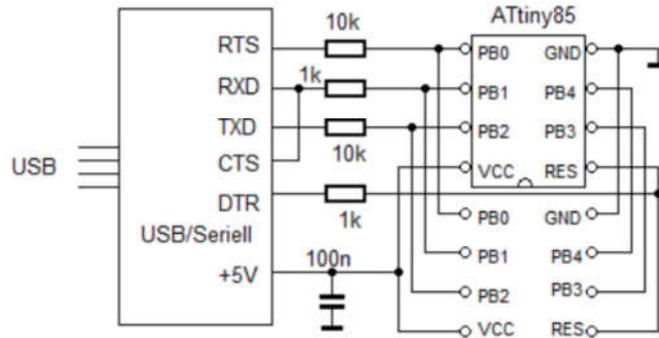
Das Experimentiersystem benötigt keine weitere Stromversorgung, da es über den USB-Anschluss mit 5 V versorgt wird. Zur Absicherung gegen Kurzschlüsse ist eine Polyswitch-Sicherung eingebaut. Dennoch sollten Kurzschlüsse und Verbindungen zu externen Stromquellen sorgfältig vermieden werden.



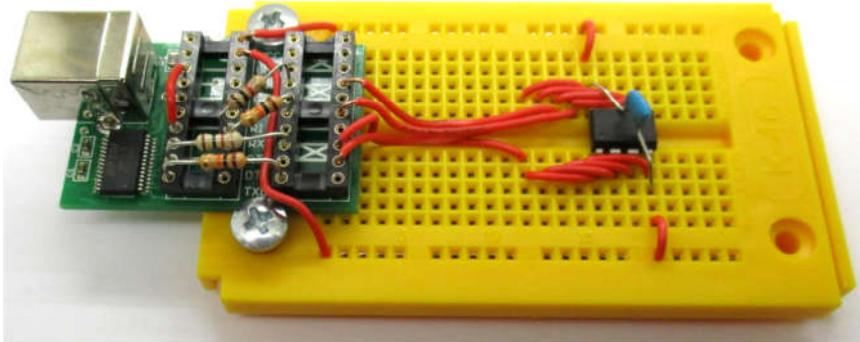


### 1.3 Verwendung eine Steckplatine

Die Schaltung der Experimentierplatine lässt sich in zwei Teile untergliedern. Links sieht man den USB-Seriell-Wandler Ch340, rechts den Mikrocontroller. Wenn man den linken Teil durch einen USB-Seriell-Wandler ersetzt, fehlen nur noch ein paar Widerstände und ein Kondensator. Diese Teile können auf einer Steckplatine untergebracht werden.

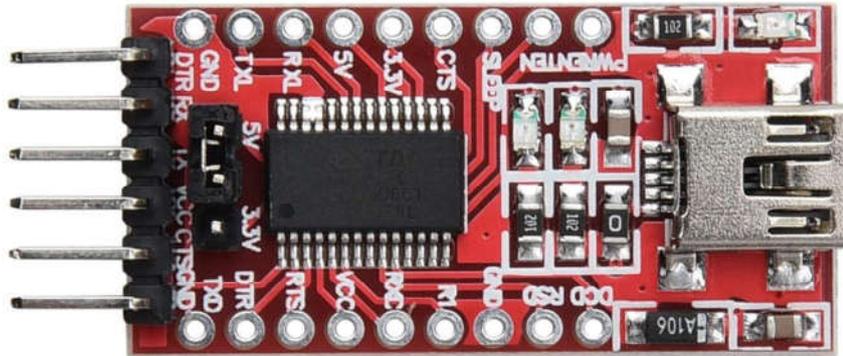


Statt eines CH340 kann auch ein FT232 verwendet werden. Wichtig ist nur, dass alle Handshakeleitungen zugänglich sind.



Dieser Aufbau verwendet die FT232-Platine aus dem Franzis Lernpaket MSR mit dem PC. Sie wurde mit zwei 4-mm-Blechschauben auf der Steckplatine befestigt und trägt schon die vier nötigen Widerstände. Der

Tiny85 auf der Steckplatine bekommt wie auf der Originalplatine eine Verdoppelung aller acht Anschlüsse, sodass die sich die gezeigten Experimente direkt umsetzen lassen. Vier Leitungen führen zum Programmierinterface. Zusätzlich wird die Betriebsspannung von der USB-Platine übernommen.



Als Alternative kommen Breakout-Boards mit dem FT232RL in Frage, die oft im Zusammenhang mit Arduino-Projekten angeboten werden. Auch damit lassen sich kompakte Entwicklungssysteme realisieren.

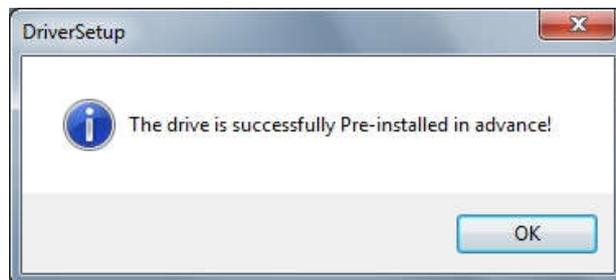
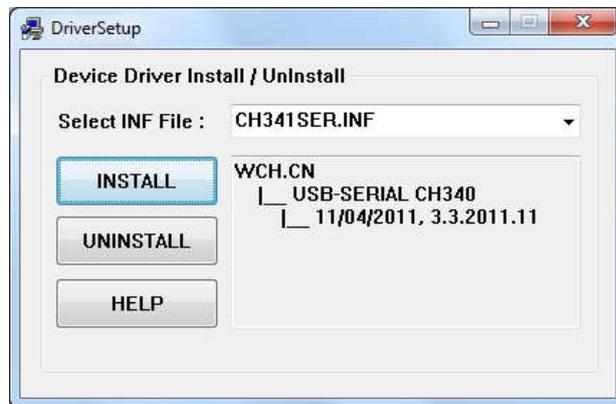
Wer sich stärker an die Platine des Lernpaktes anlehnen möchte, kann auch eine 12-polige IC-Fassung mit gedrehten Kontakten auf eine Rasterplatine löten und zusammen mit den übrigen Bauteilen verdrahten.

#### **1.4 Entwicklungs-Software**

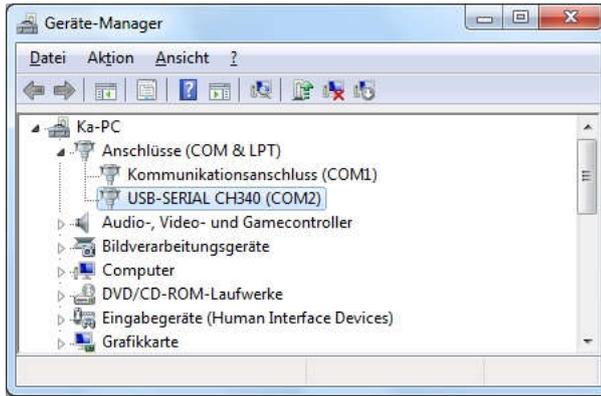
Zur Installation der Software muss das Anwenderprogramm zusammen mit weiteren Daten auf die Festplatte kopiert werden. Entpacken Sie dazu das Archiv LPmicro85.zip von der CD in ein Verzeichnis Ihrer Festplatte wie z.B. C:\LPmikro85\. Mit enthalten ist das Datenblatt des Mikrocontrollers, das man immer wieder aufschlagen sollte, wenn neue Hardware-Elemente vorgestellt und erprobt werden.

Vor dem ersten Anschluss der Platine muss der USB-Treiber des USB/Seriell-Wandlers CH340 installiert werden. Starten Sie das Programm Install.exe im Verzeichnis CH341SER. Es kopiert die

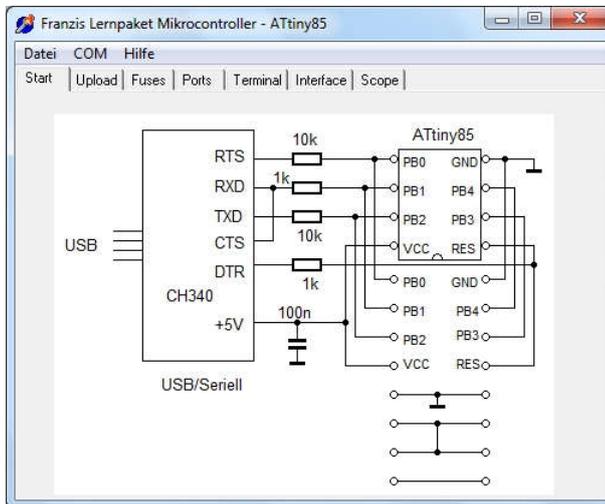
notwendigen Dateien auf Ihren PC. Der Erfolg wird abschließend angezeigt. Falls Sie mit den FT232 arbeiten, kennt Ihr den nötigen Treiber meist schon.



Schießen Sie dann die Platine über das beiliegende USB-Kabel an. Die Bereitschafts-LED auf der Platine leuchtet und zeigt an, dass Betriebsspannung anliegt. Der USB/Seriell-Wandler installiert sich als virtuelle serielle Schnittstelle, in den meisten Fällen als COM2. Sie können die Installation mit dem Windows-Gerätmanager überprüfen. Die vergebene COM-Nummer ist wichtig, weil sie in der Software angegeben werden muss.

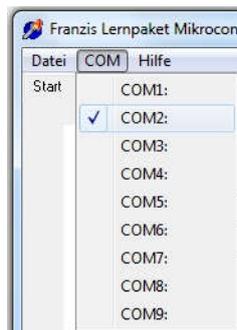


Starten Sie das Programm LPmikro85.exe. Der Startbildschirm zeigt das vereinfachte Schaltbild der Hardware und den Anschluss an die virtuelle serielle Schnittstelle. Bei der späteren Arbeit können Sie immer wieder einmal auf die Registerkarte „Start“ wechseln um dort nachzusehen, welcher Anschluss wo zu finden ist.

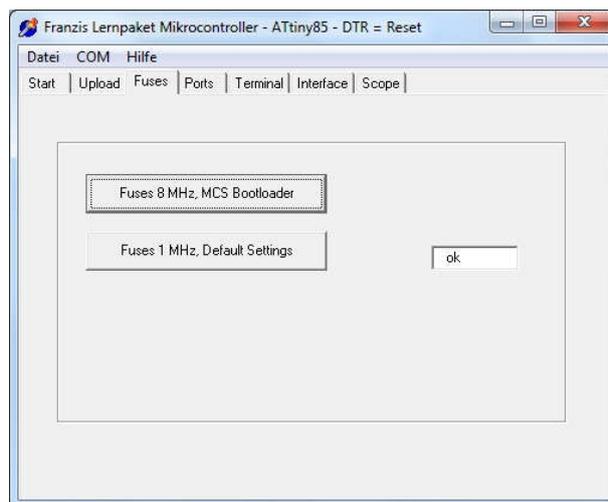


Nun muss noch die verwendete serielle Schnittstelle eingestellt werden. Klicken Sie auf COM und dann z.B. auf COM2 oder eine andere Schnittstelle entsprechend der vergebenen COM-Nummer. Wenn die Schnittstelle vorhanden ist und geöffnet werden kann, erscheint hier ein Häkchen. Beim Beenden des Programms wird die verwendete

Schnittstelle in der Datei LPmikro85.ini gespeichert. Bei jedem neuen Start wird diese Einstellung wieder übernommen.

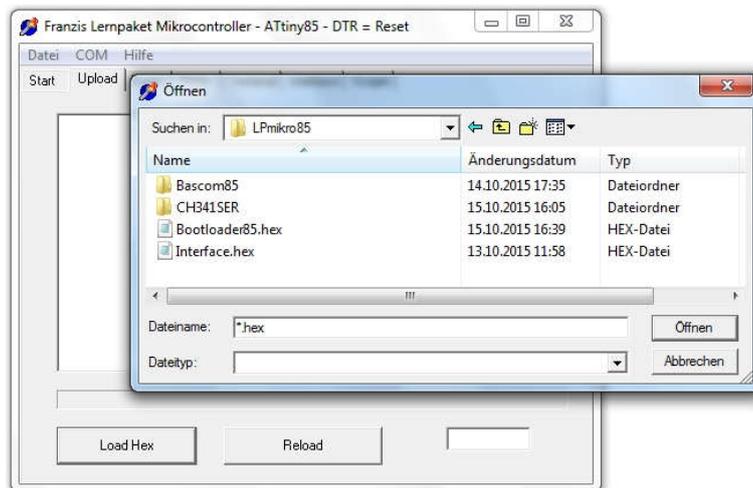
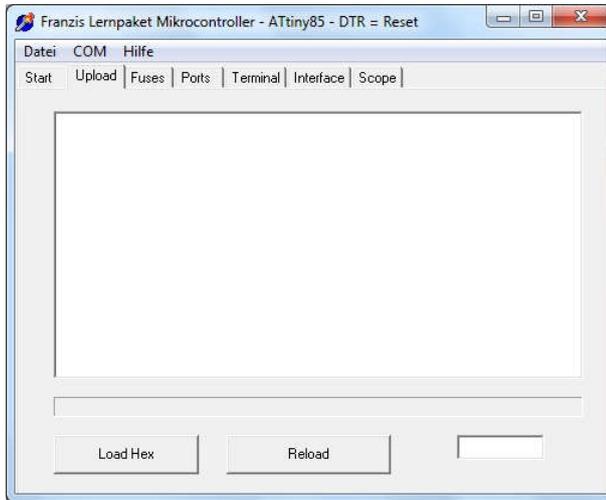


Wechseln Sie als nächstes auf die Registerkarte Fuses und klicken Sie auf die Schaltfläche 8 MHz, MCS Bootloader. Nach kurzer Zeit erhalten Sie eine ok-Meldung. Damit sind einige Grundeinstellungen des Mikrocontrollers programmiert. Er arbeitet nun mit einer Taktfrequenz von 8 MHz und ist für den späteren Einsatz mit dem Bootloader vorbereitet.



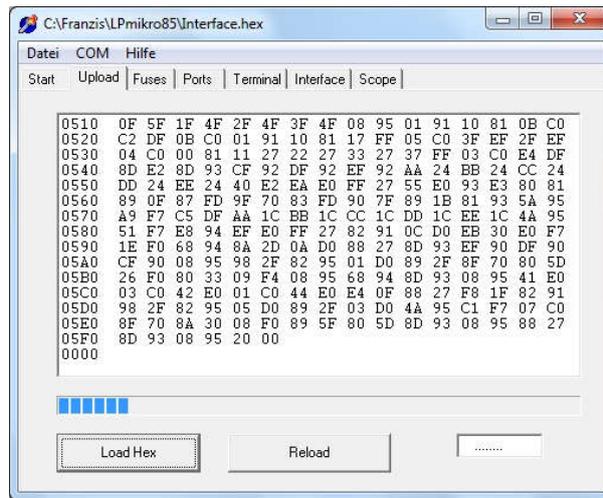
Öffnen Sie die Registerkarte Upload und klicken Sie dann auf Load Hex. Es öffnet sich ein Dateimenü für Programmdateien im Hexadezimal-

Format (Intel-Hex-Files). Damit können Sie beliebige Hex-Files in den Controller laden. Beginnen Sie mit dem Programm Interface.hex, das Sie im Grundverzeichnis LPmikro85 finden.

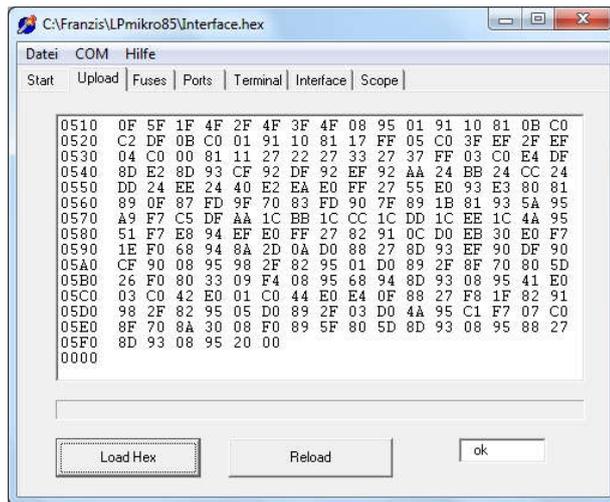


Mit der Auswahl des Hex-Files beginnt die Datenübertragung. Das Programm wird in einem Textfenster ausgegeben, sodass man bei Bedarf den Adressbereich und weitere Details nachschauen kann. Ein Balken

zeigt den Fortgang der Übertragung an. In der Kopfzeile steht der Pfad und der Dateiname des gerade geladenen Programms.



Die erfolgreiche Übertragung wird mit einer ok-Meldung angezeigt. Falls eine Fehlermeldung erscheint, liegt es in den meisten Fällen an einer falsch eingestellten COM-Schnittstelle. Mit der Reload-Taste kann dasselbe Programm noch einmal übertragen werden. Das ist allerdings nur sinnvoll, wenn es in der Zwischenzeit bearbeitet und neu kompiliert wurde und man die neue Dateiauswahl sparen will.



Das Programm Interface.hex ist relativ lang und benötigt mehr als eine Minute für die vollständige Übertragung. Die Upload-Funktion der Software ist also relativ langsam, wird aber auch nur selten benötigt.

Später wird meist der Bootloader verwendet, wobei die Übertragung direkt aus der Basom-Entwicklungsumgebung wesentlich schneller ist. Dazu muss einmal die Datei Bootloader85.hex in den Controller geladen werden. Das Programm sorgt dann für eine schnelle Selbst-Programmierung des Controllers. Jedes fertige Programm kann aber auch direkt mit der ISP-Programmierung über LPmikro85.exe in den Controller geladen werden, dann ohne den Bootloader. Damit hat man ein vielseitiges Programmiergerät für den ATtiny85, das übrigens auch für den ATtiny45 funktioniert. Der ATtiny25 kann dagegen nicht programmiert werden, weil er eine andere Blockgröße verwendet.

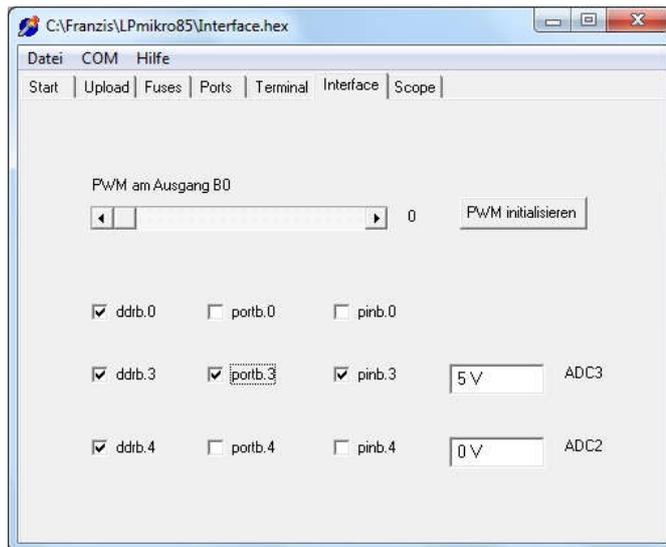
## 2 Interface-Experimente

Die ersten Schritte mit dem Mikrocontroller ATtiny85 sollen hier noch ohne eigene Programmierung vorgestellt werden. Das Programm Interface.hex bietet den Zugang zu allen Anschlüssen des Mikrocontrollers und kann zusammen mit der Interface-Funktion in LPmikro85.exe verwendet werden, um die grundlegenden Eigenschaften der Hardware kennenzulernen. Damit erhält man einen guten Überblick zu den Möglichkeiten des Controllers.

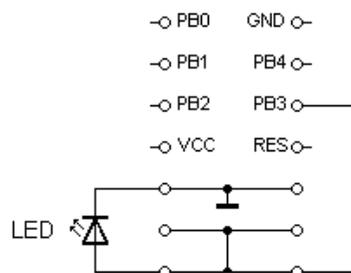
### 2.1 Portausgänge

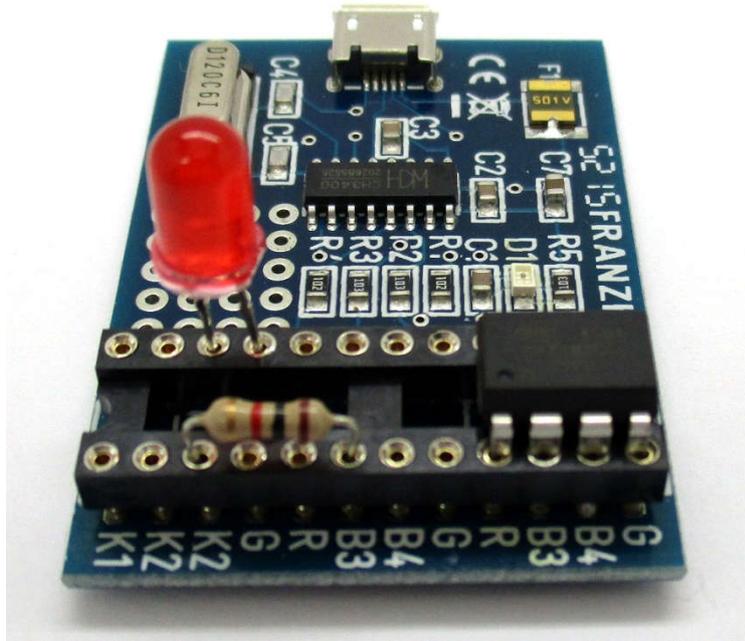
Der ATtiny85 hat acht Anschlüsse. Neben den Betriebsspannungsanschlüssen GND und VCC und dem Reset-Pin RES stehen fünf frei verwendbare Portanschlüsse zur Verfügung. Bei der Verwendung als Interface werden allerdings zwei Leitungen als Datenleitungen TXD und RXD zur seriellen Kommunikation mit dem PC gebraucht. Damit bleiben noch die drei Leitungen B0, B3 und B4 für sonstige Zwecke übrig. Jeder dieser Ports kann als Ausgang oder als Eingang verwendet werden. Außerdem haben sie jeweils noch Sonderfunktionen.

Öffnen Sie die Registerkarte Interface. Klicken Sie die Datenrichtungsbits ddrb.0, ddrb.3 und ddrb.4 aktiv. Damit sind alle drei Ports als Ausgänge initialisiert. Nun können Sie den jeweiligen Portzustand über die Kästchen portb.0 bis portb.4 umschalten. Einschalten liefert eine Spannung von 5 V am entsprechenden Pin, Ausschalten eine Spannung von 0 V. Verwenden Sie ein Voltmeter zur Überprüfung der Zustände. Gleichzeitig wird jeder Anschluss auch als Eingang gelesen und in PinB.0 bis Pinb.4 angezeigt. Der gelesene Zustand entspricht dem ausgegebenen Zustand, d.h. Sie können den realen logischen Zustand auch ohne eine Spannungsmessung erkennen.



Schließen Sie eine LED mit Vorwiderstand am Port PB3 an, die Sie dann per Software beliebig ein- und ausschalten können.

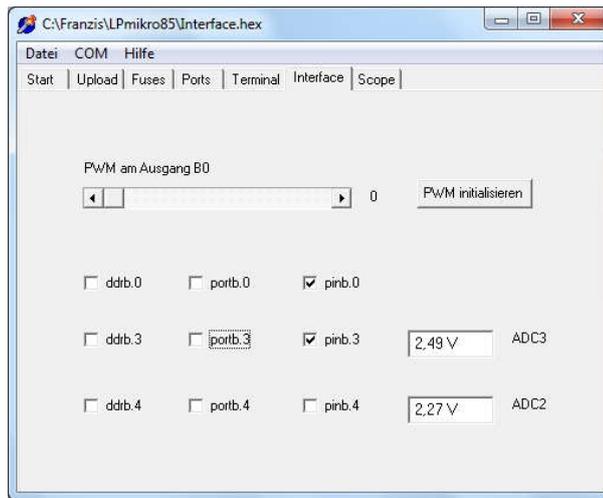




Gleichzeitig mit der Portfunktion verwendet das Interfaceprogramm die beiden Eingänge B3 und B4 als analoge Eingänge. Sie können also direkt die tatsächliche Spannung am Port ablesen. Bei Belastung mit einer LED ergibt sich aufgrund des Innenwiderstands am Port eine geringfügig kleinere Ausgangsspannung als 5 V. Der digitale Zustand wird aber immer noch als 1 gelesen.

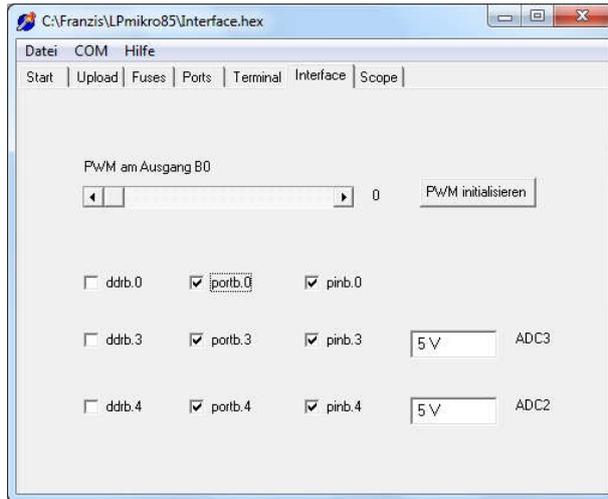
## 2.2 Porteingänge

Schalten Sie alle `ddrb`-Bits und alle `portb`-Bits aus. Die Ports sind damit hochohmige CMOS-Eingänge. Berühren Sie die Eingänge mit einem Draht oder einem Widerstand. Dabei laden sie sich zufällig auf. Sie können 0 oder 1 sein oder ständig wechseln. Tatsächlich liefern Sie beim Berühren eines Eingangs meist eine 50-Hz-Brummspannung, also einen dauernden Zustandswechsel. Je nach dem zufälligen Zeitpunkt des Loslassens bleibt ein 1- oder ein 0-Zustand stehen, der sich jedoch nach kurzer Zeit von allen wieder ändern kann.

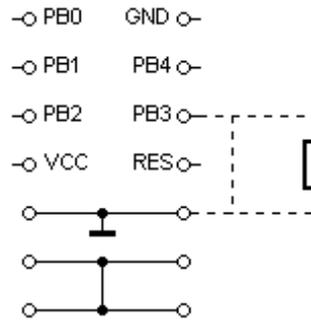


Allgemein werden offene Eingänge in der Digitaltechnik vermieden, eben weil sie keine definierten Zustände haben. Wenn ein Eingang z.B. verwendet werden soll um einen Schalter abzufragen, verwendet man zusätzliche Widerstände gegen Masse (Pull Down) oder gegen die Betriebsspannung (Pull Up). Sie können Pullup- oder Pulldown-Widerstände simulieren, indem Sie beim Berühren eines Eingangs gleichzeitig VCC oder GND berühren. Ihre Hand dient dann als Widerstand, der eine eindeutige Spannung an den Eingang legt.

Der ATtiny85 enthält aber auch interne Pullup-Widerstände, die sich bei Bedarf einschalten lassen. Dazu muss das jeweilige Port-Bit eingeschaltet werden, während das Datenrichtungsbit low ist. Schalten Sie die Bits portb.3 und portb.4 ein. Es werden dann bei offenem Eingang 1-Zustände zurückgelesen. Die entsprechenden Spannungen an den Eingängen betragen 5 V.



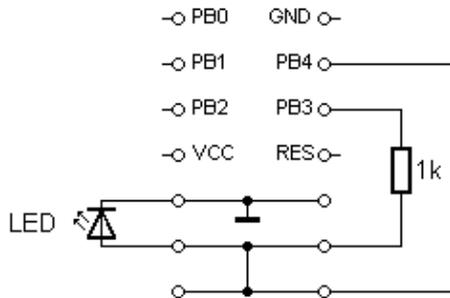
In diesem Zustand lassen sich externe Schalter gegen Masse abfragen. Ein geöffneter Schalter liefert 1, ein geschlossener Schalter 0. Verwenden Sie einen Draht nach GND zur Simulation eines geschlossenen Schalters. In diesem Fall können Sie auch den 1-k $\Omega$ -Widerstand als leitende Verbindung verwenden, weil die internen Pullups wesentlich hochohmiger sind.



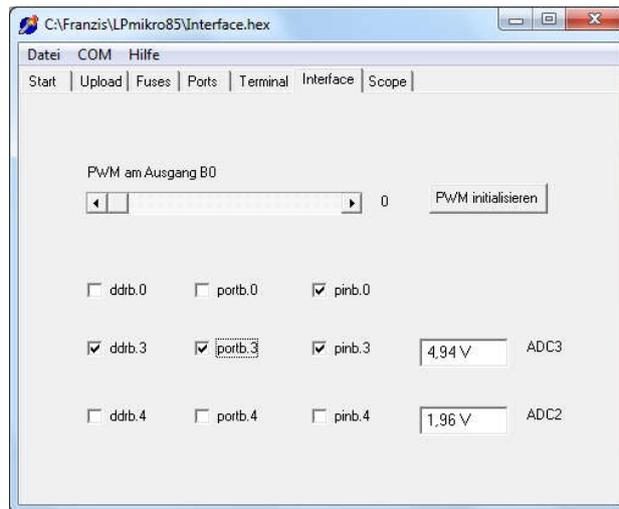
### 2.3 Spannungsmessung

Die analogen Eingänge des Mikrocontrollers lassen sich auch für allgemeine Spannungsmessungen verwenden. Untersuchen Sie z.B. die

Durchlassspannung der LED. Ein Port wird dazu als Ausgang geschaltet, der zweite als hochohmiger Messeingang. Die LED wird über den Anschluss PB3 und einen Vorwiderstand von 1 k $\Omega$  eingeschaltet. Der 10-k $\Omega$ -Widerstand dient hier zunächst nur als Drahtbrücke zur Messung der LED-Spannung über den Anschluss PB4.



Im eingeschalteten Zustand finden Sie eine LED-Spannung von ca. 1,9 V. Am Port liegt eine Spannung von 4,9 V. Damit beträgt der Spannungsabfall am Vorwiderstand  $4,9 \text{ V} - 1,9 \text{ V} = 3 \text{ V}$ . Der LED-Strom ist also  $3 \text{ V} / 1 \text{ k}\Omega = 3 \text{ mA}$ . Gleichzeitig lässt sich der Innenwiderstand des Ausgangsports bestimmen. Bei einem Spannungsabfall von 0,06 V und einem Strom von 3 mA ergibt sich ein On-Widerstand von ca.  $20 \Omega$  ( $60 \text{ mV} / 3 \text{ mA} = 20 \Omega$ ).



Verwenden Sie nun einen Vorwiderstand von 10 k $\Omega$ , indem Sie PB4 als Ausgang und PB3 zur Spannungsmessung verwenden. Die LED leuchtet schwächer, weil der LED-Strom nun nur noch etwa 0,3 mA ist. Die Spannung an der LED verringert sich aber nur geringfügig auf 1,8 V. Dies ist auf die steile Diodenkennlinie (vgl. Kap. 7.5) der LED zurückzuführen.

