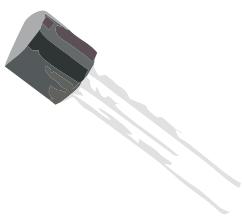


TransistorTester
mit AVR-Mikrocontroller
Ein Gerät zur Bestimmung und Messung von
elektronischen Bauteilen
und ein wenig mehr . . .

1 —||— 3
329pF

Version 1.13k



Karl-Heinz Kübbeler
kh_kuebbeler@web.de

BJT-PNP
hFE=226
Ie=1.6mA
Ube=673mV+

BJT-PNP
hFE=226
Ie=1.6mA
PIN 1=C 2=B 3=E



Eine modifizierte Ausgabe
von bm-magic und nachbearbeitet von K.-H. Kübbeler
©

1 —□— 3
0.1Ω L=.05mH

15. November 2021



1 —△— 3
Uf=670mV
25pF-17pF @0-5V
Ir=.80μA



1 —□— 2
2205 Ω

1 —□— 3
10.08 kΩ



13.05.2019/MOR

N-E-MOS
Vt=3.1V
Cg=2.86nF
RDS=.2Ω +

N-E-MOS
Vt=3.1V
Cg=2.86nF
3 —► 2 Uf=598mV+

N-E-MOS
Vt=3.1V
Cg=2.86nF
PIN 1=G 2=D 3=S

Inhaltsverzeichnis

1 Eigenschaften	6
2 Hardware	10
2.1 Die Schaltung des TransistorTesters	10
2.2 Erweiterungen für den TransistorTester	12
2.2.1 Schutz der ATmega-Eingänge	12
2.2.2 Zenerspannungsmessung	13
2.2.3 Frequenzgenerator	13
2.2.4 Frequenzzähler	14
2.2.5 Impulsdrehgeber	14
2.2.6 Anschluss eines graphischen Displays	16
2.2.7 Anschluss eines graphischen Farbdisplays	21
2.3 Hinweise für den Aufbau des TransistorTesters	22
2.4 Umrüstung von Tester Versionen nach Markus F.	23
2.5 Erweiterte Schaltung mit ATmega644 oder ATmega1284	24
2.6 Aufbau mit ATmega1280 oder Arduino Mega	26
2.7 Chinesische Nachbauten mit Textdisplay	28
2.8 Chinesische Nachbauten mit graphischem Display	29
2.9 Chinesische Bausätze mit Grafikdisplay	34
2.10 Und noch ein Clone von Hiland M644	35
2.11 Fehlersuche	37
3 Bedienungshinweise	40
3.1 Der Messbetrieb	40
3.2 Optionale Menüfunktionen für den ATmega328	41
3.3 Selbsttest und Kalibration	44
3.4 Besondere Benutzungshinweise	45
3.5 Problemfälle	46
3.6 Messung von PNP- und NPN-Transistoren	46
3.7 Messung von JFET- und D-MOS-Transistoren	47
3.8 Messung von E-MOS Transistoren und IGBTs	47
3.9 Messen von Kondensatoren	48
3.10 Messen von Spulen	48
4 Programmieren des TransistorTesters	49
4.1 Konfigurieren des TransistorTesters	49
4.2 Programmierung des Mikrocontrollers	60
4.2.1 Benutzung von Linux	62
4.2.2 Erstellen der Software unter Windows	70
4.2.3 Benutzung des WinAVR-Paketes unter Windows	70

5 Beschreibung des Messverfahrens	72
5.1 Messung von Halbleitern	74
5.1.1 Messung eines PNP-Transistors oder eines P-Kanal MOSFETs	75
5.1.2 Messung eines NPN-Transistors oder eines N-Kanal-MOSFET	76
5.1.3 Vereinfachter Ablauf der Transistorerkennung	80
5.1.4 Messung von Dioden	82
5.1.5 Ergebnisse der verschiedenen Messungen	83
5.2 Widerstands-Messung	88
5.2.1 Widerstandsmessung mit den 680-Ohm-Widerständen	88
5.2.2 Widerstandsmessung mit den 470-kOhm-Widerständen	90
5.2.3 Ergebnisse der Widerstands-Messung	91
5.3 Messen von Kondensatoren	95
5.3.1 Entladen der Kondensatoren	95
5.3.2 Messung von großen Kapazitäten	95
5.3.3 Messen von kleinen Kapazitäten	98
5.3.4 Messen sehr kleiner Kapazitäten mit der Samplingmethode	100
5.3.5 Messen des äquivalenten Serienwiderstandes ESR	101
5.3.6 Messen des äquivalenten Serienwiderstandes ESR, Methode 1	104
5.3.7 Messen des äquivalenten Serienwiderstandes ESR, Methode 2	108
5.3.8 Spannungsverlust nach einem Ladepuls, Vloss	114
5.3.9 Separate Kapazitäts- und ESR-Messung	115
5.3.10 Ergebnisse der Kondensator-Messung	116
5.3.11 Automatischer Abgleich der Kondensator-Messung	120
5.4 Messen von Induktivitäten	124
5.4.1 Ergebnisse der Induktivitäts-Messungen	125
5.4.2 Messung kleiner Induktivitäten mit der Sampling Methode	125
5.5 Selbsttest-Funktion	129
5.5.1 Einige Selbsttest-Ergebnisse	133
5.6 Frequenzmessung	137
5.6.1 Kalibration der Frequenz mit GPS- oder GLONASS-Empfänger	138
5.6.2 Kalibration der Quarzfrequenz mit einem Uhrenmodul	139
6 Signal-Erzeugung	141
6.1 Frequenz-Erzeugung	141
6.2 Pulsweiten-Erzeugung	141
7 Bekannte Fehler und ungelöste Probleme	143
8 Spezielle Softwareteile	145
9 Arbeitsliste und neue Ideen	146

Bemerkung zu dieser Ausgabe: Beim Übersetzen ins andere Sprache, wurden auch die Texte in den

Bildern und Diagrammen, die im Original Englisch sind, auch übersetzt.

- Es wurde Unterkapitel 2.10 (Clone von der Firma Hiland M644) hinzugefügt.

- In Unterkapitel 4.1 wurde das Thema Programmer ergänzt.

- Und zum Schluss wurde Unterkapitel 4.2.1 (Programmierung unter Linux) ergänzt, damit auch Linux „Neulinge“ einen Erfolg erfahren.

Über diese Maßnahmen wurde Autor informiert.

Leider wurde, meines wissen nach, bis jetzt das Dokument nicht aktualisiert.

- Ich selbst bekam keine positive Antwort.

- Da ich der Meinung bin, das die Ergänzungen für „Linux Anfänger“ wichtig sind, hat diese Ausgabe seine Berechtigung.

Das Original ist natürlich unten [3] zu erreichen.

20.02.2020

bm-magic

Viele Änderungen wurden übernommen, aber die Entstehung der Graphiken mit `gnuplot` aus Roh-Messdaten wurde beibehalten mit entsprechender Makefile, die solche Abhängigkeiten dokumentiert. Von Zeit zu Zeit werden einige Meßreihen beim Abschluß einer Softwareversion wiederholt und die Daten aktualisiert. Dadurch ändern sich die Graphiken automatisch beim Neukompilieren.

Dieses Verfahren sollte beibehalten werden und für alle Sprachen funktionieren.

Zur leichteren Anpassung an Papierformate habe ich die Größenangaben relativ gemacht statt fester Größenangaben wie cm und pt.

6.3.2021

K.-H. Kübbeler

Wie man oben sehen kann, wurde das Dokument nun durch Autor aktualisiert und z.B. Unterkapitel 4.2.1 mit vielen... mir bis her unbekannten, Tips ergänzt... dank an den Autor.

Diese Ausgabe wurde notwendig, weil die Adresse des des `~/transistortester` Archivs

von `svn://www.mikrocontroller.net/transistortester`

zum `https://github.com/Mikrocontroller-net/transistortester` [1]

umgezogen wurde.

24.03.2021

bm-magic

Vorwort

Grundsätzliches

Jeder Bastler kennt das folgende Problem: Man baut einen Transistor aus oder man nimmt einen aus einer Bastelkiste. Wenn man die Typenbezeichnung erkennen kann und man bereits ein Datenblatt hat oder eins bekommen kann, ist alles in Ordnung. Aber wenn man keine Datenblätter findet, hat man keine Idee, was das für ein Bauteil sein kann. Mit den konventionellen Messmethoden ist es schwierig und zeitaufwändig den Typ des Bauteils und dessen Parameter herauszufinden. Es könnte ein NPN, PNP, N- oder P-Kanal-MOSFET usw. sein. Es war die Idee von Markus F., diese Arbeit von einem AVR-Mikrocontroller erledigen zu lassen.

Wie meine Arbeit begann

Meine Beschäftigung mit der Software des TransistorTesters von Markus F. [2] hat begonnen, als ich Probleme mit meinem Programmer hatte. Ich hatte eine Platine und Komponenten gekauft, aber ich war mit dem Windows-Treiber nicht in der Lage den EEPROM des ATmega8 ohne Fehlermeldungen zu beschreiben. Deshalb habe ich die Software von Markus F. genommen und habe alle Zugriffe auf den EEPROM-Speicher durch Zugriffe auf den Programm-Speicher (Flash) ersetzt. Bei der Durchsicht der Software, um an anderer Stelle Programmspeicher einzusparen, hatte ich die Idee, das Ergebnis der ReadADC-Funktion von ADC-Einheiten in eine Millivolt (mV) Auflösung zu ändern. Die mV-Auflösung wird für die Ausgabe von Spannungswerten gebraucht. Wenn die ReadADC-Funktion direkt die mV-Auflösung liefert, kann man die Umwandlung für jeden Ausgabewert einsparen. Diese mV-Auflösung kann man erhalten, wenn man zuerst die Ergebnisse von 22 ADC-Einlesungen addiert. Die Summe muss mit Zwei multipliziert und durch Neun geteilt werden. Das ergibt einen Maximalwert von $\frac{1023 \cdot 22 \cdot 2}{9} = 5001$, welcher hervorragend zu der gewünschten mV-Auflösung passt. So hatte ich zusätzlich die Hoffnung, dass die Erhöhung der ADC-Auflösung durch Überabtastung helfen könnte, die Spannungs-Einlesung zu verbessern, wie es in dem Atmel-Report AVR121 [7] beschrieben ist. Die Original-Version von ReadADC hat die Ergebnisse von 20 ADC-Einlesungen addiert und danach durch 20 dividiert, so dass das Ergebnis wieder die Original-Auflösung des ADC hat. Deshalb konnte niemals eine Erhöhung der ADC-Auflösung durch Überabtastung stattfinden. So hatte ich wenig Arbeit, die ReadADC-Funktion zu ändern, aber dies erforderte die Analyse des kompletten Programms und Änderung aller „if“-Abfragen im Programm, wo Spannungswerte geprüft wurden. Aber dies war nur der Beginn meiner Arbeit!

Mehr und mehr Ideen wurden eingebaut, um die Messung schneller und genauer zu machen. Zusätzlich wurde der Bereich der Widerstands- und Kondensator-Messung erweitert. Das Ausgabe-Format für das LC-Display wurde verändert, so wurden Symbole für die Darstellung von Dioden, Widerständen und Kondensatoren verwendet. Für weitere Einzelheiten schauen Sie in das aktuelle Eigenschaften-Kapitel 1. Geplante Arbeiten und neue Ideen wurden im Arbeitslisten-Kapitel 9 gesammelt. Inzwischen kann ich unter dem Linux-Betriebssystem auch den EEPROM des ATmega8 einwandfrei beschreiben. Danken möchte ich hier dem Erfinder und Softwareautor Markus Frejek, der mit seinem Projekt die Weiterführung erst möglich gemacht hat. Außerdem möchte ich den Autoren der zahlreichen Beiträge im Diskussionsforum danken, die dabei geholfen haben, neue Aufgaben, Schwachstellen und Fehler zu finden. Weiter gilt mein Dank auch Markus Reschke, der mir die Erlaubnis gegeben hat, seine aufgeräumte Versionen im SVN-Server zu veröffentlichen. Außerdem sind einige Ideen und Softwareteile von Markus R. in meine Version eingeflossen, auch hierfür herzlichen Dank. Auch hat Wolfgang Sch. eine große Leistung vollbracht, das grafische Display mit ST7565-Controller zu unterstützen. Vielen Dank für den Patch für die 1.10k Softwareversion und für die Integration in die aktuelle Entwicklerversion. Bedanken möchte ich mich auch bei Asco B., der eine Platine für Nachbauwillige entwickelt hat und bei Dirk W., der sich um die Sammelbestellungen für diese Platine gekümmert hat. Ich hätte unmöglich selbst die Zeit dafür aufbringen können, sonst wäre die Weiterentwicklung

der Software noch nicht so weit gekommen. Auch bei den Mitgliedern des Ortsverbandes Lennestadt des Deutschen Amateur Radio Clubs (DARC) möchte ich mich für zahlreiche Anregungen und Verbesserungsvorschläge bedanken. Für die Integration der Sampling-Methode, mit der die Messung kleiner Kapazitätswerte und kleiner Induktivitäten erheblich verbessert wurde, möchte ich mich bei dem Amateurfunker Pieter-Tjerk (PA3FWM) bedanken. Nicht zuletzt möchte ich mich bei Nick L. aus der Ukraine bedanken, der mich mit Prototyp-Leiterplatten unterstützt hat, einige Hardware-Erweiterungen vorgeschlagen hat und für die russische Übersetzung dieser Beschreibung gesorgt hat.

Kapitel 1

Eigenschaften

1. Arbeitet mit Mikrocontrollern vom Typ ATmega8, ATmega168 oder ATmega328. Auch ATmega644 und ATmega1284 oder ATmega1280 und ATmega2560 können verwendet werden.
2. Anzeige der Messergebnisse auf einem 2x16 oder 4x20 Zeichen großen LC-Display. Alternativ kann bei Verwendung eines Prozessors mit mindestens 32K Flash Speicher auch ein graphisches Display mit 128x64 Pixel und ST7565-, ST7920, NT7108, KS0108 oder SSD1306-Controller benutzt werden. Dabei wird anstelle der Standard 4-Bit Parallel-Schnittstelle auch entweder eine 4-Wire SPI-Schnittstelle oder ein I²C-Bus benutzt. Sogar Farbdisplays mit ILI9163 oder ST7735 Controller sind mit der SPI-Schnittstelle anschliessbar. Für den NT7108 oder KS0108 Controller ist ein seriell-parallel Wandler 74HC(T)164 oder 74HC(T)595 erforderlich, da diese Controller nur den 8-Bit parallel Anschluß erlauben. Displays mit PCF8812 oder PCF8814 Controller können ohne die großen Transistor-Symbole benutzt werden, da die Displaygröße unzureichend ist (102x65 und 96x65).
3. Ein-Tasten-Bedienung mit automatischer Abschaltfunktion.
4. Batterie-Betrieb ist möglich, weil der Strom im abgeschalteten Zustand nur etwa 20nA beträgt. Ab Softwareversion 1.05k wird nach Möglichkeit in den Messpausen der Schlafzustand des ATmega zum Stromsparen benutzt, wenn kein Impulsdrehgeber benutzt wird.
5. Preisgünstige Lösung ist möglich ohne Quarz und ohne automatische Abschaltung.
6. Automatische Erkennung von NPN und PNP bipolaren Transistoren, N- und P-Channel MOSFETs, JFETs, Dioden, Doppeldioden, N- und P-IGBTs, Thyristoren und Triacs. Für Thyristoren und Triacs müssen die Zünd- und Halteströme für die richtige Erkennung erreicht werden können. Bei IGBTs muß die Gate Schwellwertspannung unter 5V liegen.
7. Darstellung der Pin-Belegung der erkannten Bauteile.
8. Messung des Stromverstärkungsfaktors und der Basis-Emitter-Schwellspannung für bipolare Transistoren.
9. Darlington-Transistoren können durch die höhere Schwellspannung und durch den hohen Stromverstärkungsfaktor erkannt werden.
10. Automatische Erkennung einer Schutzdiode bei bipolaren Transistoren und bei MOSFETs.
11. Messung der Schwellwert-Spannung, der Gate-Kapazität und des R_{Dson} bei einer Gatespannung von knapp 5V von MOSFETs.

12. Bis zu zwei Widerstände werden gemessen und mit den  Symbolen und den Widerstands-Werten mit bis zu vier Dezimalstellen in der richtigen Dimension angezeigt. Alle Symbole werden eingerahmt mit den gefundenen Testpin Nummern des Testers (1-3). Deshalb können auch Potentiometer gemessen werden. Wenn der Schleifer eines Potentiometers auf eine Endposition gestellt ist, kann der Tester nicht mehr zwischen mittlerem Anschluss und Endanschluss unterscheiden.
13. Die Auflösung der Widerstandsmessung ist jetzt bis zu $0,01\Omega$, Werte von bis zu $50M\Omega$ werden erkannt.
14. Ein Kondensator kann erkannt und gemessen werden. Der wird mit dem Symbol  und dem Kapazitätswert mit bis zu vier Dezimalstellen in der richtigen Dimension angezeigt. Der Wert kann zwischen $25pF$ (bei $8MHz$ Takt, $50pF$ bei $1MHz$ Takt) bis $100mF$ liegen. Die Auflösung kann bis zu $1pF$ (bei $8MHz$ Takt) betragen.
15. Bei Kondensatoren mit einer Kapazität über $20nF$ wird zusätzlich der äquivalente Serienwiderstand (ESR) des Kondensators mit einer Auflösung von $0,01\Omega$ gemessen und mit zwei Dezimalstellen angezeigt. Diese Fähigkeit steht nur zur Verfügung, wenn der ATmega mindestens 16K Flashspeicher besitzt.
16. Für Kondensatoren mit einem Kapazitätswert über $5000pF$ kann der Spannungsverlust V_{loss} nach einem Ladepuls bestimmt werden. Der Spannungsverlust gibt einen Hinweis auf die Güte des Kondensators.
17. Bis zu zwei Dioden werden mit dem Symbol  oder dem Symbol  in der richtigen Reihenfolge angezeigt. Zusätzlich werden die Schwellspannungen angezeigt.
18. Eine LED wird als Diode erkannt, die Schwellspannung ist viel höher als bei einer normalen Diode. Doppeldioden werden als zwei Dioden erkannt.
19. Zener-Dioden können erkannt werden, wenn die Zener-Spannung unter $4,5V$ ist. Sie werden als zwei Dioden angezeigt, man kann das Bauelement nur mit den Spannungen erkennen. Die äußeren Testpin-Nummern, welche die Dioden Symbole umgeben, sind in diesem Fall identisch. Man kann die wirkliche Anode der Diode nur durch diejenige Diode herausfinden, deren Schwellwert-Spannung nahe bei $700mV$ liegt!
20. Wenn mehr als 3 Dioden erkannt werden, wird die gefundene Anzahl der Dioden zusammen mit der Fehlermeldung angezeigt. Das kann nur passieren, wenn Dioden an alle drei Test-Pins angeschlossen sind und wenigstens eine eine Zener-Diode ist. In diesem Fall sollte man nur zwei Test-Pins anschließen und die Messung erneut starten, eine Diode nach der anderen.
21. Der Kapazitätswert einer einzelnen Diode in Sperr-Richtung wird automatisch ermittelt. Bipolare Transistoren können auch untersucht werden, wenn nur die Basis und entweder Kollektor oder Emitter angeschlossen wird. Bei ATmega mit mehr als 8k Flash Speicher wird außerdem der Sperrstrom mit einer Auflösung von $2nA$ gemessen. Der Wert wird nur ausgegeben, wenn er nicht null ist.
22. Die Anschlüsse einer Gleichrichter-Brücke können mit nur einer Messung herausgefunden werden.
23. Kondensatoren mit Kapazitätswerten von unter $25pF$ werden normalerweise nicht erkannt, aber sie können zusammen mit einer parallel geschalteten Diode oder mit einem parallel geschaltetem Kondensator mit wenigstens $25pF$ gemessen werden. In diesem Fall muss der Kapazitätswert des parallel geschalteten Bauteils vom Messergebnis abgezogen werden. Bei Prozessoren mit mindestens 32K Flash Speicher wechselt der Tester mit einem Kondensator $> 25pF$ an TP1 und TP3 in eine Kondensator-Meßfunktion, die auch Kapazitäten ab $1pF$ direkt mißt.

24. Bei Widerständen unter 2100Ω wird auch eine Induktivitätsmessung durchgeführt, wenn der ATmega wenigstens 16K Flashspeicher besitzt. Dabei wird zusätzlich zum Widerstands-Symbol  ein Induktivitäts Symbol  angezeigt.
Der Anzeigebereich ist etwa $0,01mH$ bis über $20H$, die Genauigkeit ist allerdings nicht hoch. Das Ergebnis wird nur bei einem Einzelwiderstand zusammen mit dem Widerstandswert angezeigt.
25. Die Messzeit beträgt ungefähr zwei Sekunden, nur Kapazitätsmessungen und Induktivitätsmessungen können länger dauern.
26. Die Software kann für Messserien mit vorgebbarer Wiederhol-Zahl konfiguriert werden, bevor die automatische Abschaltung ausschaltet.
27. Eingebaute Selbsttest-Funktion inklusive einem optionalen $50Hz$ Frequenz-Generator um die Genauigkeit der Taktfrequenz und der Verzögerungszeiten zu überprüfen (nur mit mindestens 16K Flash Speicher).
28. Wählbare Möglichkeit den Nullabgleich für die Kondensatormessung und die Innenwiderstände für die Portausgänge beim Selbsttest automatisch zu bestimmen (nur mit mindestens 16K Flash Speicher). Ein externer Kondensator mit einer Kapazität zwischen $100nF$ und $20\mu F$ an Pin 1 und Pin 3 ist notwendig, um die Offset-Spannung des analogen Komparators zu kompensieren. Dies kann den Messfehler bei Kapazitätssmessungen bis zu $40\mu F$ reduzieren. Mit dem gleichen Kondensator wird eine Korrekturspannung zum Einstellen der richtigen Verstärkung für die ADC Messung mit der internen $1,1V$ Referenzspannung berechnet.
29. Anzeige des Kollektor-Emitter-Reststroms I_{CEO} mit stromloser Basis ($1\mu A$ Auflösung) und des Kollektor-Emitter Reststroms I_{CES} mit der Basis auf Emitter-Potential gehalten (nur bei mindestens 16K Flash). Diese Werte werden nur angezeigt, wenn sie nicht Null sind (besonders für Germanium-Transistoren).
30. Für ATmega mit mindestens 32K Flash-Speicher wechselt der Tester vom Multifunktionstest zu einem Modus als Widerstands-Meßgerät, wenn bei der automatischen Bauteile-Erkennung nur ein Widerstand an Test Pin 1 (TP1) und Test Pin 3 (TP3) erkannt wird. Wenn in der Makefile auch die Induktivitätsmessung beim Widerstandsmeßgerät mit der Option RMETER_WITH_L gewünscht wurde, werden bei der Widerstandsmessung auch Induktivitäten gemessen. Der Betriebsmodus wird durch [R] oder [RL] auf der rechten Seite von Zeile 1 des Displays angezeigt. Genau so wechselt der Tester zu einem Kapazitätsmeßgerät, wenn bei der Bauteile-Untersuchung eine Kondensator an TP1 und TP3 erkannt wurde. Dieser Betriebsmodus wird durch [C] auf der rechten Seite der Zeile 1 angezeigt. In dieser Betriebsart können Kondensatoren ab $1pF$ gemessen werden. Lediglich für den automatischen Start der Funktion braucht man einen Kondensator mit mehr als $25pF$. Beide Sonderfunktionen können durch einen Tastendruck wieder beendet werden. Der Tester fährt dann mit der normalen Meßfunktion fort.
31. Für Prozessoren mit mindestens 32K Flash kann eine Dialogfunktion gewählt werden, die weitere Einsatzmöglichkeiten zugänglich machen kann. Natürlich kann über den Dialog auch zu der Transistor-Tester-Funktion zurückgekehrt werden.
32. Mit Dialogfunktion kann am PD4-Port des ATmega eine Frequenzmessung vorgenommen werden. Die Auflösung beträgt bei Eingangsfrequenzen über $33kHz$ ein Hertz. Bei niedrigeren Frequenzen kann die Auflösung bis zu $0,001mHz$ betragen. Lesen Sie bitte das Unterkapitel 2.2.4 auf Seite 14, wie ein Frequenzsignal angeschlossen werden muß.
33. Mit Dialogfunktion und ohne serielle Ausgabe kann eine externe Spannung bis $50V$ über einen 10:1 Spannungsteiler am PC3 Pin gemessen werden. Bei der PLCC-Version des ATmega328

kann auch einer der beiden zusätzlichen Pinne für die Spannungsmessung zusammen mit der seriellen Ausgabe benutzt werden. Wenn die Erweiterung für die Zenerdiodenmessung (DC-DC-Konverter) eingebaut ist, können in diesen Zweig auch Zenerdioden bei gedrückter Taste getestet werden.

34. Mit Dialogfunktion kann eine Frequenzausgabe auf dem TP2-Pin (PB2-Port des ATmega) erfolgen. Derzeit können von Frequenzen von 1Hz bis 2MHz eingestellt werden.
35. Mit Dialogfunktion kann eine feste Frequenz mit einstellbarer Pulsweite auf dem TP2-Pin (PB2-Port des ATmega) ausgegeben werden. Die Breite kann mit kurzem Tastendruck um 1% und mit längerem Tastendruck um 10% erhöht werden.
36. Mit der Dialogfunktion kann eine spezielle Kondensatormessung mit ESR-Messung gestartet werden. Die Funktion wird bei der Auswahl C+ESR@TP1:TP3 genannt. Kapazitäten ab etwa $2\mu\text{F}$ bis zu 50mF können meistens wegen der geringen Messspannung von etwa 300mV im eingebauten Zustand gemessen werden.
37. Bei Prozessoren mit mindestens 32K Flash-Speicher (Mega328) kann der ADC mit der Sampling-Methode so genutzt werden, daß Kondensatoren unter 100pF mit einer Auflösung von 0.01pF gemessen werden können. Mit der gleichen Methode können auch Spulen unter 2mH mit deutlich besserer Auflösung über die Resonanzfrequenz mit einem parallelgeschalteten Kondensator bekannter Größe bestimmt werden.

Bei Einsatz des Testers für die Prüfung von Kondensatoren in einer Schaltung sollte besonders darauf geachtet werden, dass die Kondensatoren vor der Messung keine Restladung mehr haben.

Thyristoren und Triacs können nur erkannt werden, wenn der Test-Strom über dem Halte-Strom liegt. Einige Thyristoren und Triacs brauchen auch einen höheren Zündstrom als dieser Tester liefern kann. Der verfügbare Teststrom ist nur ungefähr 6mA ! Ebenso können IGBTs nur erkannt werden, wenn eine Spannung von 5V für die Gate-Ansteuerung reicht. Beachten Sie bitte, dass viele Möglichkeiten nur für Mikrocontroller mit wenigstens 16K Programmspeicher wie ATmega168 zur Verfügung stehen. Alle Eigenschaften sind sogar nur für Prozessoren mit wenigstens 32K Programmspeicher wie ATmega328 oder ATmega1284 möglich.

Achtung: Stellen Sie immer sicher, dass **Kondensatoren** vor dem Anschluss an den Tester **entladen** sind! Der Tester könnte sonst beschädigt werden bevor er eingeschaltet ist. Es gibt nur wenig Schutzfunktion der ATmega-Anschlüsse. Besondere Vorsicht ist auch geboten, wenn versucht wird, Bauelemente in einer Schaltung zu messen. Das Gerät sollte in jedem Fall vorher von der Strom einspeisung getrennt sein und man sollte sicher sein, dass **keine Restspannung** im Gerät vorhanden ist.

Kapitel 2

Hardware

2.1 Die Schaltung des TransistorTesters

Die Schaltung des TransistorTesters in Abbildung 2.1 basiert auf der Schaltung von Markus F., die er in Abb. 1 des AVR-Transistortester Reports [2] veröffentlicht hat. Geänderte oder verschobene Bauteile sind mit **grüner Farbe** markiert, optionale Teile sind mit **roter Farbe** gekennzeichnet.

Einige Änderungen wurden gemacht, weil die Strom-Abschaltung in einigen Nachbauten Probleme bereitet hatte. Deshalb ist der Widerstand R7 auf $3,3k\Omega$ reduziert. Der Kondensator C2 ist auf $10nF$ verkleinert und der Widerstand R8 ist verschoben, so dass der Ausgang PD6 nicht versucht, den C2-Kondensator direkt zu laden. Zusätzliche Abblock-Kondensatoren wurden hinzugefügt und sollten nahe den Versorgungs-Anschlüssen des ATmega und nahe bei dem Spannungsregler plaziert werden. Weil der PD7-Eingang und der PC6-Anschluss (RESET) die einzigen Anschlüsse sind, wo „pull-up“ Widerstände gebraucht werden, wurde ein zusätzlicher $27k\Omega$ Widerstand am PD7 (Pin 13) vorgesehen. Mit dieser Änderung können die internen „Pull-Up“-Widerstände des ATmega abgeschaltet werden. Ein Quarz mit seinen $22pF$ -Kondensatoren wurde zusätzlich vorgesehen. Ein Quarz hat Vorteile für die Kapazitätsmessung wegen der genaueren Zeitmessung.

Die neue Software kann den Spannungsbereich für den ADC umschalten. Die Umschalt-Geschwindigkeit wird durch den externen Kondensator C1 am AREF-Pin (21) des ATmega reduziert. Um die Messung nicht langsamer als notwendig machen zu müssen, sollte der Kondensator auf $1nF$ reduziert werden. Ein Entfernen des Kondensators ist ebenfalls möglich. Zum Anpassen der Software an die jeweilige Schaltung schauen Sie bitte in dem Konfigurations-Kapitel 4.1 ab Seite 49 nach.

Einige unterschiedliche Kombinationen von R11 / R12 zirkulieren im Internet. Ich habe die Software an den Original-Entwurf von Markus F. [2] mit $10k\Omega$ und $3,3k\Omega$ angepasst. Da Spannungsverhältnis kann in der Makefile angepaßt werden.

Die zusätzliche 2,5V Präzisions-Spannungsreferenz, die an Pin PC4 (ADC4) angeschlossen ist, wird für die Überprüfung und Kalibration der VCC-Versorgungsspannung benutzt, ist aber nicht erforderlich. Sie können eine LM4040-AIZ2.5 (0,1%), eine LT1004CZ-2.5 (0,8%) oder eine LM336-Z2.5 (0,8%) Spannungsreferenz benutzen. Wenn sie weder die Präzisionsreferenz noch die Relais-Erweiterung zum Schutz der Eingänge benutzen, sollten Sie wenigstens einen „Pull Up“-Widerstand R16 an PC4 installieren mit einem höheren Widerstandswert ($47k\Omega$). Dies hilft der Software, die fehlende Spannungsreferenz zu entdecken. Ein zusätzlicher ISP-Anschluss wurde hinzugefügt, um leichter neue Software-Versionen laden zu können.

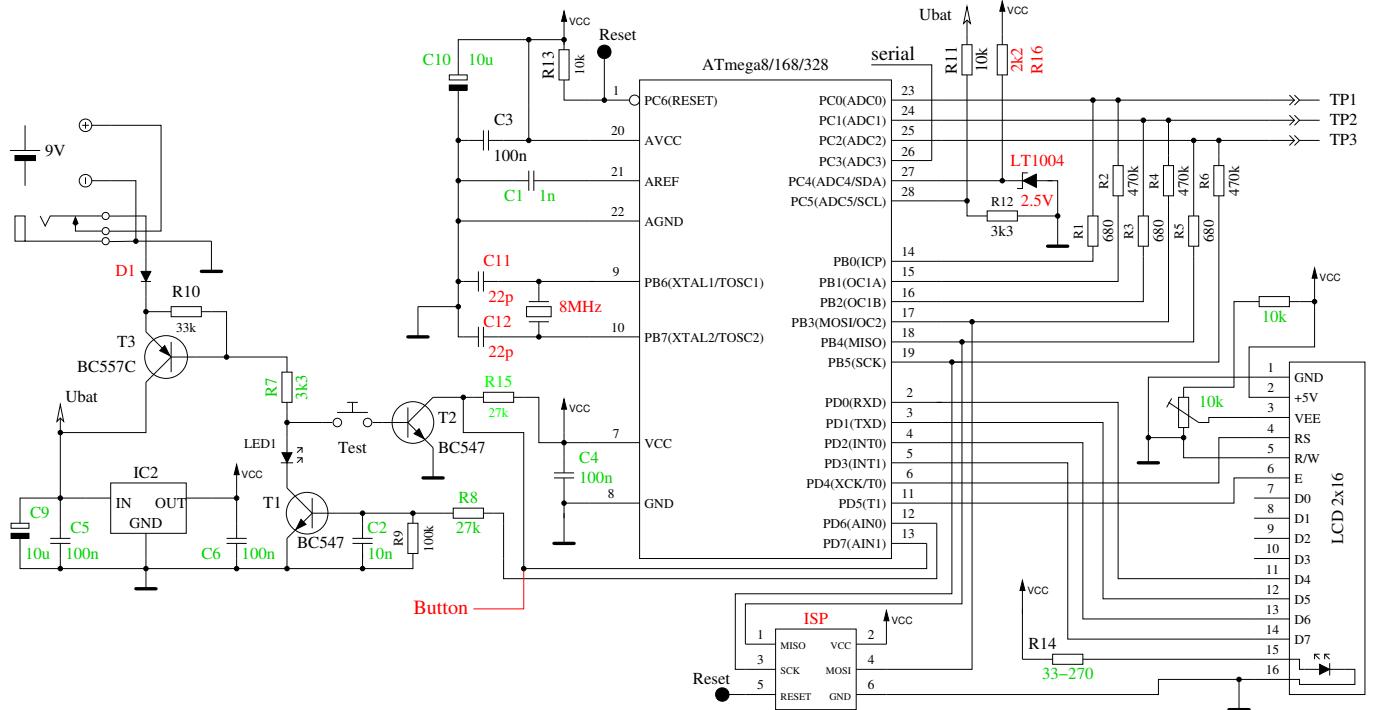


Abbildung 2.1. Neue TransistorTester-Schaltung

Die Tabelle 2.1 zeigt die Belegung des PD Ports für verschiedene Display-Varianten und die Belegung der Zusatzfunktionen. Bei allen Varianten dieser Tabelle sollten die Zusatzfunktionen möglich sein. Das Signal LCD-CE bei der SPI Schnittstelle ist am ATmega-Port verfügbar. Der Eingang CE (Chip Enable) des Controllers kann aber auch fest auf GND gelegt werden anstelle mit dem LCD-CE Ausgang des ATmega verbunden zu werden.

Port	Character LCD	ST7565 LCD SPI	ST7920 LCD serial	NT7108 LCD serial	SSD1306 I ² C	Zusatzfunktion
PD0	LCD-D4	LCD-REST	LCD-REST	595-PCLK		
PD1	LCD-D5	LCD-RS		LCD-CS2		Drehgeber-2
PD2	LCD-D6	LCD-SCLK	LCD-B0	164-595-CLK	LCD-SDA	
PD3	LCD-D7	LCD-SI		LCD-CS1		Drehgeber-1
PD4	LCD-RS			LCD-RS 164-595-SER		Frequenzzähler
PD5	LCD-E	(LCD-CE)	LCD-EN	LCD-EN	LCD-SCL	
PD7	Tastensignal	Tastensignal	Tastensignal	Tastensignal	Tastensignal	

Tabelle 2.1. Pinbelegung für verschiedene Displays

Um einen einfacheren Anschluss des Displays an den ATmega auf Streifenleiterplatten zu erreichen, kann die Software auch eine andere Port-D-Belegung berücksichtigen. Die folgende Tabelle 2.2 zeigt die Änderungen der Pinbelegungen für ein Textdisplay und alternative Anschlüsse für graphische Displays bei einem ATmega328 Mikrocontroller. Außerdem werden die Belegungen der Port-Eingänge für zusätzliche Funktionen gezeigt. Bei der Belegung für das grafische Display mit der Streifenraster-Option (STRIP_GRID_BOARD=1) kann die Frequenzzähler-Funktion nicht benutzt werden, da diese den Port PD4 (T0) benutzt. Diese Belegung wird aber von einer chinesischen Version mit graphischem Display benutzt. In den meisten Fällen sind die Platinenversionen mit einem Textdisplay für die Nachrüstung der Frequenzzähler Funktion und der Drehgeber Option besser geeignet, weil die erforderlichen Signale am Displayanschluß zur Verfügung stehen.

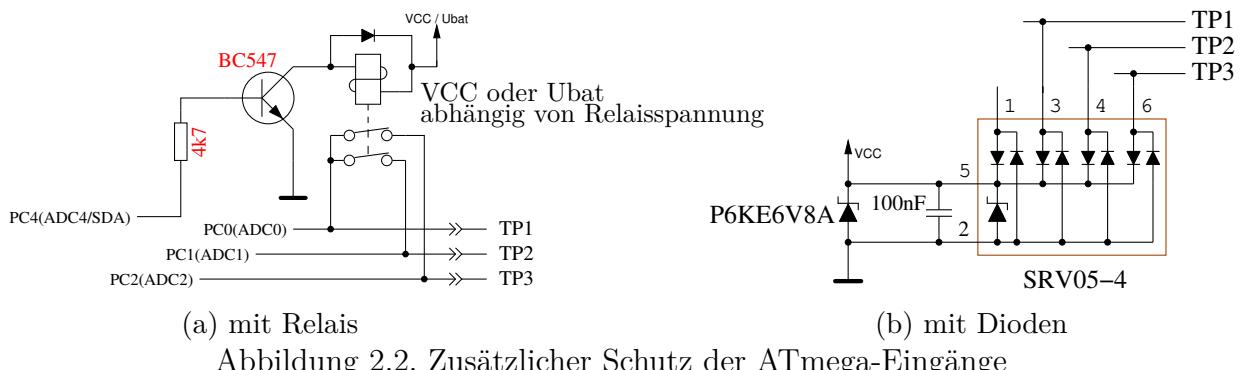
Port	Char. LCD =1	ST7565 LCD =1	ST7565 LCD =5	Zusatzfunktion
PD0	Tastensignal			
PD1	LCD-D7	LCD-SI	LCD-A0 (RS)	Drehgeber-2
PD2	LCD-D6	LCD-SCLK	LCD-REST	
PD3	LCD-D5	LCD-A0 (RS)	LCD-SCLK	Drehgeber-1
PD4	LCD-D4	LCD-REST	LCD-SI	Frequenzzähler
PD5	LCD-E (LCD-CE)			
PD7	LCD-RS	Tastensignal	Tastensignal	

Tabelle 2.2. Alternative Pinbelegungen mit der Option STRIP_GRID_BOARD

2.2 Erweiterungen für den TransistorTester

2.2.1 Schutz der ATmega-Eingänge

Zum besseren Schutz der ATmega-Eingänge kann eine Erweiterung mit einem Relais oder mit Dioden nach Schaltbild 2.2 angeschlossen werden. Die Ruhekontakte des Relais schützen den ATmega im spannungslosen Zustand. Die Konakte werden von der Software nur für die Messung freigegeben. Auch der Einbau von einem Überspannungsschutz mit Dioden verbessert die Chancen des ATmega den Anschluss eines Kondensators mit höherer Restspannung zu überleben. Ein vollständiger Schutz ist aber nicht möglich. Deshalb sollten Kondensatoren vor dem Messen immer entladen werden.



(a) mit Relais

(b) mit Dioden

Abbildung 2.2. Zusätzlicher Schutz der ATmega-Eingänge

Ein noch besserer Schutz ist mit einem Relais mit 3 Umschaltkontakten nach Abbildung 2.3 möglich. Der Entladestrom wird hier durch Widerstände begrenzt und die ATmega-Eingänge sind im geschützten Zustand getrennt. Man darf aber nicht vergessen, daß der Tester während der Messung trotzdem ungeschützt bleibt.

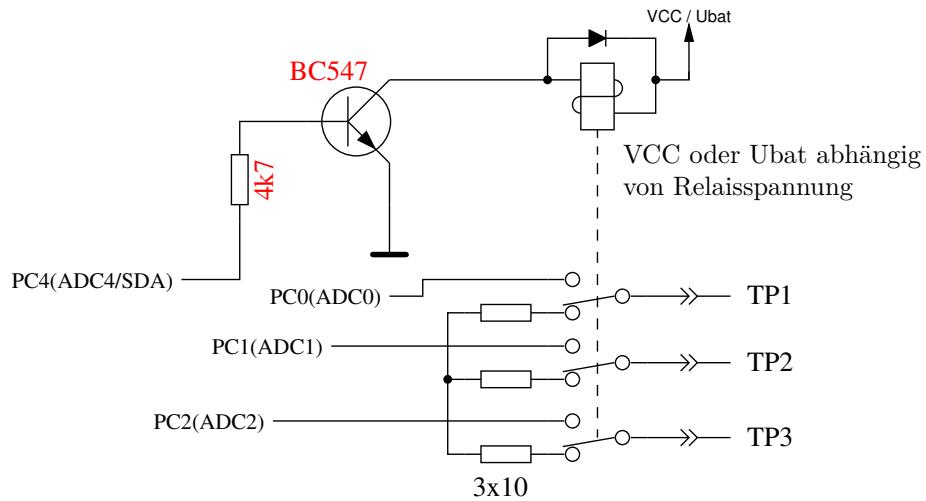


Abbildung 2.3. Verbesserter Schutz mit Relais

2.2.2 Zenerspannungsmessung

Wenn die Ausgabe der seriellen Texte nicht gebraucht wird, kann der Pin PC3 des ATmega für die Messung einer externen Spannung benutzt werden. Die Spannung kann mit dem optionalen 10:1-Widerstandsteiler bis zu 50V betragen und kann für die Messung der Zenerspannung einer Diode benutzt werden. Ein Strombegrenzendes Netzteil mit bis zu 50V Ausgangsspannung kann mit dem 0V-Signal des ATmega PD7-Pins eingeschaltet werden, um zum Beispiel die Durchbruchspannung von Zenerdiode zu messen. Einen Vorschlag für diese Erweiterung zeigt Abbildung 2.4. Der Tester zeigt die externe Spannung so lange an, wie man den Taster gedrückt lässt. Ungefähr 40mA mehr Batteriestrom wird für diese Erweiterung bei gedrückter Taste gebraucht.

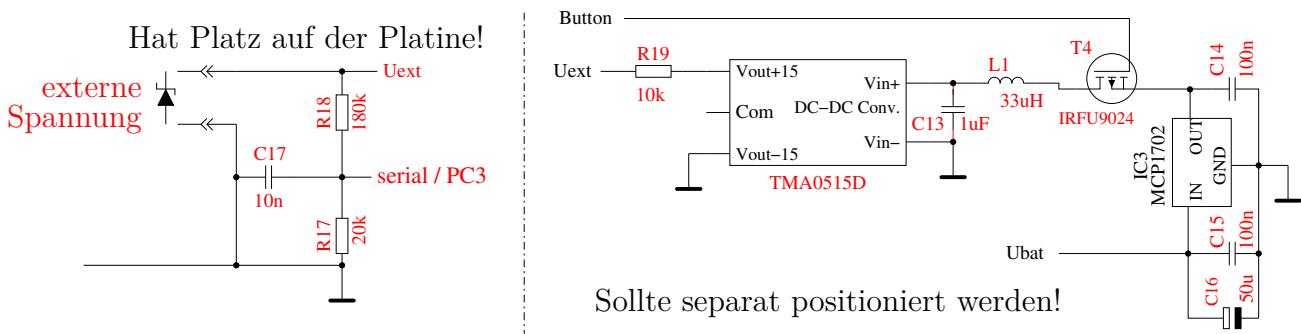


Abbildung 2.4. Erweiterung zum Messen von Zenerspannungen

Der 10:1-Spannungsteiler kann mit dem optionalen Dialogteil für den ATmega328 auch ohne den Spannungswandler benutzt werden. Ohne die gedrückte Taste ist der Spannungswandler nicht in Betrieb, sodass eine externe Spannung (z.B. Batteriespannung) am Zenerdioden-Port gemessen werden kann. Es können nur positive Gleichspannungen bis 50V gemessen werden. Man muss also auf die richtige Polarität achten.

2.2.3 Frequenzgenerator

Mit dem Dialogteil des ATmega kann außerdem ein Frequenzgenerator angewählt werden, der derzeit Frequenzen von 2MHz bis 1Hz ausgeben kann. Der Ausgang des 5V-Signals erfolgt über einen 680Ω -Widerstand auf den Testport TP2. Als Massesignal kann der Minus-Anschluss der Zenerdiodenerweiterung oder Anschluss TP1 benutzt werden. Auch TP3 ist über den 680Ω Widerstand mit Masse verbunden. Natürlich kann an den ATmega-Port PB2 auch eine Schaltung zum Verstärken

des Ausgangssignals für einen getrennten Ausgang angeschlossen werden. Dabei sollte der Eingang dieser Schaltung aber keine hohe kapazitive Last für den ATmega-Ausgang darstellen.

2.2.4 Frequenzzähler

Für die ebenfalls über die Dialogfunktion wählbare Frequenz-Messung ist eine kleine Erweiterung der Schaltung erforderlich. Als Eingang für die Frequenzmessung wird der PD4-Pin (T0/PCINT20) des ATmega benutzt. Der gleiche Pin wird auch zum Anschluss des LCD benutzt. Bei dem normalen Layout ist dies LCD-RS, beim Streifenraster-Layout ist dies LCD-D4. Für beide Signale kann der PD4-Pin auf Eingang geschaltet werden und für die Messung benutzt werden, solange keine Ausgabe auf das LCD erfolgt. Das LCD interessiert sich für das Signal nur, wenn LCD-E auf GND geschaltet wird. Für die Einspeisung des Testsignals ist mindestens ein Serienwiderstand von 270Ω erforderlich. Besser ist eine Schaltung nach Abbildung 2.5 zu benutzen. Die Spannung am PD4-Pin (LCD-RS oder LCD-D4) sollte ohne eingesteckten ATmega oder im Frequenz-Messbetrieb auf etwa 2,4V eingestellt sein, damit die größte Empfindlichkeit für das Eingangssignal erzielt wird. Das LCD sollte aber eingesteckt sein, weil dessen Pull-Up Widerstände die Spannung verändern.

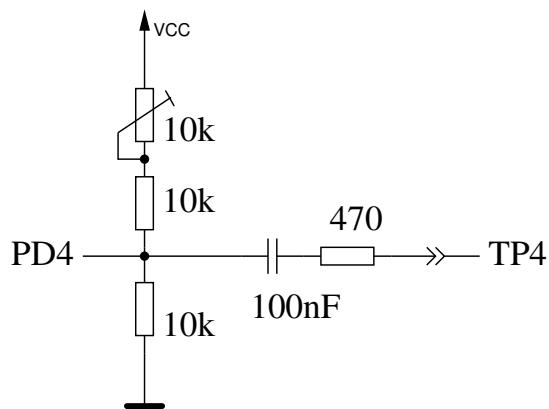


Abbildung 2.5. Erweiterung zum Messen von Frequenzen

2.2.5 Impulsdrehgeber

Zur leichteren Bedienung der Menüfunktion für den ATmega328 kann die Schaltung um einen Impulsdrehgeber mit Taster erweitert werden. Die Schaltung 2.6 gibt die Standard-Belegung für ein normales LCD an. Alle Anschlüsse für den Impulsdrehgeber sind an der Steckerleiste für das LCD verfügbar. Deswegen kann der Impulsdrehgeber in vielen Fällen leicht nachgerüstet werden. Vielfach ist das graphische Display auch mit einem Adapter-Board an die Steckerleiste des LCD angeschlossen. Deswegen ist auch hier die Nachrüstung des Impulsdrehgebers nicht sehr schwierig.

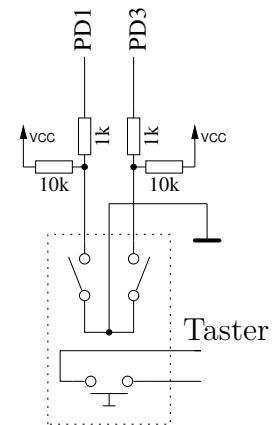


Abbildung 2.6.
Impulsdrehgeber-Erweiterung

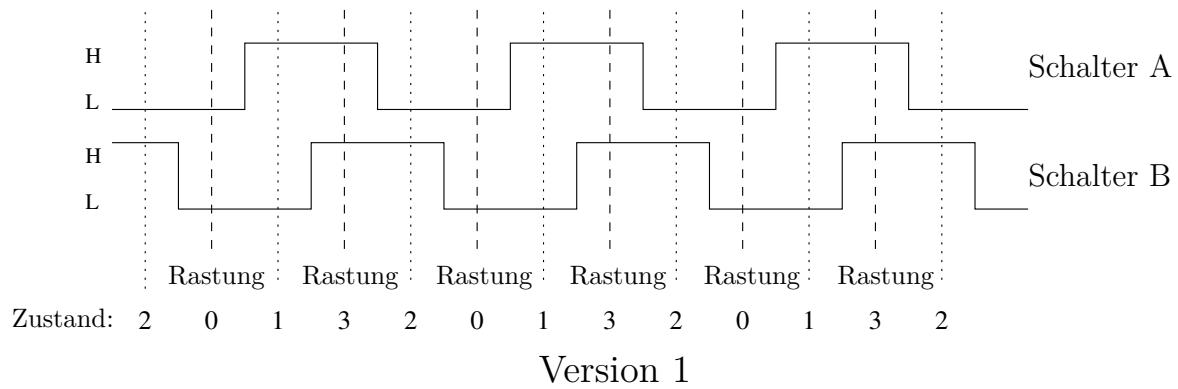
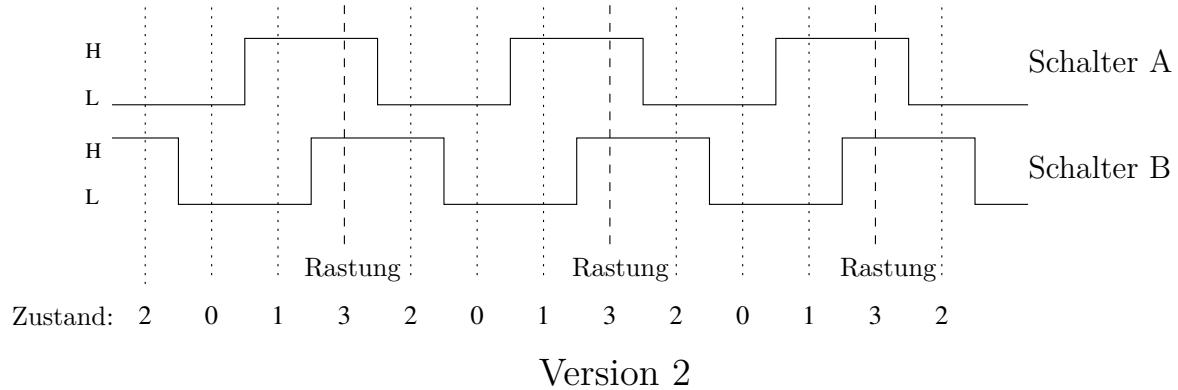
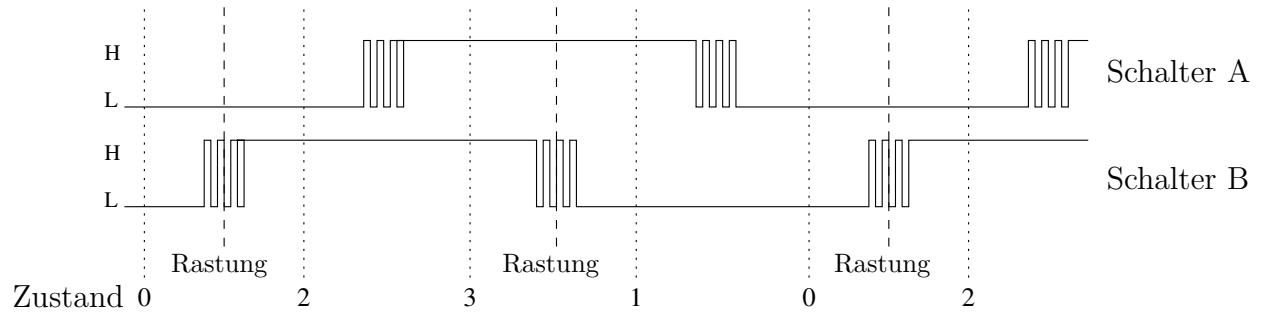


Abbildung 2.7. Zwei unterschiedliche Versionen eines Impulsdrehgeber

Die Abbildung 2.7 zeigt zwei Versionen von Drehgebern. Die Version 1 hat doppelt so viele Raststellungen (detent) pro Umdrehung wie Pulse pro Umdrehung. Die Version 2 hat gleich viele Pulse pro Umdrehung wie Raststellungen. Manchmal liegt bei einigen Drehgebern die Schaltflanke eines der beiden Schalter genau an der Raststellung.



Mögliche Zustände von links nach rechts:

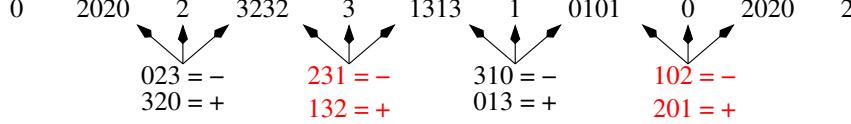


Abbildung 2.8. Ein Impulsdrehgeber mit prellenden Kontakten

Die Abbildung 2.8 zeigt einen Impulsdrehgeber, der nicht nur prellendekontakte hat, sondern auch an den Raststellungen (detent) einen unsicheren Zustand eines der Kontakte besitzt. Jeder Wechsel eines Schalterzustandes wird vom Programm überwacht und in einem zyklischen Puffer gespeichert. Damit kann bei einem Wechsel des Zustandes auch die beiden vorigen Zustände geprüft werden. Insgesamt können für einen Zyklus der Schaltzustände vier Zustandsfolgen für jede Drehrichtung

festgelegt werden. Wenn nur eine Raststellung pro Zyklus vorhanden ist, reicht die Abfrage eines Paares dieser Zustandsfolgen für das Zählen der Raststellung in beide Richtungen aus (WITH_ROTARY_SWITCH=2 oder 3). Bei zwei Raststellungen, wie in der Abbildung 2.8 gezeigt ist, müssen zwei Paare abgefragt werden (WITH_ROTARY_SWITCH=1). Für Impulsdrehgeber ohne Rasterung kann die Einstellung für WITH_ROTARY_SWITCH beliebig auf 2 oder 3 für die niedrigste Empfindlichkeit, auf 1 für eine mittlere Empfindlichkeit und auf 5 für die höchste Empfindlichkeit gesetzt werden. Ein Pendeln der Einstellung (Zähler rauf, Zähler runter) wird durch die Art der Abfrage vermieden, bei ungünstiger Lage der Schaltflanken kann aber ein Zählpuls für eine Raststellung ausbleiben. Anstelle der beiden Kontakte für den Impulsdrehgeber können auch zwei Taster für Rauf (Up) und Runter (Down) eingebaut werden, wenn kein Impulsdrehgeber vorhanden oder gewünscht ist. Für diesen Fall muss die Option WITH_ROTARY_SWITCH auf 4 gesetzt werden, damit das Programm entsprechend reagiert.

2.2.6 Anschluss eines graphischen Displays

Dank der Arbeit von Wolfgang Sch. für die Unterstützung der chinesischen Version mit grafischem 128x64 Pixel LCD, kann mittlerweile auch ein grafisches LCD mit ST7565-Controller angeschlossen werden. Da der ST7565-Controller seriell angesteuert wird, werden nur vier Signalleitungen benötigt. Dadurch werden zwei Pinne des Port D für andere Verwendung frei. Der ATmega-Prozessor sollte mindestens 32k Flash-Speicher besitzen. Der ST7565-Controller wird mit einer Betriebsspannung von 3,3V betrieben. Deswegen ist ein zusätzlicher Spannungsregler notwendig. Nach dem Controller Datenblatt dürfen keine 5V Signale direkt mit den Controller-Eingängen verbunden werden. Deshalb ist in der Erweiterung nach Bild 2.9 ein zusätzlicher CMOS 74HC4050 für die Pegelanpassung vorgesehen. Man kann auch versuchen, die vier Gatter des 74HC4050 durch vier Widerstände von etwa $2,7k\Omega$ zu ersetzen. Über den Spannungsabfall an den Widerständen wird verhindert, dass die 3,3V-Versorgung des Controllers über die Schutzdiode der Controllereingänge von den 5V ATmega-Ausgängen über die 3,3V-Grenze angehoben werden kann. Ob die Signalform für den ST7565-Controller so noch akzeptiert wird, sollte man vorher testen. Mit den Gattern des 74HC4050 bleibt die Signalform jedenfalls besser erhalten.

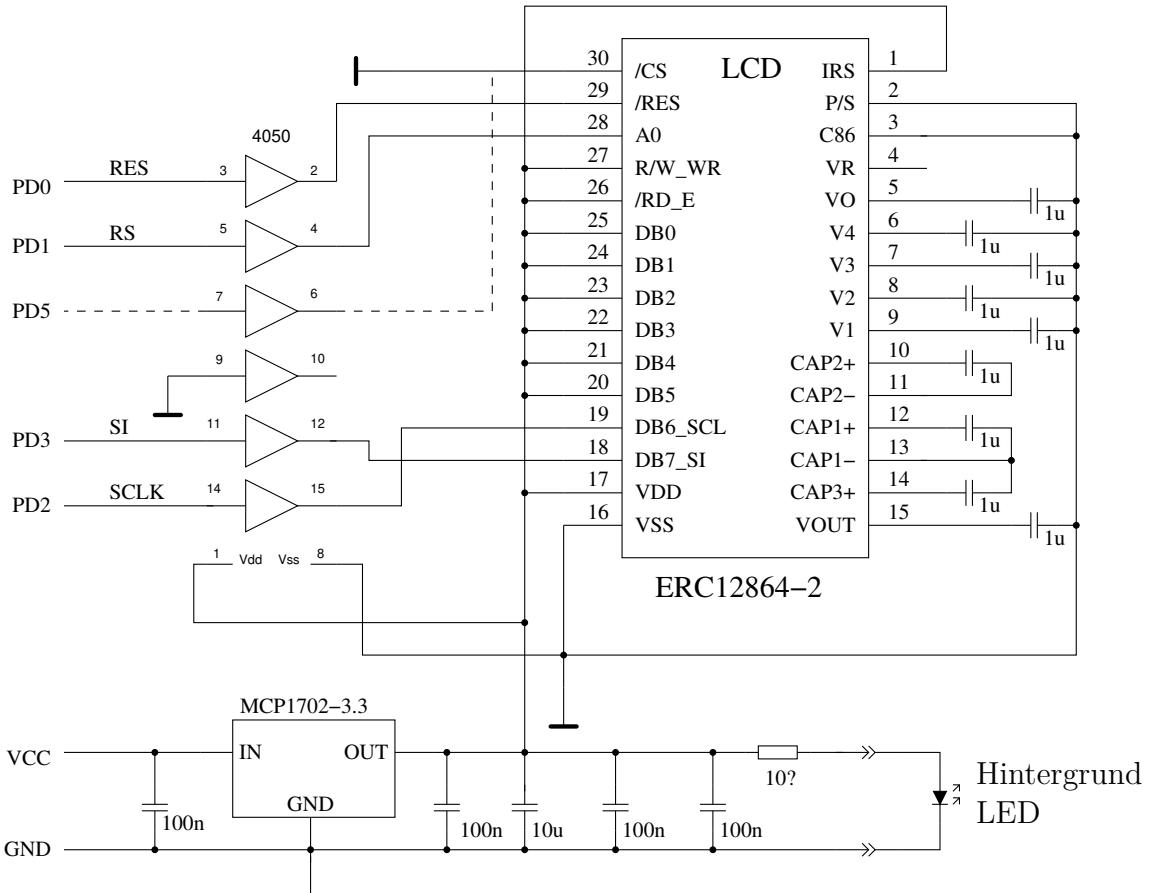


Abbildung 2.9. Anschluß eines graphischen Displays mit ST7565 Controller

Die Tabelle 2.3 zeigt weitere Anschlußmöglichkeiten für den ATmega328 und andere Prozessoren mit der SPI (LCD_INTERFACE_MODE=4) oder der 3LINE (LCD_INTERFACE_MODE=3) Schnittstelle.

Verschiedene Belegungen für einen Prozessor werden mit der Makefile Option STRIP_GRID_BOARD eingestellt. Die Belegungen sind in der Datei config.h festgelegt. Wenn weitere Varianten der Belegung erforderlich sind, sollten diese mit weiteren Kennziffern der Option STRIP_GRID_BOARD auswählbar sein und in config.h ergänzt werden.

Processor STRIP_GRID_BOARD	m644	m1280	m1280 1	m328	m328 1	m328 2	m328 5
Signal: RES	PB4	PA0	PA4	PD0	PD4	PD0	PD2
EN, CLK	PB6	PA2	PA2	PD2	PD2	PD2	PD3
RS, D/C	PB5	PA1	PA3	PD1	PD3	PD3	PD1
B0, MOSI	PB7	PA3	PA1	PD3	PD1	PD1	PD4
CE, CS	PB3	PA4	PA5	PD5	PD5	PD5	PD5

Tabelle 2.3. SPI Anschlußbelegungen für verschiedene Prozessoren

Normalerweise wird der ST7565- oder der SSD1306-Controller mit einer 4-Wire SPI-Schnittstelle angeschlossen. Beim SSD1306-Controller kann auch eine I²C-Schnittstelle mit PD2 als SDA- und PD5 als SCL-Signal benutzt werden. Die SDA- und SCL-Signale müssen mit einem „Pull-up“ Widerstand von ungefähr 4,7kΩ nach 3,3V versehen sein. Eine Anschlußmöglichkeit zeigt die Abbildung 2.10. Vor der Verwendung der „Pull-up“ Widerstände nach 5V sollte überprüft werden, ob die Eingänge dieses Signal vertragen. Normalerweise sind die Eingänge des Controllers über Dioden nach 3,3V

geschützt. Die Ausgänge des ATmega werden für die I²C-Signale nur nach 0V geschaltet. Es sollte aber sichergestellt sein, daß das Programm mit der I²C Schnittstelle in den ATmega geladen wurde, bevor das Display angeschlossen wird. Wenn ein Programm für eine andere Schnittstelle geladen wurde, werden die Ausgänge auch nach 5V geschaltet. Da ich eine Beeinflussung der Meßergebnisse der Testers über den VCC-Anschluß der OLED-Module festgestellt habe, ist eine Entkopplung über einen Serienwiderstand von 68Ω mit zusätzlichem $10\mu F$ Abblock-Kondensator zu empfehlen. Anstelle des 68Ω Widerstand kann auch eine Drossel von etwa $1mH$ verwendet werden. Ohne das zusätzliche Filter wurde bei meinem Tester Kollektorrestströme bei bipolaren Transistoren mit einem OLED angezeigt. Außerdem sollte die Belegung der Pinne beim OLED Modul geprüft werden, bei einigen Modulen ist GND und VCC vertauscht!

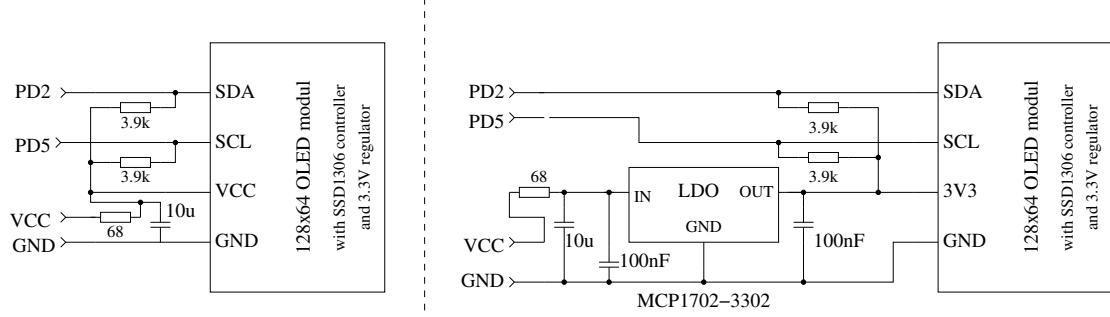


Abbildung 2.10. Anschluss eines graphischen OLED Displays mit I²C Schnittstelle

Bei der ATmega644 Prozessorserie werden die Pinne PB3 (SCL) und PB4 (SDA) anstelle PD5 und PD2 für den Anschluß benutzt. Bei der ATmega1280 Prozessorserie werden die Pinne PA5 (SCL) und PA4 (SDA) benutzt. Ein Austausch des Textdisplays gegen ein grafisches Display ist mit einer Adapterplatine möglich, da alle notwendigen Datensignale und Versorgungssignale an der LCD-Steckerleiste vorhanden sind. Etwas einfacher ist der Anschluß eines graphischen Displays mit einem ST7920 Controller weil der Controller mit 5V Betriebsspannung läuft. Dabei sollte das Display 128x64 sichtbare Pixel besitzen. Das Display Modul mit dem ST7920 Controller kann entweder mit der 4-Bit Parallelschnittstelle oder mit einer speziellen Seriellschnittstelle angeschlossen werden, wie die Abbildung 2.11 zeigt.

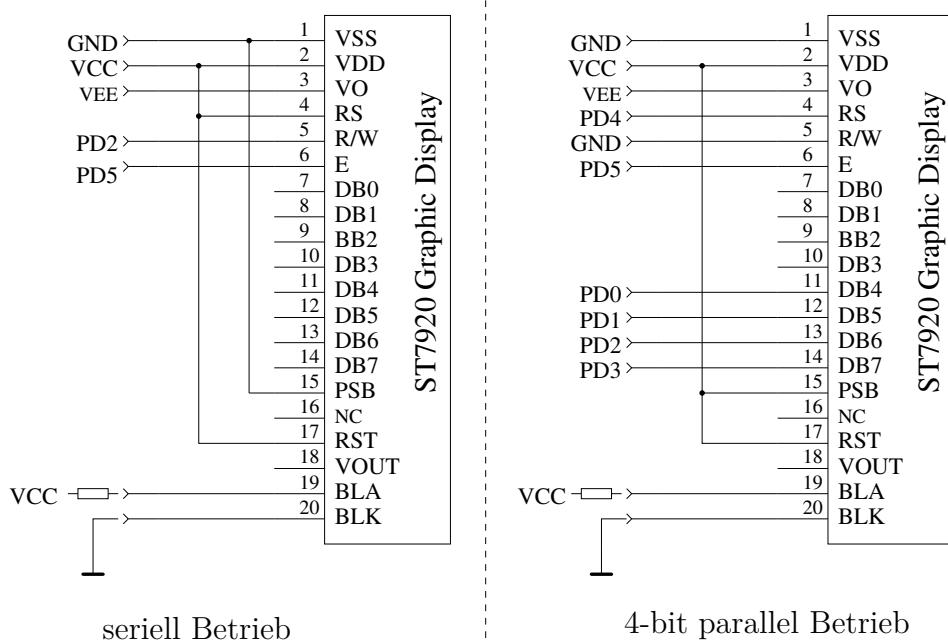


Abbildung 2.11. Anschluss eines Displays mit ST7920 Controller

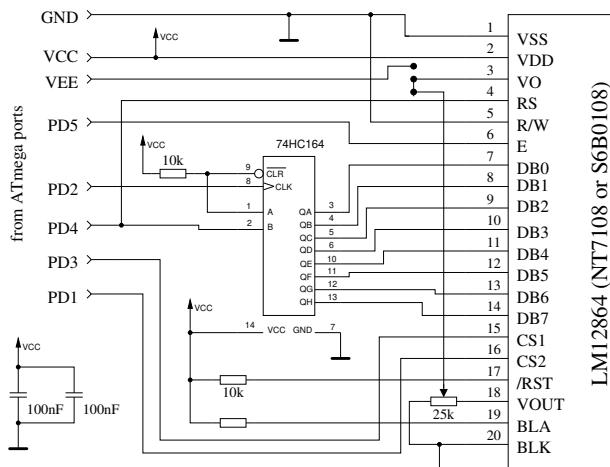
Für beide Anschlußarten muß die Software speziell konfiguriert werden. Die Makefile Option „WITH_LCD_ST7565 = 7920“ muß dazu auf jeden Fall gesetzt werden, für die serielle Anschlußart muß außerdem die Option „CFLAGS += -DLCD_INTERFACE_MODE=5“ gesetzt sein. Die Tabelle 2.4 zeigt die Belegung der seriellen Anschlußsignale bei dem Anschlußtyp 5 (ST7920) und 7 (SSD1803) für verschiedene Prozessoren.

Processor STRIP_GRID_BOARD	m644	m644 1	m1280	m328
Signal:				
EN	PB3	PB6	PA5	PD5
B0, R/W	PB4	PB7	PA4	PD2
RESET	PB2	PB4	PA0	PD0

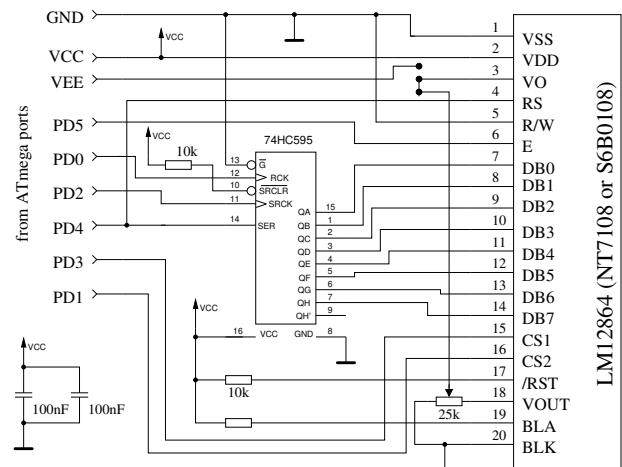
Tabelle 2.4. Serielle Anschlußbelegungen für verschiedene Prozessoren

Die Ausrichtung der Darstellung kann wie bei allen graphischen Displays mit den Optionen LCD_ST7565_H_FLIP und LCD_ST7565_V_FLIP angepaßt werden.

Einen Sonderfall stellen Displays mit NT7108 oder KS0108 (S6B0108) Controller dar. Da die Displays nur mit 8-Bit Parallelschnittstelle angesteuert werden können, ist ein externer seriell-paralleler Wandler erforderlich. Die einfachste Lösung scheint mir mit einem 74HCT164 oder einem 74HCT595 Chip möglich zu sein. Einen entsprechenden Schaltungsvorschlag zeigt das Bild 2.12.



(a) mit 74HCT164



(b) mit 74HCT595

Abbildung 2.12. Anschluß eines graphischen Displays mit NT7108 Controller

Sie müssen die Pinreihenfolge bei Ihrem LCD-Modul prüfen, einige Module haben eine andere Signalreihenfolge.

Einige verschiedene Pinbelegungen aus Datenblättern der ABG128064 Serie zeigt Tabelle 2.5.

Signal	128064H	128064G	128064C	128064B
VDD (5V)	1	2	4	2
VSS (GND)	2	1	3	1
VO (Drive)	3	3	(5)	3
DB0-DB3	4-7	7-10	9-12	7-10
DB4-DB7	8-11	11-14	13-16	11-14
CS1	12	15	1	15
CS2	13	16	2	16
Reset	14	17	-	17
R/W	15	5	7	5
RS	16	4	6	4
E	17	6	8	6
VEE	18	18	-	18
LEDA	19	19	17	(19)
LEDK	20	20	18	-

Tabelle 2.5. Pinbelegung verschiedener NT7108 Module

Die Tabelle 2.6 zeigt die Belegung des seriellen NT7108 Controller Anschluß für verschiedene Prozessor-Serien.

Processor	m644	m1280	m328
Signal:			
EN	PB3	PA5	PD5
RS	PB2	PA4	PD4
B0	PB2	PA4	PD4
CS1	PB7	PA3	PD3
CS2	PB5	PA1	PD1
CLK	PB6	PA2	PD2
PCLK	PB4	PA0	PD0

Tabelle 2.6. Serieller NT7108 Anschluß für verschiedene Prozessoren

Es können auch Displays mit einem PCF8814 Controller verwendet werden, wie sie beispielsweise in einem Nokia 1100 Handy verbaut sind. Hierbei ist zu prüfen, welches Interface von dem Display-Modul verwendet wird. Der PCF8814 Controller unterstützt die SPI-Schnittstelle als 3-line und 4-line, die I²C-Schnittstelle und eine spezielle 3-line Schnittstelle, bei der das Daten/Instruktion - Signal als erstes serielles Bit übertragen wird. Das Display besitzt nur 96x65 Pixel, daher werden keine großen graphischen Symbole für die Transistoren bei diesem Controller benutzt. Die Ausgabe ist also ähnlich wie auf einem Textdisplay. Wie bei den meisten graphischen Displays ist auch hier die Betriebsspannung 3.3 Volt. Deswegen ist eine Signalanpassung an die 5V Ausgänge des ATmega erforderlich. Für die SPI-Schnittstelle und die 3-line Schnittstelle können die Ausgänge des ATmega mit der Makefile Option LCD_SPI_OPEN_COL wie ein „Open Collector“ Ausgang benutzt werden. Es sind aber „Pull-Up“ Widerstände erforderlich oder es darf die Option PULLUP_DISABLE in der Makefile nicht gesetzt sein. Getestet ist derzeit nur die 3-line Schnittstelle.

Port	PCF8814 SPI	PCF8814 3-line	PCF8814 I ² C	Zusatzfunktion
PD0	LCD-REST	LCD-REST		
PD1	LCD-D/C	LCD-SCE		Drehgeber-2
PD2	LCD-SCLK	LCD-SCLK	LCD-SDIN	
PD3	LCD-SDIN	LCD-SDIN		Drehgeber-1
PD4				Frequenzzähler
PD5		LCD-EN	LCD-SCLK	

Tabelle 2.7. Pinbelegung für verschiedene Anschlussvarianten des PCF8814 Kontrollers

Es ist auch eine Unterstützung für einen PCF8812 controller mit 102x65 Pixeln bereits vorhanden, aber völlig ungetestet.

2.2.7 Anschluss eines graphischen Farbdisplays

Von chinesischen Händlern werden preisgünstige Farbdisplay-Module mit einer SPI Schnittstelle angeboten. Die Abbildung 2.13 zeigt die Rückseite der beiden unterstützten Module mit 128x128 Pixeln und 128x160 Pixeln. Diese Module sind sehr klein, so werden auch Texte und Symbole sehr klein dargestellt. Aber das Erscheinungsbild ist scharf und klar.



Abbildung 2.13. Rückseite der beiden Color-LCD's

Das 128x128 Pixel Modul benutzt einen ILI9163 Kontroller. Das 128x160 Pixel Modul benutzt einen sehr ähnlichen ST7735 Kontroller. Getestet habe ich die Module mit einem Adapterboard, der die Verbindungen der SPI-Signale und der Stromversorgung zu der Anschlußleiste des normalen Textdisplays herstellt. Die Anpassung der 5V Signale des ATmega an die 3.3V Eingänge des Kontrollers habe ich mit seriellen 10kΩ Widerständen vorgenommen. Die Hintergrundbeleuchtung (LED) ist für diese Module unbedingt erforderlich, da sonst nichts erkennbar ist. Aufgrund der hohen Pixelzahl in vertikaler Richtung können auf den Displays mehr Textzeilen dargestellt werden. Für das 128x128

Pixel Display können bis zu 8 Textzeilen mit einem 12x8 Font dargestellt werden. Für das 128x160 Pixel Display sind es sogar 10 Textzeilen. Auf dem Foto 2.14 ist das Ergebnis einer Messung eines Germanium Transistors auf einem 128x128 Pixel Display zu sehen.

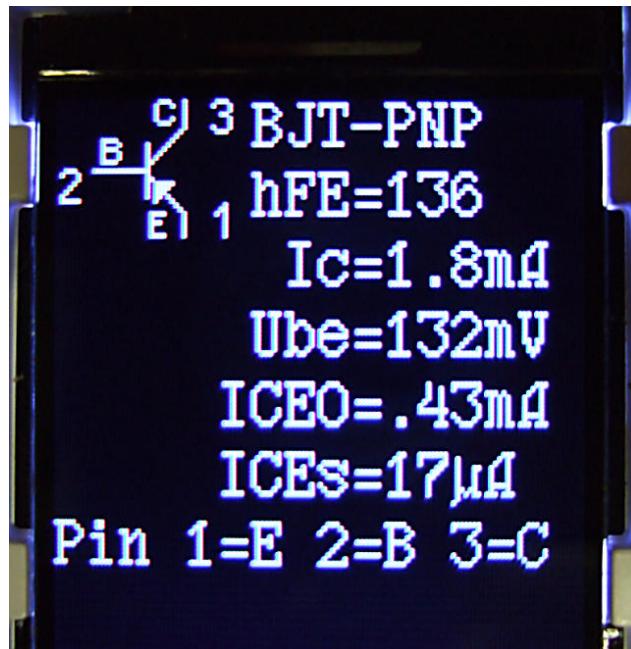


Abbildung 2.14. Messung eines bipolaren PNP Transistors

Die Farbmöglichkeiten der Displays werden derzeit nicht genutzt. Lediglich die Hintergrundfarbe und die Vordergrundfarbe könnten in der Datei lcdDefines.h oder in der Makefile verändert werden. Es wird das 16-Bit Farbmodell der Kontroller benutzt, die Vordergrundfarbe kann mit der Konstanten LCD_FG_COLOR und die Hintergrundfarbe mit der Konstanten LCD_BG_COLOR eingestellt werden.

2.3 Hinweise für den Aufbau des TransistorTesters

Jede LCD-Anzeige-Einheit mit mindestens 2x16 Zeichen und einem HD44780-kompatiblen Controller kann mit dem TransistorTester benutzt werden. Man sollte auf den Strombedarf der Hintergrundbeleuchtung achten, einige Anzeigen benötigen mehr Strom als andere. Ich habe OLED-Anzeigen ausprobiert, aber diese Anzeigen haben teilweise die Messung des ATmega beeinflusst und werden nicht empfohlen. Auch das Laden der Spezialzeichen für die Widerstandsdarstellung hat mit OLED's Schwierigkeiten ergeben.

Die Widerstände R1 bis R6 sind kritisch für die Messungen und diese 680Ω und $470k\Omega$ Widerstände sollten Messwiderstände sein (Toleranz von 0,1%), um die volle Genauigkeit zu erreichen. Man sollte Präzisions-Sockel für den ATmega-Mikrocontroller verwenden, um die Austauschbarkeit des Mikrocontrollers sicherzustellen. Es kann ein ATmega8, ATmega168 und ATmega328 Mikrocontroller verwendet werden. Empfohlen wird ein ATmega328, wenn man alle Funktionen nutzen möchte.

Jedenfalls sollte man zuerst alle Bauteile ohne den Mikrocontroller bestücken. Es wird als IC2 ein moderner „low voltage drop“-Spannungsregler wie MCP1702-5002 empfohlen, weil dieser nur $2\mu A$ Ruhestrom benötigt und auch noch 5V liefern kann, wenn die Eingangsspannung nur 5,4V beträgt. Aber dieser Regler ist leider nicht Pin-kompatibel zum bekannteren 78L05-Regler im TO-92-Gehäuse! Nachdem alle benötigten Bauteile bestückt sind, sollte zuerst die Batterie oder das Netzteil angeschlossen werden. Das LCD sollte dabei nicht angeschlossen sein, und der Mikrocontroller sich noch nicht im Sockel befinden. Man sollte die Betriebsspannung des Mikrocontrollers und der LCD-Anzeige überprüfen während der Start-Taster gedrückt wird. Die Betriebsspannung sollte verschwinden, wenn

man den Start-Taster loslässt. Wenn die Betriebsspannung die richtige Polarität und Grösse hatte, sollte man die Strom-Versorgung entfernen und den Mikrocontroller richtig herum einstecken. Seien Sie bitte vorsichtig und stellen Sie sicher, dass alle Pinne des Mikrocontrollers im Sockel stecken. Danach können Sie das LCD anschliessen. Prüfen Sie, dass die GND- und VCC-Anschlüsse des LCD richtig mit der Baugruppe verbunden sind.

Wenn Sie sicher sind, dass alles richtig angeschlossen ist, schliessen Sie die Spannungsversorgung wieder an. Wenn Sie den ATmega schon programmiert haben, können Sie den Start-Taster drücken. Durch das Drücken des Start-Tasters sollte die Hintergrundbeleuchtung der LCD-Anzeige angehen. Wenn Sie den Taster loslassen, sollte die LED auf der Platine schwach leuchten. Beachte, dass die Software für den Mikrocontroller für den richtigen Prozessor-Typ übersetzt sein muss. Ein Programm für den ATmega8 läuft nicht auf einem ATmega168!

2.4 Umrüstung von Tester Versionen nach Markus F.

Spannungsüberwachung Das Problem zeigt sich durch sofortiges Abschalten beim Einschaltversuch. Bei den von mir empfohlenen Einstellungen der fuses (Makefile) wird die Spannungsüberwachung der verschiedenen ATmega-Versionen auf 4V gesetzt (brown out level). Deswegen kann es beim Einschalten des Testers zu Problemen kommen, weil der Pin PD6 versucht, den $100nF$ -Kondensator C2 direkt zu schalten. Dabei kann es zu einem unerwünschten 5V Spannungseinbruch kommen. Der Kondensator C2 kann problemlos auf $< 10nF$ verkleinert werden. Nach Möglichkeit sollte man statt der direkten Verbindung PD6 zum Kondensator einen Widerstand $> 220\Omega$ als Verbindung benutzen.

Verbessern des Einschaltverhaltens Der Fehler zeigt sich oft, dass der Tester bei gedrücktem Taster zwar einschaltet, aber wieder abschaltet, wenn der Taster losgelassen wird. Das Problem tritt öfter auf, wenn die Hintergrund- Beleuchtung des LCD viel Strom braucht. Der Widerstand R7 zur Basis des PNP-Transistors T3 war mit $27k\Omega$ sehr auf Stromsparen optimiert. Der Widerstand sollte besser auf $3,3k\Omega$ verkleinert werden um auch bei geringerer Batteriespannung oder bei geringem Stromverstärkungsfaktor des PNP-Transistors T3 ein sicheres Einschalten zu gewährleisten.

Zusätzlicher Pull-Up-Widerstand an PD7 Der Fehler zeigt sich dadurch, dass der Tester nach einer kurzen Anzeigezeit mit der Meldung „Timeout“ abschaltet. Die Software ist standardmäßig so konfiguriert (Option PULLUP_DISABLE), dass die internen Pull-Up-Widerstände abgeschaltet sind. Dadurch ist der Pegel am Pin PD7 nicht mehr definiert, wenn er nicht durch den Taster oder T2 auf GND-Potential geschaltet ist. Ein externer Pull-Up-Widerstand von $27k\Omega$ nach VCC vermeidet diesen Fehler.

Kondensator C1 am AREF Pin In vielen Entwürfen wird hier ein $100nF$ -Kondensator verwendet, so auch im Entwurf vom Markus Frejek. Solange die Referenzspannung des ADC nicht verändert wird, ist das auch in Ordnung. Bei der Software für den Transistorsteller für den ATmega168/328 wird aber eine automatische Umschaltung der Referenzspannung von 5V auf die interne Referenzspannung von 1,1V vorgenommen, wenn die Eingangsspannung unter etwa 1V liegt. Damit wird eine bessere Auflösung erreicht. Leider erfolgt die Umschaltung von 5V auf die 1.1V sehr langsam, was eine zusätzliche Wartezeit von $10ms$ erfordert. Durch den Austausch des $100nF$ Kondensators gegen einen $1nF$ kann die Wartezeit deutlich verringert werden. Einen Einfluß des kleineren Kondensators auf die Qualität der Messergebnisse habe ich nicht feststellen können. Selbst das Entfernen des Kondensators hat keinen wesentlichen Einfluß. Wer den $100nF$ Kondensator unbedingt beibehalten möchte, kann die Makefile Option NO_AREF_CAP entfernen, um die längere Wartezeit im Programm zu aktivieren.

Nachrüsten eines 8MHz Quarz Mit etwas Geschick kann auf der Lötseite der Platine ein 8MHz Quarz direkt an PB6 und PB7 (Pin 9 und Pin 10) nachgerüstet werden. Bei meiner Nachrüstung habe ich auf die beiden 22pF-Kondensatoren verzichtet. Bei allen eingesetzten Prozessoren hat diese Lösung problemlos funktioniert. Aber die Nachrüstung ist nicht unbedingt erforderlich. Die Taktfrequenz sollte aber wegen der besseren Zeitkonstanten-Auflösung für die Kapazitäts-Messung auf jeden Fall 8MHz betragen. Die Frequenz 8MHz ist auch bei RC-Oszillator-Betrieb durch Setzen der fuses möglich.

Abblocken der Betriebsspannung Im Original-Schaltbild vom Markus F. ist nur ein 100nF-Kondensator zum Abblocken der VCC-Spannung (5V) eingezeichnet. Das ist deutlich zu wenig. Es sollte sowohl ein 100nF in unmittelbarer Nähe des ATmega als auch ein 100nF in unmittelbarer Nähe des Spannungsreglers vorhanden sein. Auch an den Eingang des Reglers gehört ein 100nF-Kondensator. Zusätzliche 10 μ F-Kondensatoren (Elektrolyt oder Keramik) am Eingang und Ausgang des Reglers können die Spannungsstabilität verbessern. Keramische 10 μ F Kondensatoren in SMD Bauform sind zum Nachrüsten meist besser geeignet und haben üblicherweise einen niedrigeren ESR-Wert.

Auswahl des ATmega-Prozessors Die Grundfunktion des Testers ist immer noch mit dem ATmega8 möglich. Dabei ist der Programmspeicher nahezu zu 100% benutzt. Da die ATmega168 oder ATmega328 Prozessoren pinkompatibel zum ATmega8 sind, kann der Austausch nur empfohlen werden. Mittlerweile sind die Preise für den ATmega328 so günstig, dass eigentlich nichts mehr für den ATmega168 spricht. Mit dem **ATmega168/328** gewinnt man folgende **Vorteile**:

- Selbsttestfunktion mit automatischem Abgleich.
- Erhöhung der Messgenauigkeit durch automatische Umschaltung der ADC-Referenzspannung.
- Messung von Induktivitäten, deren Widerstandswert < 2100 Ω ist.
- Messung des ESR-Wertes von Kondensatoren > 20nF.
- Die Auflösung der Widerstandsmessung unter 10 Ω beträgt 0,01 Ω .
- Der PC3-Pin kann für eine serielle Ausgabe genutzt werden.

Fehlende Präzisionsreferenz Normalerweise sollte die fehlende Präzisionsreferenz auch bei unbeschaltetem PC4-Pin erkannt werden. In diesem Fall wird keine VCC=x.xV Anzeige in Zeile 2 beim Einschalten angezeigt. Falls es zu der Anzeige kommen sollte, hilft ein nach VCC geschalteter 2,2k Ω Widerstand am PC4-Eingang.

2.5 Erweiterte Schaltung mit ATmega644 oder ATmega1284

Eine erweiterte Schaltung für ATmega644/1284-Prozessoren wurde in Zusammenarbeit mit Nick L. aus der Ukraine entwickelt. Die Schaltung nach Abbildung 2.16 ermöglicht zusätzlich einen Test von Quarzen und einen erweiterten Frequenzbereich für die Frequenzmessung. Obwohl die Grundschaltung sehr ähnlich der Schaltung 2.1 ist, werden hier andere Portbelegungen benutzt. Ein Impulsdrehgeber nach Schaltung 2.6 kann hier an PB5 und PB7 (statt PD1 und PD3) angeschlossen werden. Beide Signale und auch die Versorgungssignale VCC und GND sind am ISP-Stecker verfügbar, sodass die Erweiterung auch hier angeschlossen werden kann.

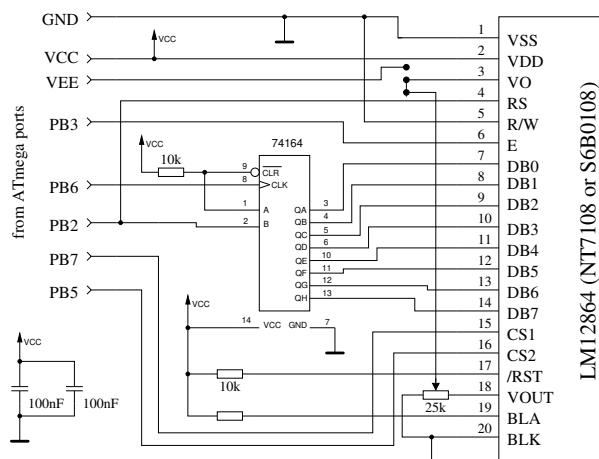
Der 16:1-Frequenzteiler 74HC4060 wird für höhere Frequenzen als 2MHz immer benutzt. Der Teiler kann aber auch für Frequenzen von 25kHz bis 400kHz benutzt werden, um mit der Periodenmessung die Auflösung der Frequenzmessung zu verbessern. Für die Umschaltung der Betriebszustände (Frequenzteiler und Quarz-Oszillator) werden die Analogschalter 74HC4052 benutzt. Die Tabelle 2.8 zeigt die Pinbelegung für den ATmega324/624/1284 Mikrocontroller für verschiedene Displayanschlüsse. Die I²C-Schnittstelle ist nur mit dem SSD1306-Controller möglich. Die Signale der I²C-Schnittstelle

erfordern einen „Pull-Up“-Widerstand von etwa $4,7k\Omega$ nach $3,3V$. Die Ausgänge des ATmega werden für die I²C-Signale nur nach $0V$ geschaltet.

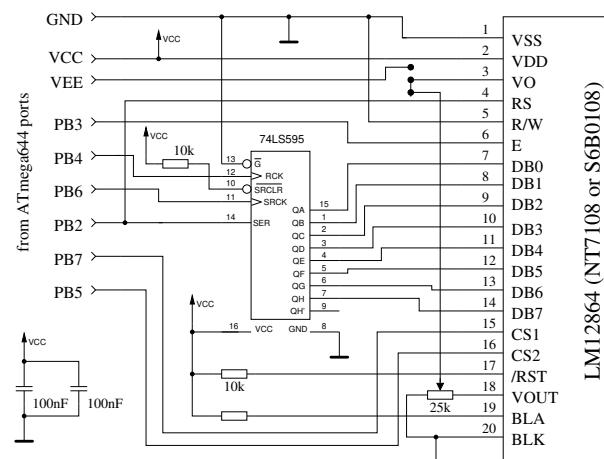
Port	Character LCD	Graphik LCD SPI 4-Wire	Graphik LCD I ² C	Zusatzfunktion
PB2	LCD-RS			
PB3	LCD-E	(LCD-CE)	LCD-SCL	
PB4	LCD-D4	LCD-REST	LCD-SDA	
PB5	LCD-D5	LCD-RS		ISP-MOSI Drehgeber 2
PB6	LCD-D6	LCD-SCLK		ISP-MISO
PB7	LCD-D7	LCD-SI		ISP-SCK Drehgeber 1

Tabelle 2.8. Verschiedene Varianten der Display-Anschlüsse

Auch ein Display mit NT7108 (KS0108, S6B0108) Controller kann mit einer kleinen Zusatzschaltung an einen ATmega644 oder ATmega1284 angeschlossen werden wie in der Abbildung 2.15 gezeigt. Bitte beachten Sie auch die verschiedenen Pinbelegungen der Displaymodule mit NT7108 Controller wie in Tabelle 2.5 auf Seite 20 angegeben.



(a) mit 74HCT164



(b) mit 74HCT595

Abbildung 2.15. Anschluss des NT7108 Controllers an einen ATmega644/1284

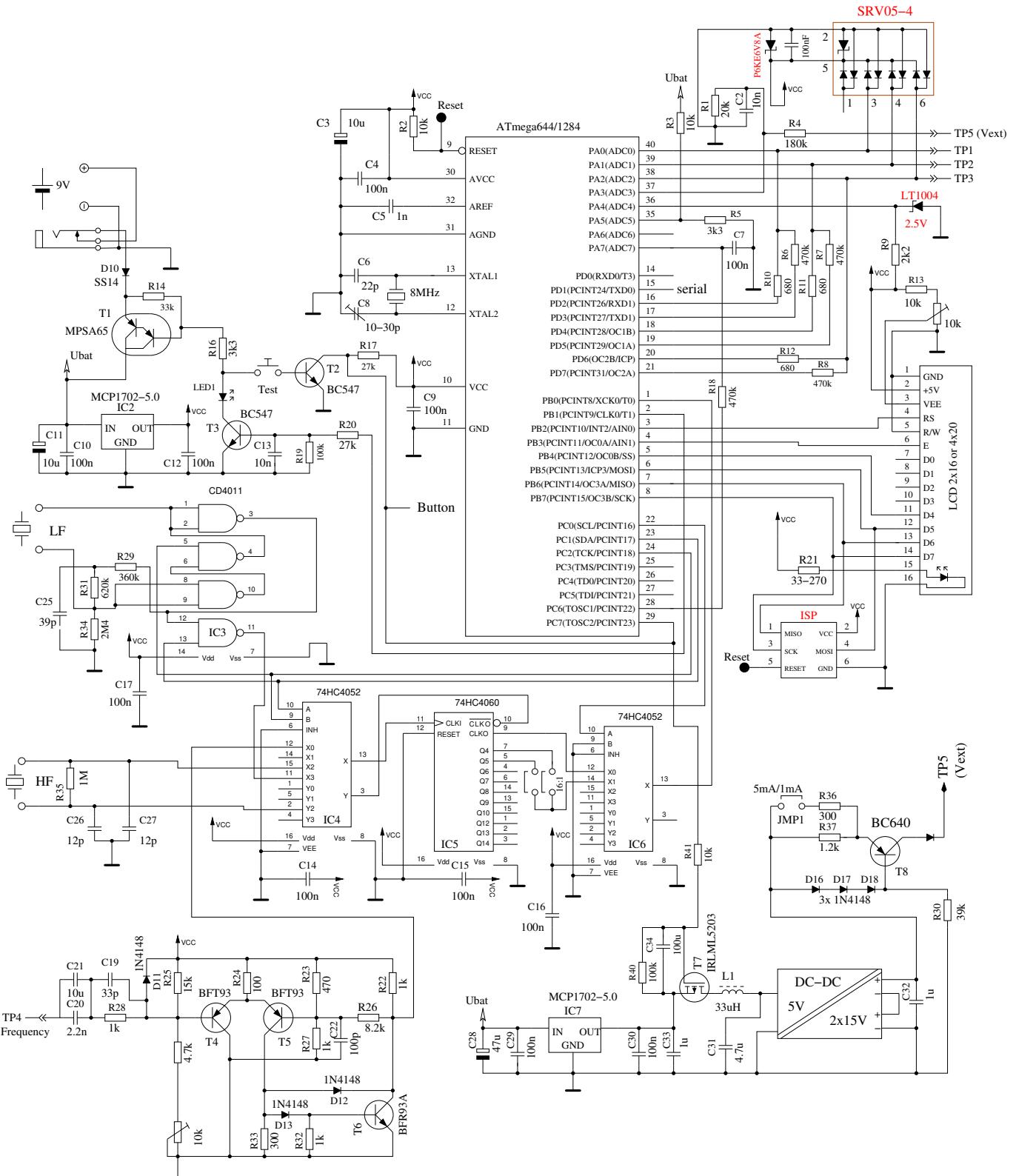


Abbildung 2.16. Erweiterte Transistortester Schaltung mit ATmega644

2.6 Aufbau mit ATmega1280 oder Arduino Mega

Die Grundschaltung des Testers lässt sich auch mit einem Arduino Mega mit ATmega1280 oder ATmega2560 als Shield aufbauen. Die notwendigen Verbindungen zeigt die Abbildung 2.17. Die Steckverbindungen des Arduino für die Datenverbindungen des Displays sind in grüner Farbe eingezzeichnet. Die Bauteile mit roter Bezeichnung sind für die Funktion des Testers nicht erforderlich. Der

ATmega2560 Controller hat zwar viele Anschlußpins, aber nur ein einziger Pin hat die erforderlichen Verbindungen für beide Methoden der Frequenzmessung. Der Anschlußpin muß als Takteingang für einen Zähler geschaltet werden können und außerdem muß der Pin ein Unterbrechungssignal (Interrupt) bei Pegelwechsel erzeugen können. Dies ist nur für den Pin PE6 (T3/INT6) der Fall. Die anderen Takteingänge der Zähler PD7 (T0), PD6 (T1), PH7 (T4) und PL2 (T5) können nicht für die Erzeugung des Unterbrechungssignals bei Pegelwechsel verwendet werden. Leider ist der PE6 Pin nicht mit den Steckerleisten des Arduino verbunden. Der PE5 Pin (7) ist mit dem Anschluß 3 der PWM Buchsenleiste des Arduino verbunden und kann am Atmega2560 mit dem PE6 Pin (8) gebrückt werden. Der Ausgang der Frequenzerzeugung ist am Pin PB6 (OC1B) verfügbar. Dieser Pin ist mit Anschluß 12 der PWM-Buchsenleiste verbunden. Auf den ISP-Stecker kann für den Arduino verzichtet werden, da das Programm über das USB-Interface und den Bootloader geladen werden kann. Mit dem Bootloader gibt es lediglich eine kleine Verzögerung des Programmstarts.

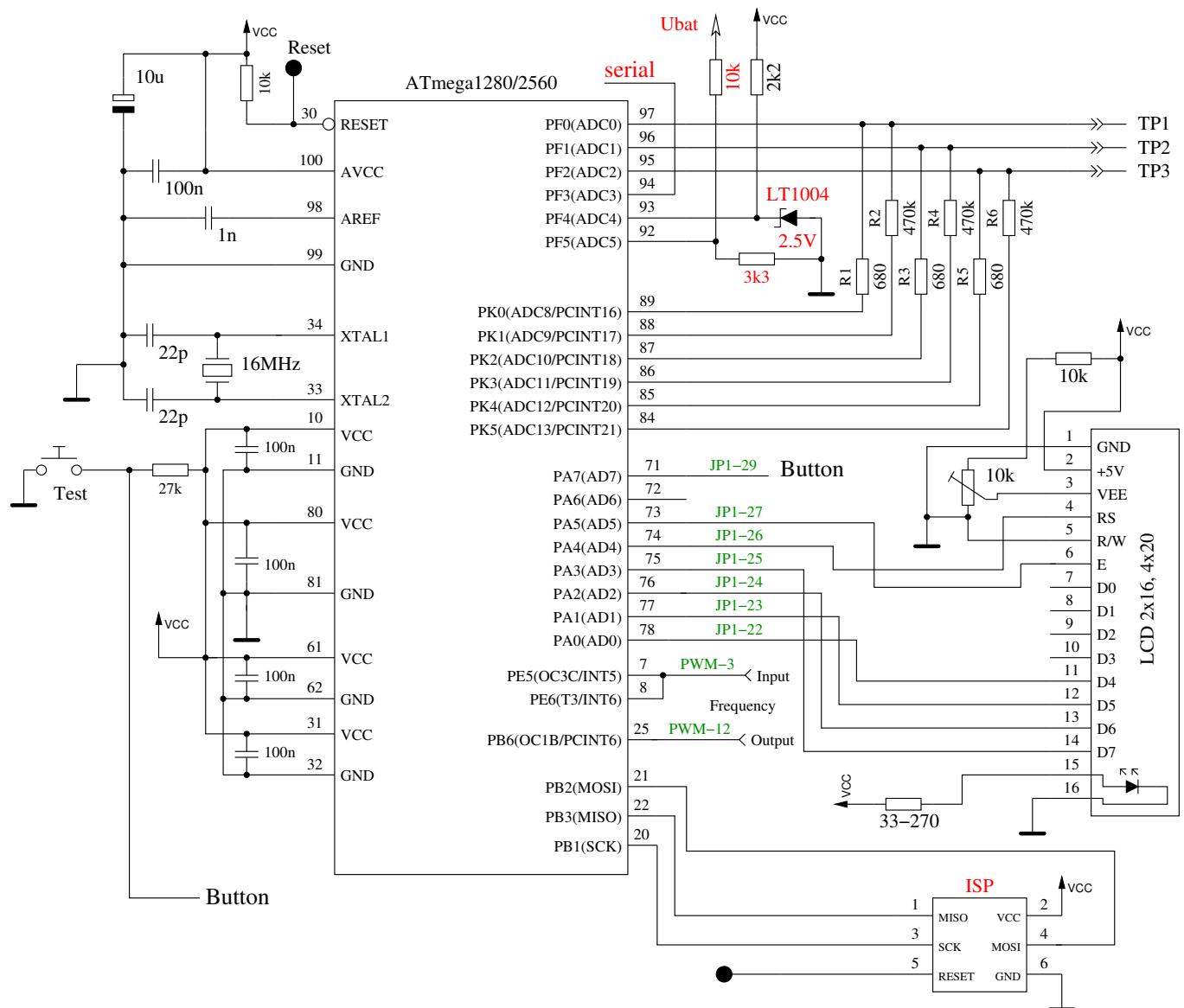


Abbildung 2.17. Transistortester Schaltung mit ATmega1280, ATmega2560 oder Arduino Mega

Natürlich können alle unterstützten Displays auch an die ATmega1280 oder ATmega2560 angeschlossen werden, wie in der Tabelle 2.9 angegeben.

Port	Character LCD	ST7565 SPI	ST7920 seriell	NT7108 seriell	SSD1306 I ² C	Zusatzfunktion
PA0	LCD-D4	LCD-REST	LCD-RESET	HC595-RCK		
PA1	LCD-D5	LCD-RS		LCD-CS2		Drehgeber-2
PA2	LCD-D6	LCD-SCLK		HC164-CLK		
PA3	LCD-D7	LCD-SI		LCD-CS1		Drehgeber-1
PA4	LCD-RS		LCD-B0	LCD-RS HC164-SER	LCD-SDA	
PA5	LCD-E	(LCD-CE)	LCD-EN	LCD-EN	LCD-SCL	
PA7	Tastensignal					

Tabelle 2.9. Pinbelegungen für verschiedene Displays an ATmega1280/2560

2.7 Chinesische Nachbauten mit Textdisplay

Der Tester wird in China nach meinem Kenntnisstand in zwei Versionen mit Textdisplay nachgebaut. Die erste Variante ist der Nachbau des ersten Entwurfs von Markus F. ohne ISP-Schnittstelle. Der bestückte ATmega8 ist bei dieser Version gesockelt, kann also auch durch einen ATmega168/328 ausgetauscht werden. Für diese Version gelten alle Hinweise des Unterkapitels 2.4. Zusätzliche $100nF$ keramische Kondensatoren sollten in der Nähe des ATmega an die VCC-GND und AVCC-GND Anschlüsse zur besseren Spannungsstabilisierung angebracht werden. Da auf der Platine ein ISP-Stecker fehlt, muß entweder ein ISP-Anschluß nachgerüstet werden oder der Prozessor für die Programmierung ausgebaut werden. Dabei muss beachtet werden, dass bei der Nachrüstung eines Quarzes der ISP-Programmer selbst einen externen Takt zuführen muss, oder der Programmiersockel mit einem Quarz ausgerüstet sein muß.

Die zweite Variante ist weitgehend in SMD-Technik aufgebaut. Auch der ATmega168 oder ATmega328 ist in einem 32TQFP Gehäuse fest verbaut. Dafür ist ein 10-poliger ISP-Stecker für die Programmierung auf der Platine vorgesehen. Ich habe die Version „2.1 2012/11/06“ analysiert. Ein Fehler ist die Bestückung des Bauteils „D1“, welches eigentlich die 2,5V Präzisionsreferenz sein soll. Bestückt ist aber eine Zenerdiode. Dieses Bauteil sollte entfernt werden. Hier kann eine Präzisionsreferenz wie LM4040AIZ2.5 oder LT1004CZ-2.5 angeschlossen werden. Eine fehlende Präzisionsreferenz wird von der Software erkannt, so dass sie nicht unbedingt erforderlich ist. Bei meinem Exemplar war die Software Version 1.02k installiert. Der 10-polige ISP-Stecker war nicht bestückt und ich musste zusätzlich eine Brücke von Pin 10 nach Pin 6 nachlöten. Mein Programmer hat eine GND-Verbindung an Pin 10 erwartet, der Tester hatte aber nur bei Pin 4 und Pin 6 eine GND-Verbindung. Die Beschriftung des ATmega168 war abgeschliffen und es gab keine Dokumentation zum Gerät. Die Sicherheits-Bits des ATmega waren gesetzt, sodass sich das Programm nicht auslesen ließ. Ich konnte aber problemlos die Software Version 1.05k installieren. Die gleiche Softwareversion 1.05k hat bei einem anderen Nutzer mit der China-Version „2.2 2012/11/26“ Probleme gemacht. Die Software 1.05k lief hier erst, als ein weiterer $100nF$ SMD Kondensator zwischen die Pins 18-AVCC und 21-GND ergänzt wurde. Die Software 1.05k benutzt bei Wartezeiten den Schlafzustand des ATmega. Deswegen wechselt der Strombedarf häufiger und der Spannungsregler wird mehr beansprucht. Aufgefallen ist mir weiter, dass die VCC-Spannung mit einem $100nF$ keramischen Kondensator und mit einem $220\mu F$ elektrolytischen Kondensator in der Nähe des 78L05-Spannungsreglers abgeblockt ist. Die 9V Spannungszufuhr ist mit den gleichen Kondensatoren abgeblockt, allerdings am Emitter des PNP-Transistor (parallel zur Batterie), nicht direkt am Reglereingang. Die Leiterbahnen vom ATmega zu den Testports sind teilweise sehr dünn, sodass ich einen Widerstand von etwa $100m\Omega$ pro Signalweg messen konnte. Dies ist wohl mit der Grund dafür, dass zwei mit 0Ω verbundene Pinne einen Widerstandswert von $0,3\Omega$ messen. Bei der ESR-Messung kann dies normalerweise durch den Nullabgleich kompensiert werden.

Bei der Messung von Widerständen unter 10Ω werden die im Selbsttest ermittelten Offsets ab der Softwareversion 1.07k berücksichtigt.

2.8 Chinesische Nachbauten mit graphischem Display

Neuere Nachbauten verwenden ein 128x64 Pixel graphisches Display wie zum Beispiel eine Version von Fish8840. Bei dieser Version wird eine veränderte Schaltung für das Einschalten benutzt. Die Abbildung 2.18 zeigt einen Ausschnitt der Schaltung.

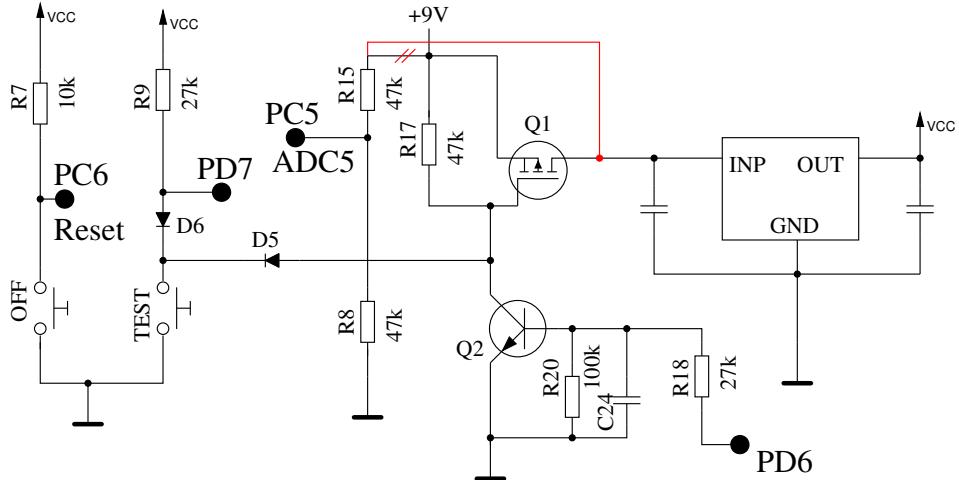


Abbildung 2.18. Ausschnitt aus der Schaltung von Fish8840

Wie an den Widerständen R8 und R15 zu erkennen ist, wird mit 2:1 ein anderes Teilverhältnis für die Batteriespannung wie im Original verwendet. Außerdem ist R15 direkt an die Batterie angeschlossen, was zu einem geringen Stromverbrauch im ausgeschalteten Zustand führt. Hier sollte R15 besser an das Drain von Q1 beziehungsweise an den Eingang des Spannungsreglers angeschlossen werden um den unnötigen Batterieverbrauch zu vermeiden. Eine entsprechende Änderung auf der Platine zeigt die Abbildung 2.19. Eine Leiterbahn ist zwischen R17 und D5 getrennt und mit einem Stück Kupferlackdraht eine neue Verbindung zwischen Q1 und R15 gelötet.

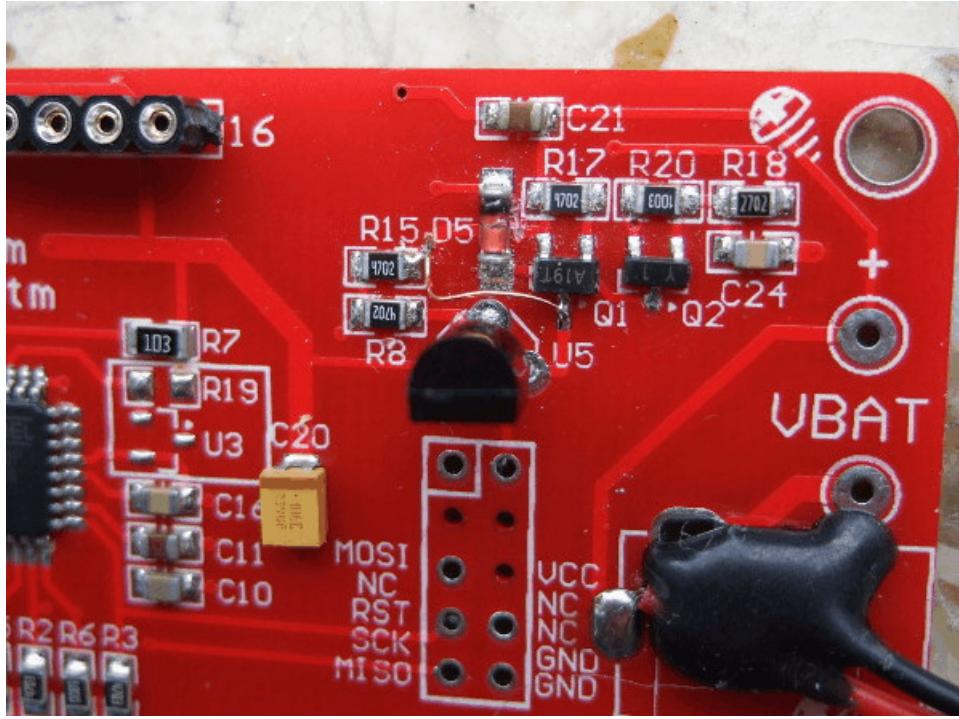


Abbildung 2.19. Foto der geänderten Fish8840 Platine

Das Verhältnis des Spannungsteilers muss bei der Konfiguration in der Makefile auf jeden Fall angegeben werden bevor eine Software aufgespielt wird (z.B. mit BAT_NUMERATOR=66).

Das Display Module des Fish8840 Testers besitzt einen 3.3V Spannungsregler, der die Betriebsspannung des Display-Controllers anpassen soll. Da es über die Datenleitungen des Display-Moduls zu einer Erhöhung der 3.3V Betriebsspannung wegen der 5V Signale vom ATmega kommt, wird eine Adapterschaltung nach Abbildung 2.20 empfohlen. Hier sind in die vier Datenleitungen vier serielle Widerstände mit jeweils $2.7k\Omega$ auf einer kleinen Lochrasterplatine zwischengeschaltet. Längere Gewindegelenke müssen nun verwendet werden, um das Display mit der Adapterplatine auf dem Fish8840 Tester zu befestigen.



(a) Display mit Adapter

(b) Fertig montierter Tester

Abbildung 2.20. Adapter für einen korrekten Display Anschluß

Anstelle dieser Umrüstung kann mit der Makefile Option LCD_SPI_OPEN_COL auch eine spezielle Ausgangsschaltung der 4 SPI Signale des ATmega benutzt werden. Dabei werden die Ausgänge nicht auf VCC Pegel geschaltet, sondern nur die internen „Pull-Up“ Widerstände während der Ausgabe

für den high Pegel freigeschaltet. Wenn die Option PULLUP_DISABLE gesetzt ist, ist ein externer Pull-Up Widerstand für das Reset Signal (PD0) erforderlich. Weil die Datensignale nie direkt auf VCC geschaltet werden, kann die 3.3V Versorgung des LCD-Controllers auch nicht überhöht werden. Die mir vorliegende Platine des Fish8840 Testers hat alle Signale für den Anschluß eines Textdisplays an die LCD-Steckerleiste geführt. Deshalb kann die Platine auch mit einem Textdisplay umgerüstet werden, wenn die Buchsenleiste ergänzt wird und das Potentiometer für die Kontrasteinstellung nachgerüstet wird. Der Versorgungspin 15 für die Hintergrundbeleuchtung ist allerdings direkt mit VCC verbunden. Bei einer Installation eines Textdisplays sollte darauf geachtet werden, daß auf dem Displaymodul ein serieller Widerstand zur LED vorhanden ist. Selbstverständlich muß die Software für eine andere Displayversion angepasst werden. Auch die Erweiterungsmöglichkeiten des Testers sind bei der Fish8840 Platine möglich.

Für die Funktionsfähigkeit kann natürlich keine Gewähr gegeben werden. Leider kann der Originalzustand der chinesischen Software-Variante auch nicht gesichert werden, da die Sicherheits-Bits des ATmega328 gesetzt sind. Es gibt also keinen Weg zurück zur Originalversion der Software.

Eine weitere Version mit graphischem Display ist die WEI_M8 Platine, die in Abbildung 2.21. Dieser Nachbau benutzt als Stromquelle einen LiIon Akkumulator im AA (Mignon) Format, der über eine Micro-USB Schnittstelle geladen werden kann. Ein Betrieb ohne Akku ist über die USB-Schnittstelle ebenfalls möglich.

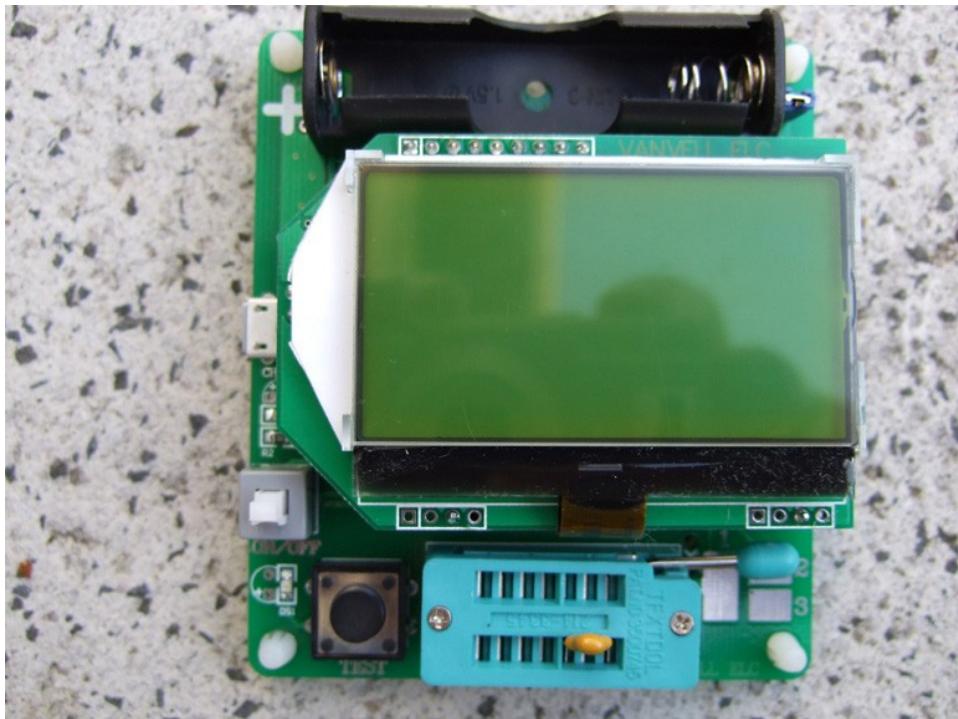
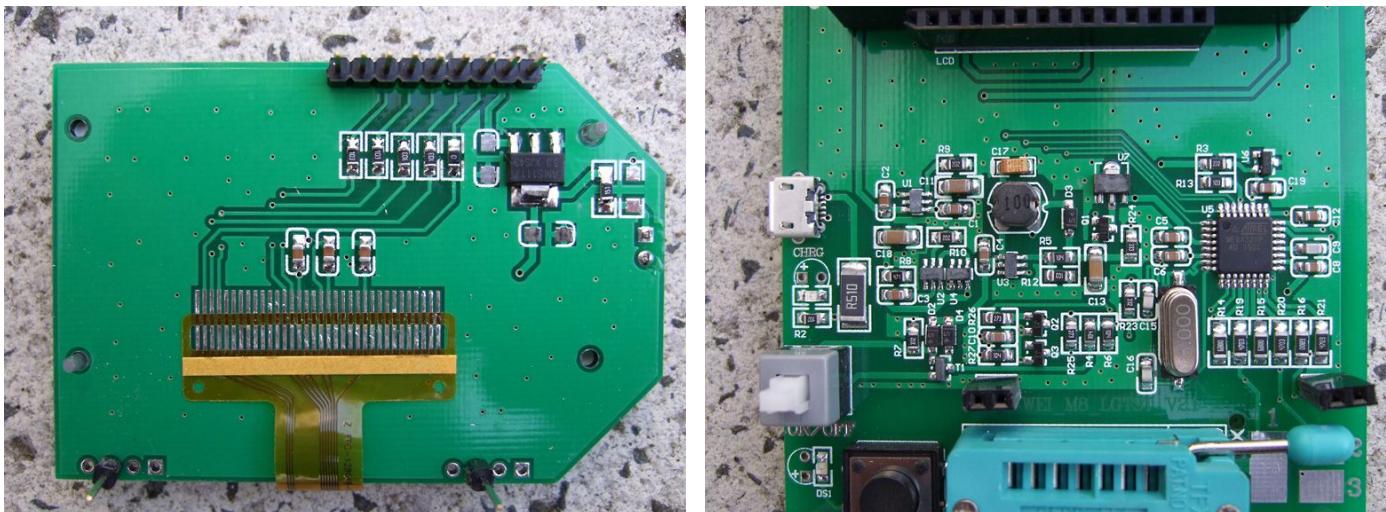


Abbildung 2.21. Foto des chinesischen WEI_M8 Testers

Erfreulich ist, dass für die Signalleitungen des Displays serielle Widerstände auf der Adapterplatine vorgesehen sind, wie in der linken Abbildung von 2.22 zu sehen ist. Damit ist eine Überhöhung der 3.3V Versorgung für den Displaycontroller wegen der 5V Signalpegel vom ATmega nicht zu befürchten.



(a) Adapterplatine für das Display

(b) Grundplatine

Abbildung 2.22. Innenleben des WEI_M8 Testers

Bei der Aufrüstung auf die 1.12k Version der Software ergaben sich einige Schwierigkeiten. Wenn die extended Fuse wie empfohlen auf 0x04 (oder 0xfc) gesetzt wird, kam es bei bestimmten Messungen zu „Brown Out“ Resets des Prozessors. Diese Resets werden durch kurzzeitige Spannungseinbrüche der VCC Versorgung verursacht. Die Platine wurde deshalb mit einem $4.7\mu F$ keramischen Kondensator am Eingang des Spannungsreglers und mit einem $10\mu F$ keramischen Kondensator am Reglerausgang (VCC) zusätzlich abgeblockt. Sowohl vor der Nachrüstung als auch danach wurde bei bipolaren Transistoren ein Kollektor-Reststrom (ICEO oder ICEs) gemessen. Erst der Austausch des LDO-Spannungsreglers (unbekannter Typ) durch einen MCP1702-5002 brachte hier Abhilfe. Die Abbildung 2.23 zeigt die umgerüstete Platine mit den schräg montierten nachgerüsteten Kondensatoren und dem MCP1702 Regler. Wer die Kondensatoren nicht nachrüsten möchte, sollte die extended Fuse auf 0x07 (0xff) setzen, damit ein Betrieb möglich ist. Damit bleiben die Spannungseinbrüche unentdeckt.

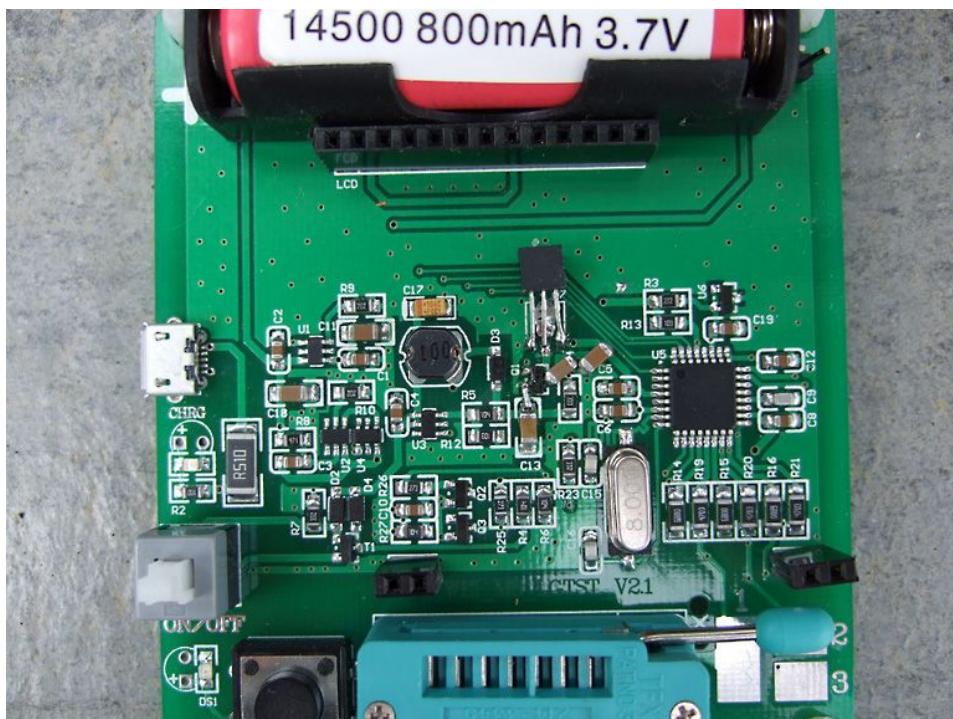


Abbildung 2.23. Foto des umgerüsteten WEI_M8 Testers

Eine weitere Version mit graphischem Display ist der LCD-T4 Tester mit einer gelben Platine.

Für das Aufspielen neuer Software habe ich das Display abgenommen. Auf dem rechten Foto der Abbildung 2.24 ist der Stecker für die ISP-Verbindung rechts oben neben den entsprechenden Bohrungen der Platine in der richtigen Ausrichtung aufgelegt. Für die Programmierung habe ich den Stecker nicht eingelötet, sondern nur eingesteckt und durch seitlichen Druck mit einem Finger für die Dauer der Programmierung fixiert. Dadurch kann der Stecker nach der Programmierung wieder leicht entfernt werden und das Display an die ursprüngliche Stelle montiert werden. Die Software 1.12k konnte übrigens problemlos aufgespielt werden. Auch das Aktivieren der „Brown-Out“ Erkennung des Prozessors mit der extended fuse brachte keine Überraschungen.

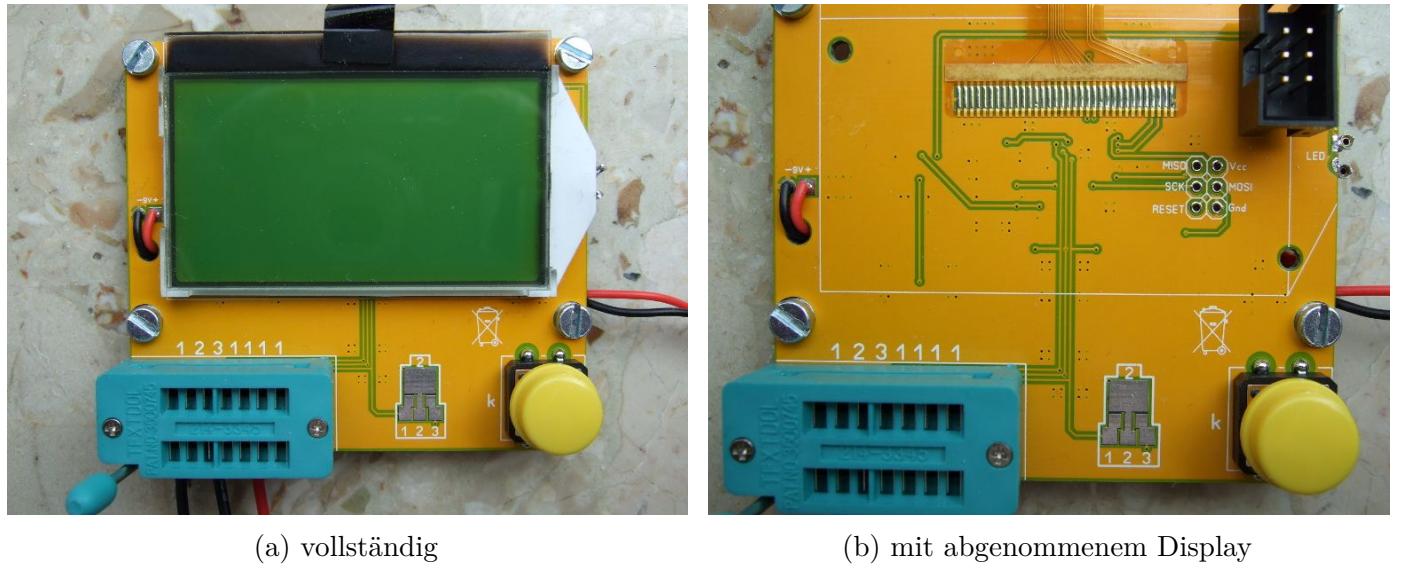


Abbildung 2.24. Frontansicht des T4 Testers

Auf den Fotos der Rückseite in Abbildung 2.25 sieht man die nachgerüsteten 5mm Gewindestifte sowie die nachträglich angelöteten Kabel mit Messclips. Da aber für die Datensignale des graphischen LCD-Controllers auch hier keine Signalanpassung ($5V \rightarrow 3.3V$) erfolgt, ist das Setzen der Option `LCD_SPI_OPEN_COL` ratsam. Da die Nachrüstung eines „pull-up“ Widerstandes für das SPI-Reset Signal hier aber schwierig durchzuführen ist, bleibt nur das Entfernen der Option `DISABLE_PULLUP` übrig. Für nachträgliche Erweiterungen ist diese Platine weniger geeignet, auch ein Austausch des Displays ist hier nur schwer möglich.

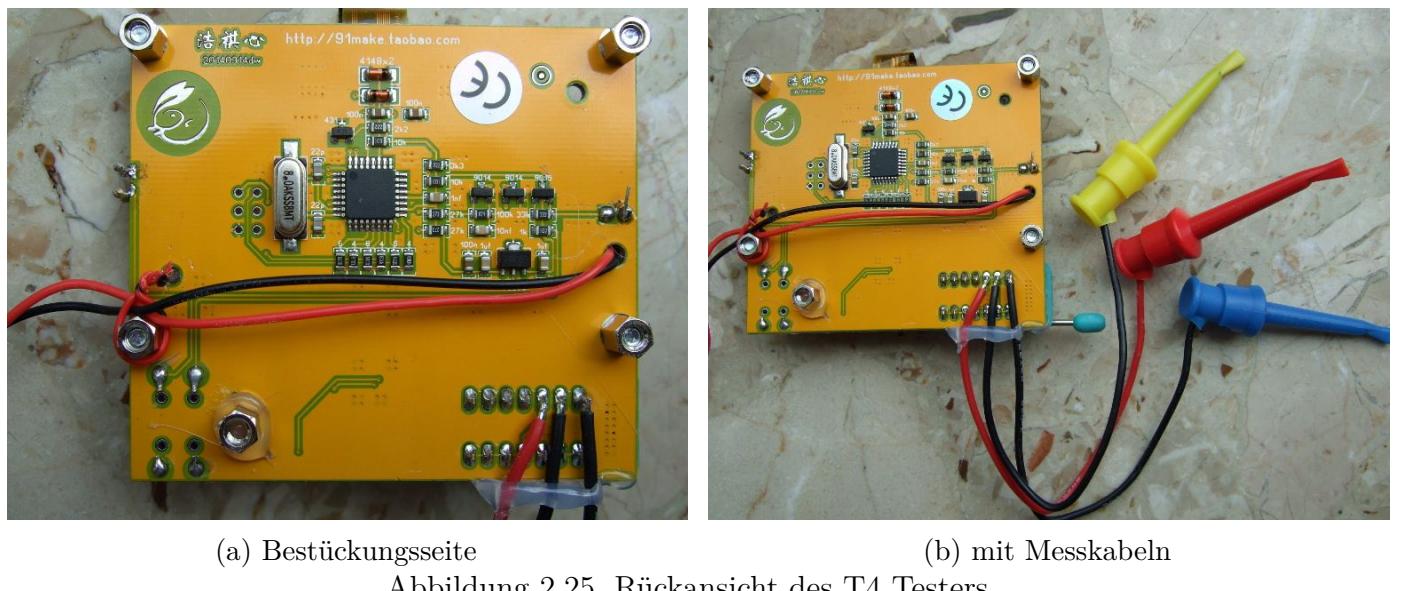


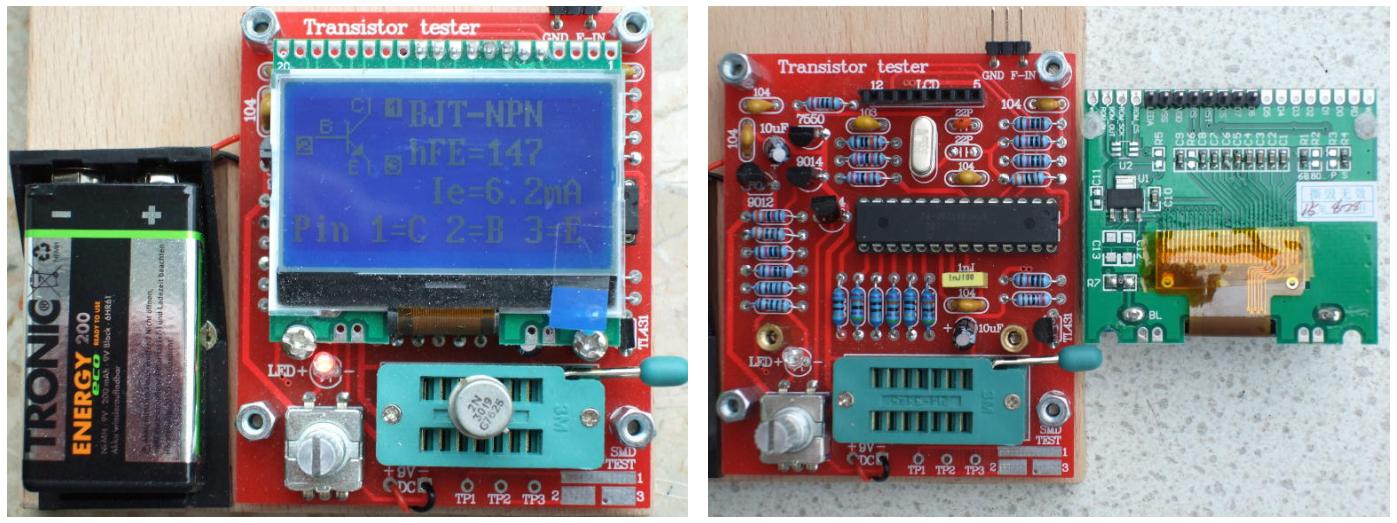
Abbildung 2.25. Rückansicht des T4 Testers

Noch eine chinesische Version mit graphischem Display wird mit der Bezeichnung „GM328“ angeboten.

Bei dieser Version ist das graphische Display an eine 16-pol Buchsenleiste mit einer Tochterplatine angeschlossen. Dabei ist auch der Port PD5 über die Buchsenleiste Pin 6 an das CE (Chip Enable) Signal des graphischen Controllers angeschlossen. Das CE Signal ist aber auf der Tochterplatine auch mit der 0V Power (GND) verbunden. Das führt zu einem Kurzschluß, wenn der ATmega den PD5 Ausgang auf 5V schalten möchte. Neuere Versionen der Software schalten aber auch das CE Signal, auch wenn das Signal für den Betrieb nicht benötigt wird. Deswegen sollte bei dem „GM328“ Tester die Verbindung der Tochterplatine zum Pin 6 der Anschlußleiste unterbrochen werden.

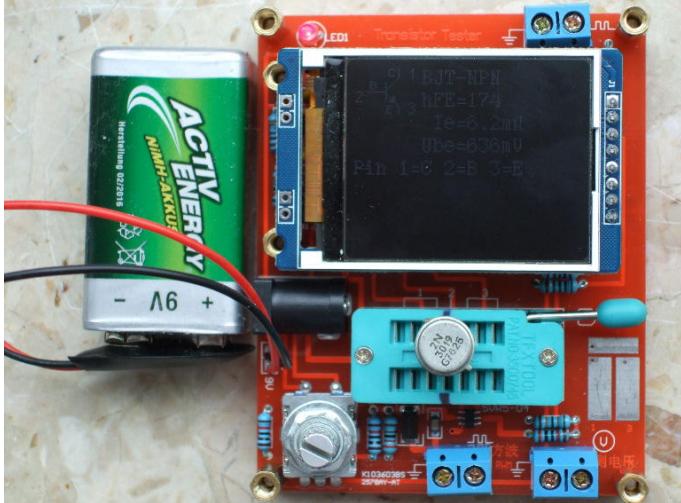
2.9 Chinesische Bausätze mit Grafikdisplay

Bisher sind zwei Bausätze mit grafischen Display und Drehimpulsgeber bekannt. Der zuerst erschienene Bausatz verwendet ein Display mit ST7565 oder kompatiblen Controller (128x64 Pixel). Außer dem Drehimpulsgeber ist auch ein Eingang für die Frequenzmessung vorgesehen. Für die Testports steht ein 14-poliger Textool Sockel, 3 Lötaugen für Kabel sowie ein Testpad für SMD Bauteile zur Verfügung. Die Fotos 2.26 zeigen den zusammengebauten Bausatz. Einer der beiden 22pF Kondensatoren wurde hier durch einen Trimmer auf der Lötseite ersetzt. Mit dem Trimmer lässt sich die Frequenz des Quarzes für genauere Frequenzmessung bzw. Frequenzerzeugung abgleichen.

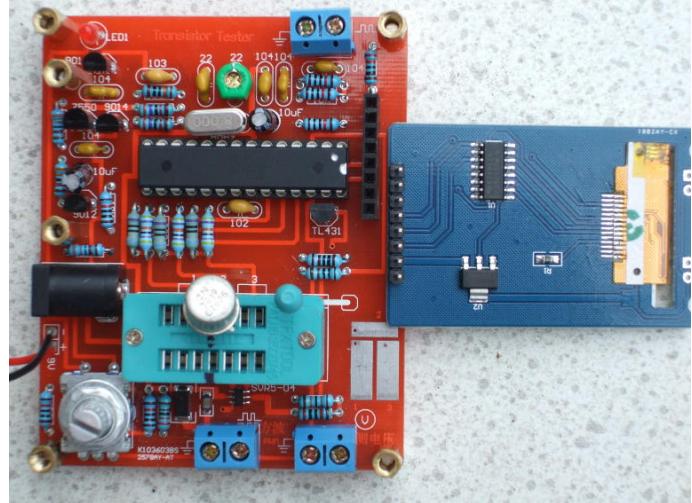


(a) zusammengebaut (b) mit abgenommenem Display
 Abbildung 2.26. Bestückter Bausatz mit 128x64 Pixel Display

Der später erschienene Bausatz verwendet ein Farbdisplay mit ST7735 Controller (128x160 Pixel) und hat zusätzlich zum Frequenzeingang noch einen Eingang für Spannungsmessung und einen Frequenzausgang. Der Frequenzausgang ist aber nicht gepuffert, sondern ist lediglich zum Anschluß TP2 parallel angeschlossen. Bei der Spannungsmessung handelt es sich nur um die Messung einer positiven Gleichspannung bis 50V, ein DC-DC Wandler für die Zenerspannungsmessung wurde nicht vorgesehen. Die Fotos 2.27 zeigen den zusammengebauten Bausatz. Auch hier wurde einer der beiden 22pF Kondensatoren wurde hier durch einen Trimmer (grün) ersetzt.



(a) zusammengebaut



(b) mit abgenommenem Display

Abbildung 2.27. Bestückter Bausatz mit 160x128 Pixel Farbdisplay

Beide Bausätze verwenden einen gesockelte DIP Version des ATmega328P und haben keinen ISP Stecker für das Aufspielen neuer Softwareversionen. Die erste Version verwendet ausschließlich bedrahtete Bauteile für die Hauptplatine. Die Meßwiderstände 680Ω und $470k\Omega$ waren bei meinem Bausatz mit einer 0.1% Toleranz geliefert worden. Selbst ein $220nF$ Kondensator für die Kalibration wurde bei diesem Bausatz mitgeliefert. Der Bausatz mit dem Farbdisplay hat zusätzlich zum Anschluß für die 9V Batterie noch eine Buchse für einen Netzteilanschluß. Die wenigen SMD-Bauteile waren schon auf der Platine bestückt, so daß auch bei diesem Bausatz nur einfache Lötarbeiten durchzuführen sind. Ein kleiner Nachteil der Version mit dem Farbdisplay ist die Geschwindigkeit der Ausgabe. Speziell bei der Menübedienung fällt die deutlich langsamere Ausgabe negativ auf. Andererseits hat das Farbdisplay eine deutlich höhere Pixelzahl und kann damit mehr Text darstellen. Beide Bausätze verwenden einen 3.3V Spannungsregler für die Versorgung der Display Controller auf den vollständig bestückten Display-Platinen. Lediglich die Stifteleiste muß bei der Displayplatine eingelötet werden. Die Farbversion verwendet einen CD4050 Puffer für die Anpassung der Signalpegel. Auf der Display-Platine der ST7565 Variante ist keine Anpassung der Signalpegel erkennbar. Wahrscheinlich toleriert die gewählte Controller-Variante die 5V Signalpegel des ATmega328. Es ist jedenfalls keine Schutzdiode gegen die 3.3V Versorgung bei diesem Controller feststellbar.

2.10 Und noch ein Clone von Hiland M644

Dieses Nachbau basiert auf dem Schaltplan von Nick L. aus der Ukraine siehe Abbildung 2.16 auf der Seite 26.

Bedient wird der Tester mit einem Knopf, der sowohl **Taster** als auch **Drehgeber** ist. Er bietet folgende Zusätze:

- die Frequenzmessung
- den f-Generator
- 10-bit PWM
- Impulsdrehgeber
- die Quarz-Messung
- Spannung und Zenerdioden Bestimmung (bis fast 50V).

Die Platine ist mit einem 8MHz Quarz bestückt. Besser wäre wohl ein 16MHz Quarz, wobei die Nachrüstung eines Trimmers (genauere Frequenz) schwierig ist. Die Kontakte für die ISP-Schnittstelle sind auf eine 6-pol Lochreihe unterhalb dem steckbaren Display-Modul herausgeführt die folgend belegt ist:

von links nach rechts: 1 -Reset; 2 -SCK; 3 -MISO; 4 -MOSI; 5 -+5V; 6 -GND.

Um den Tester, aktualisieren zu können, benötigt man ein angepasstes Kabel, der man relativ einfach selbst erstellen kann. Bei der Auslieferung ist der Nullkraft-Testsockel über Buchsenleisten mit der Platine verbunden.

Bei dem unten abgebildeten Tester wurde der Sockel direkt angelötet und die hiermit gesparte Buchsenleiste auf ein vorhandenes Flachbandkabel mit Pfostenstecker angelötet und mit Schrumpfschlauch fixiert.

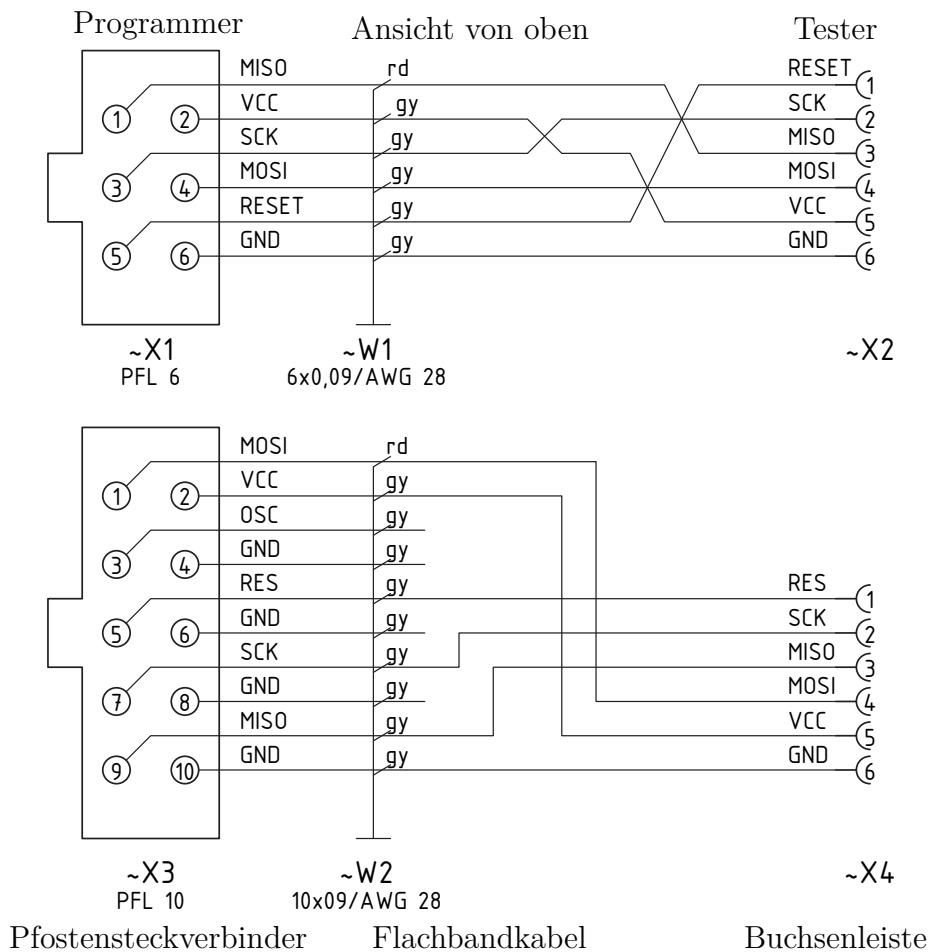
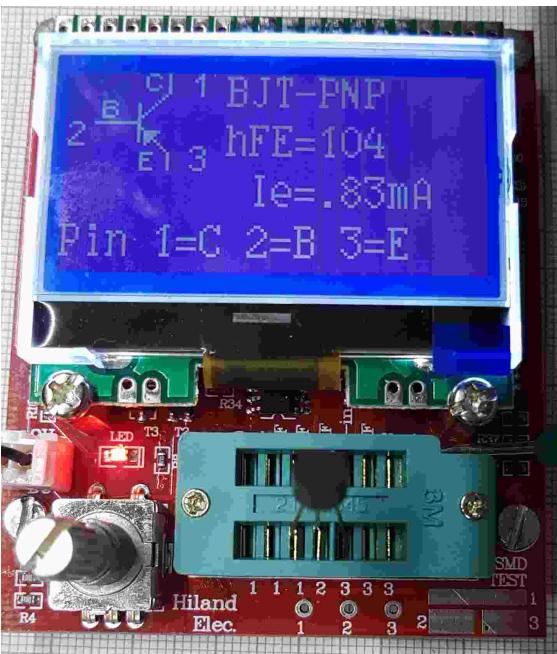
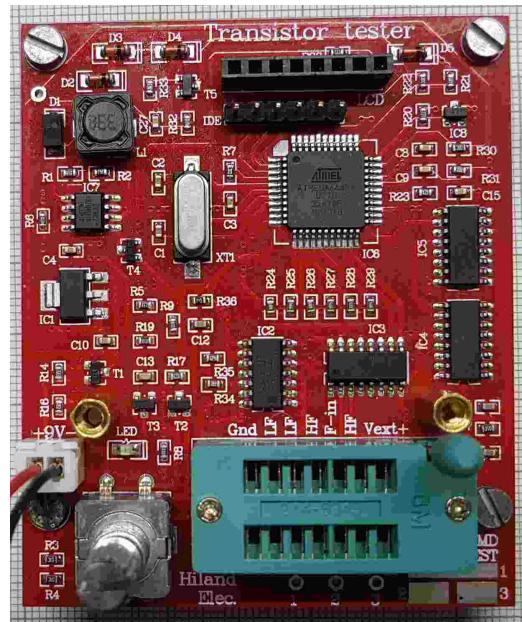


Abbildung 2.28. 6 und 10 Pol Kabel für die Programmierung



(a) unten TP1 bis TP3 für Bauteilebestimmung



(b) oben Pins für Auswahlmenü und die IDE

Abbildung 2.29. Hiland Tester mit Testsockel und 128x64 Pixel Display

Die **Testports TP1, TP2 und TP3** werden für die automatische Bauteilerkennung benutzt und sind auf der Platine mit den Ziffern **1 1 1 2 3 3,3** gekennzeichnet.

Die gleiche Bezeichnung befindet sich auf dem SMD Testfeld und es besteht auch eine Möglichkeit, eigene Test Kabel anzulöten. Der **Testport TP2** wird auch für die Ausgabe der Sonderfunktion „f-Generator“ benutzt.

Die mit **LF** bezeichneten Pins sind für die Messung von Quarzen mit niedriger Resonanzfrequenz vorgesehen, und die mit **HF** bezeichneten Pins sind für Quarze mit hoher Resonanzfrequenz bestimmt. Der Pin **F-in** wird zusammen mit **Gnd** für die Sonderfunktion Frequenz benutzt. Und der Pin **Vext+** wird genauso mit **Gnd** für die Spannungsmessung **und** die Zenerdiodenmessung benutzt.

2.11 Fehlersuche

Bei den meisten Problemen werden Sie den Text auf dem LCD-Display vermissen. Zuerst sollten Sie prüfen, ob die LED auf der Platine schwach leuchtet, wenn Sie den Start-Taster loslassen.

Gerät schaltet nicht ein Wenn die LED nicht leuchtet, aber die VCC-Spannung den richtigen

Wert hat, wenn man den Start-Taster gedrückt hält, schaltet der Mikrocontroller nicht richtig ein.

Der Mikrocontroller sollte die Spannung behalten indem der Ausgang PD6 auf 5V geschaltet wird, was üblicherweise als eine der ersten Aktionen getan wird. Wenn man die Start-Taste gedrückt hält, bleibt die Spannung ohnehin eingeschaltet. So können Sie mit gedrücktem Taster den Wert der VCC-Spannung und zusätzlich den Spannungswert am Ausgang PD6 prüfen.

Wenn die VCC-Spannung den richtigen Wert (5V) hat, aber die Spannung am Ausgang PD6 unter 4V ist, startet der Mikrocontroller nicht richtig. Für diesen Fall sollten Sie prüfen, ob die Programmdaten für den Flashspeicher für den richtigen Prozessor-Typ ist und ob der Prozessor richtig konfiguriert ist (fuses). Wenn der ATmega den Ausgang PD6 auf 5V schaltet und die Betriebsspannung trotzdem nicht eingeschaltet bleibt, wenn man den Start-Taster loslässt, ist der Grund schwieriger zu finden. Zuerst kann man die LED kurzschließen und es noch einmal versuchen. Wenn der Tester jetzt startet, ist die LED möglicherweise falsch herum eingebaut. Wenn das nicht die Ursache ist, könnte der Grund ein unzureichender Stromverstärkungs-Faktor des Transistors T3 (BC557C) sein. Der Strom in die Basis von T3 ist niedriger, wenn der Mikrocontroller mit der LED einschaltet wie im „Taster gedrückt“-Zustand.

Nichts ist lesbar auf der LCD-Anzeige Prüfen Sie die Spannung am Kontrast-Pin der LCD-Anzeige (Pin 3). Stellen Sie mit dem Trimmer den Wert auf einen im Datenblatt angegebenen Wert und optimieren Sie durch Sichtkontrolle. Wenn Sie ein Hochtemperatur-Display haben, brauchen Sie eine negative Kontrast-Spannung für den Betrieb. In diesem Fall kann man den ICL 7660-Baustein zum Erzeugen der negativen Spannung aus der positiven 5V verwenden.

Der Tester kann für viele verschiedene Controller mit unterschiedlichen Anschlußarten konfiguriert werden. Prüfen sie auf jeden Fall, ob die Software zu ihrem Display paßt. Wenn keine Anzeige auf dem LCD erkannt wird und wenn die Hintergrundbeleuchtung an ist, sollten Sie die Spannungsversorgung trennen und alle vier Datenverbindungen sowie die beiden Steuersignale überprüfen. Wenn alle Verbindungen in Ordnung sind, sehe ich als Ursache nur noch die Möglichkeit einer falschen Zeitabfolge der Steuersignale. Die Ursache hierfür kann sein, dass der LCD-Controller langsamer ist als es die Software des ATmega erwartet. Es könnte auch sein, dass der ATmega auf der falschen Taktrate läuft. Bitte überprüfen Sie für welche Taktrate die Software übersetzt ist und ob die fuses des ATmega für diese Geschwindigkeit richtig gesetzt sind. Sie finden die eingestellte Taktrate in der betreffenden Makefile. Wenn der Tester ohne die Abschalt-Elektronik aufgebaut ist, kann man mit einer an die Test Pins angeschlossenen LED testen, ob das Programm arbeitet. Wenn die LED flackert, läuft das Programm. Der Fehler muss in diesen Fall an dem Anschluss des LCD's liegen. Bei einigen graphischen Displays kann der Kontrast über eine Menüfunktion verstellt werden. Wenn man hier versehentlich den Kontrast zu weit verstellt hat, ist auf dem Display nicht mehr zu erkennen, man kann also auch nicht mehr weiter bedienen. Hier können Sie nur versuchen, ob von der Seite (schräg) doch noch etwas zu erkennen ist und über das Menü den Kontrast zurückstellen. Wenn dan nicht geht, können sie das EEPROM des ATmega mit einem ISP programmer neu beschreiben und damit den Kontrastwert zurückstellen.

Einiges, aber nicht alles ist auf der LCD-Anzeige lesbar Überprüfen Sie ob die .eep-Daten in den EEPROM des ATmega geladen wurden. Wenn alle Programmdaten richtig geladen wurden, sollten Sie die Taktrate ihrer Programmdaten (Makefile) und die ATmega-Einstellungen prüfen (fuses).

Messung ist zu langsam und Kapazitäten werden um Faktor 8 zu klein gemessen. Sie betreiben die Software, die für 8MHz übersetzt wurde mit einer Taktrate von 1MHz. Bitte konfigurieren Sie den ATmega mit den fuses richtig.

Die Messung ergibt seltsame Ergebnisse Überprüfen Sie ob der ISP-Programmierstecker noch verbunden ist. Der ISP-Stecker sollte nicht während einer Messung eingesteckt bleiben. Sehr oft ist der Grund für falsche Messergebnisse, dass die Software mit der AUTOSCALE_ADC Option und der NO_REF_CAP Option übersetzt wurde, aber der Kondensator am AREF-Pin hat immer noch einen Wert von $100nF$. Falsche Bestückung von Bauteilen können auch eine Ursache für Messfehler sein oder zurückgebliebene Flussmittelreste können die Messung stören. Bitte prüfen Sie nach Möglichkeit mit der Selbsttest-Funktion der TransistorTester-Software. Zu den Einzelheiten schauen Sie in das Selbsttest-Kapitel 5.5.

Andernfalls prüfen Sie Ihre Platine visuell und prüfen Sie die Widerstandswerte mit einem Ohmmeter. Sie können die Pins des ATmega für diese Prüfung benutzen, zum Beispiel können Sie den Widerstand R1 zwischen Pin 23 und Pin 14 messen. Schauen Sie in das Schaltbild 2.1 für die Einzelheiten. Man braucht den Mikrocontroller nicht zu entfernen, nur die Stromversorgung sollte vorher getrennt werden.

Der Tester schaltet nach 2 Sekunden Anzeigezeit aus Dies ist dann der Fall, wenn der Pull-Up-Widerstand am PD7-Eingang fehlt oder der Taster dauernd gedrückt wird. Die Software

schaltet die internen Pull-Up-Widerstände ab, um eine Beeinflussung der Messergebnisse auszuschließen. Deswegen ist ein externer Widerstand erforderlich.

Der Tester zeigt immer nur Vext=xx.xV in Zeile 2 an Dies ist dann der Fall, wenn der Pull-Up-Widerstand am PD7 Eingang fehlt oder der Taster dauernd gedrückt wird. Die Software ist außerdem ohne seriellen Ausgang (ohne Option WITH_UART) und ohne interne Pull-Up Widerstände (mit Option PULLUP_DISABLE) konfiguriert. Installieren Sie einen Pull-Up Widerstand an PD7.

Kapitel 3

Bedienungshinweise

3.1 Der Messbetrieb

Die Bedienung des Transistortesters ist einfach. Trotzdem sind einige Hinweise erforderlich. Meistens sind an die drei Testports über Stecker-Leitungen mit Krokodilklemmen oder anderen Klemmen angeschlossen. Es können auch Fassungen für Transistoren angeschlossen sein. In jedem Fall können Sie Bauteile mit drei Anschlüssen mit den drei Testports in beliebiger Reihenfolge verbinden. Bei zweipoligen Bauteilen können Sie die beiden Anschlüsse mit beliebigen Testports verbinden. Normalerweise spielt die Polarität keine Rolle, auch Elektrolytkondensatoren können beliebig angeschlossen werden. Die Messung der Kapazität wird aber so durchgeführt, dass der Minuspol am Testport mit der kleineren Nummer liegt. Da die Messspannung aber zwischen 0,3V und maximal 1,3V liegt, spielt auch hier die Polarität keine wichtige Rolle. Wenn das Bauteil angeschlossen ist, sollte es während der Messung nicht berührt werden. Legen Sie es auf einen isolierenden Untergrund ab, wenn es nicht in einem Sockel steckt. Berühren Sie auch nicht die Isolation der Messkabel, das Messergebnis kann beeinflusst werden. Dann sollte der Starttaster gedrückt werden. Nach einer Startmeldung erscheint nach circa zwei Sekunden das Messergebnis. Bei einer Kondensatormessung kann es abhängig von der Kapazität auch deutlich länger dauern.

Was dann weiter geschieht, hängt von der Softwarekonfiguration des Testers ab.

Einzelmessung Wenn der Tester für Einzelmessung konfiguriert ist (POWER_OFF-Option), schaltet der Tester nach einer Anzeigezeit von 28 Sekunden (konfigurierbar) wieder automatisch aus, um die Batterie zu schonen. Während der Anzeigezeit kann aber auch vorzeitig eine neue Messung gestartet werden. Nach der Abschaltung kann natürlich auch wieder eine neue Messung gestartet werden, entweder mit dem gleichen Bauteil oder mit einem anderen Bauteil. Wenn die Elektronik zum Abschalten fehlt, wird das letzte Messergebnis weiter angezeigt.

Dauermessung Einen Sonderfall stellt die Konfiguration ohne die automatische Abschaltfunktion dar. Hierfür wird die POWER_OFF-Option in der Makefile nicht gesetzt. Diese Konfiguration wird normalerweise nur ohne die Transistoren für die Abschaltung benutzt. Es wird stattdessen ein externer Ein-/Aus-Schalter benötigt. Hierbei wiederholt der Tester die Messungen solange, bis ausgeschaltet wird.

Serienmessung In diesem Konfigurationsfall wird der Tester nicht nach einer Messung sondern erst nach einer konfigurierbaren Zahl von Messungen abgeschaltet. Hierfür wird der POWER_OFF-Option in der Makefile eine Wiederholzahl (z.B. 5) zugewiesen. Im Standardfall wird der Tester nach fünf Messungen ohne erkanntes Bauteil abgeschaltet. Wird ein angeschlossenes Bauteil erkannt, wird erst bei der doppelten Anzahl, also zehn Messungen abgeschaltet. Eine einzige Messung mit nicht erkanntem Bauteil setzt die Zählung für erkannte Bauteile auf Null zurück. Ebenso setzt eine einzige Messung mit erkannten Bauteil die Zählung für die nicht erkannten

Bauteile auf Null zurück. Dies hat zur Folge, dass auch ohne Betätigung des Starttasters immer weiter gemessen werden kann, wenn Bauteile regelmässig gewechselt werden. Ein Bauteilwechsel führt in der Regel durch die zwischenzeitlich leeren Klemmen zu einer Messung ohne erkanntes Bauteil.

Eine Besonderheit gibt es in diesem Betriebsmodus für die Anzeigezeit. Wenn beim Einschalten der Starttaster nur kurz gedrückt wurde, beträgt die Anzeigezeit der Messergebnisse nur 5 Sekunden. Wenn der Starttaster bis zum Erscheinen der ersten Meldung festgehalten wurde, beträgt die Anzeigezeit wie bei der Einzelmessung 28 Sekunden. Ein vorzeitiger neuer Messbeginn ist aber während der Anzeigezeit durch erneutes Drücken des Starttasters möglich.

3.2 Optionale Menüfunktionen für den ATmega328

Wenn die Menüfunktion eingeschaltet ist, startet der Tester nach einem längeren Tastendruck ($> 0.5s$) ein Auswahlmenü für zusätzliche Funktionen. Diese Funktion ist auch für andere Prozessoren mit mindestens 32K Flash Speicher verfügbar. Die wählbaren Funktionen erscheinen in Zeile 2 eines zweizeiligen Displays oder werden als gekennzeichnete Funktion in Zeile 3 eines vierzeiligen Displays gezeigt. Dabei wird die vorige und nächste Funktion in Zeile 2 und 4 angezeigt. Nach einer längeren Wartezeit ohne jegliche Bedienung kehrt das Programm zu der normalen Transistor-Tester-Funktion zurück. Durch kurzen Tastendruck kann zur nächsten Auswahl gewechselt werden. Mit einem längeren Tastendruck startet die angezeigte Zusatzfunktion. Nach Anzeige der letzten Funktion „Schalte aus“ wird wieder die erste Funktion angezeigt.

Wenn der Tester mit einem Drehimpulsgeber ausgerüstet ist, kann die Menü-Auswahl auch mit einem schnellen Drehen des Encoders während der Anzeige einer vorausgegangenen Messung aufgerufen werden. Die Menüfunktionen können mit einem langsamen Drehen des Encoders in beliebiger Richtung ausgewählt werden. Die ausgewählte Menüfunktion kann aber nur mit einem längeren Tastendruck gestartet werden. Innerhalb einer Menüfunktion können Parameter durch eine langsame Drehung des Encoders ausgewählt werden. Eine schnelle Drehung des Encoders kehrt zur Menü-Auswahl zurück.

Frequenz Die Zusatzfunktion „Frequenz“ (Frequenzmessung) benutzt als Eingang den PD4-Pin des ATmega, der auch an das LCD angeschlossen ist. Es wird immer zunächst die Frequenz gemessen, bei Frequenzen unter $25kHz$ wird auch die mittlere Periode des Eingangssignals bestimmt und daraus die Frequenz mit einer Auflösung von bis zu $0,001Hz$ berechnet. Bei gesetzter POWER_OFF-Option in der Makefile wird die Dauer der Frequenzmessung auf 8 Minuten beschränkt. Die Frequenzmessung wird durch Tastendruck beendet und in das Auswahlmenü zurückgekehrt.

f-Generator Bei der Zusatzfunktion „f-Generator“ (Frequenz-Generator) können die Frequenzen zwischen 1Hz und 2MHz gewählt werden. Die Einstellung der Frequenz kann jeweils nur für die höchste dargestellte Stelle (Ziffernposition) verändert werden. Für die Stellen 1Hz bis 10kHz sind jeweils die Ziffern 0-9 wählbar. Bei der 100kHz Stelle ist 0-20 wählbar. In Spalte 1 der Frequenzzeile wird durch ein $>$ oder $<$ Symbol angezeigt, ob durch einen längeren Tastendruck ($> 0.8s$) zur höheren oder niedrigen Stelle geschaltet wird. Zur niedrigeren Stelle ($<$) kann nur geschaltet werden, wenn die augenblickliche Stelle auf 0 gestellt wurde und wenn nicht die Stelle mit 1Hz Schritten gewählt wurde. Bei der gewählten 100kHz Stelle ist das $>$ Symbol durch ein R Zeichen ersetzt. Der längere Tastendruck bewirkt dann eine Rücksetzung der Frequenz auf den Startwert 1Hz. Bei gesetzter POWER_OFF-Option in der Makefile muss für den Frequenzwechsel die Taste länger gedrückt werden, da durch einen kurzen Tastendruck ($< 0,2s$) nur die Zeitüberwachung von 4 Minuten zurückgesetzt wird. Die abgelaufene Zeit wird in Zeile 1 durch einen Punkt für jede abgelaufene 30 Sekunden angezeigt. Durch einen regelmässigen

kurzen Tastendruck kann die vorzeitige Abschaltung der Frequenzerzeugung verhindert werden. Ein längerer Tastendruck (> 2s) kehrt wieder zur Auswahl der Funktionen zurück.

10-bit PWM Bei der Zusatzfunktion „10-bit PWM“ (Pulsweitenmodulation) wird eine feste Frequenz mit einstellbarer Pulsweite an Pin TP2 erzeugt. Mit einem kurzen Tastendruck (< 0,5s) wird die Pulsweite um 1% erhöht, mit einem längeren Tastendruck um 10%. Bei Überschreiten von 99% werden 100% vom erhöhten Wert abgezogen. Bei gesetzter POWER_OFF-Option in der Makefile wird die Frequenzerzeugung nach 8 Minuten ohne Bedienung beendet. Durch sehr langen Tastendruck (> 1,3s) kann die Frequenzerzeugung auch beendet werden.

C+ESR@TP1:3 Bei der Zusatzfunktion „C+ESR@TP1:3“ wird eine separate Kondensatormessung mit ESR-Messung an TP1 und TP3 gestartet. Messbar sind Kondensatoren mit mehr als $2\mu F$ bis zu $50mF$. Wegen der geringen Messspannung von etwa 300mV sollte in vielen Fällen die Messung in der Schaltung ohne vorherigen Ausbau möglich sein. Bei gesetzter POWER_OFF-Option in der Makefile ist die Anzahl der Messungen auf 250 beschränkt, kann aber sofort wieder gestartet werden. Die Mess-Serie kann durch einen längeren Tastendruck beendet werden.

Widerstandsmeßgerät Mit dem 1—■—3 Symbol wird der Tester in ein Ohmmeter an TP1 und TP3 verwandelt. Diese Betriebsart wird durch ein [R] in der rechten Ecke der ersten Displayzeile angezeigt. Weil bei dieser Betriebsart die ESR-Meßfunktion nicht benutzt wird, beträgt auch für Widerstände unter 10Ω die Auflösung nur 0.1Ω . Wenn die Ohmmeter Funktion mit der Induktivitätsmessung konfiguriert wurde, erscheint hier ein 1—■—■■■—3 Symbol. Dann beinhaltet die Ohmmeter Funktion die Messung von Induktivitäten für Widerstände unter 2100Ω . In der rechten Ecke der ersten Zeile des Displays wird dann ein [RL] angezeigt. Für Widerstände unter 10Ω wird dann auch die ESR-Meßmethode benutzt, wenn keine Induktivität festgestellt wurde. Damit erhöht sich die Auflösung für Widerstände unter 10Ω auf 0.01Ω . In dieser Betriebsart werden die Meßwerte fortlaufend ermittelt. Mit einem Tastendruck beendet der Tester diese Betriebsart und kehrt wieder zum Menü zurück. Die gleiche Betriebsart wird auch automatisch gestartet, wenn zwischen TP1 und TP3 ein einzelner Widerstand angeschlossen wurde und der Start-Taster gedrückt wurde. In diesen Fall kehrt der Tester mit einem Tastendruck wieder zu der normalen Teseterfunktion zurück.

Kondensatormeßgerät Mit dem 1—■—3 Symbol wird der Tester in ein reines Kondensator-Meßgerät an TP1 und TP3 verwandelt. Die Betriebsart wird durch ein [C] in der rechten Ecke der ersten Zeile des Displays angezeigt. Bei dieser Betriebsart können Kondensatoren ab $1pF$ bis zu $100mF$ gemessen werden. In dieser Betriebsart werden die Meßwerte fortlaufend ermittelt. Mit einem Tastendruck wird dieser Sonderbetrieb beendet und der Tester kehrt wieder zum Menü zurück. Auch hier wird wie bei der Widerstands-Meßfunktion in den Betriebs-Modus automatisch gewechselt, wenn zwischen TP1 und TP3 ein Kondensator mit der normalen Teseterfunktion gemessen wurde. Der Tester kehrt nach dem automatischen Start wieder zu der normalen Teseterfunktion zurück, wenn der Taster gedrückt wird.

Impulsdrehgeber Mit der Zusatzfunktion „Impulsdrehgeber“ kann ein Drehgeber untersucht werden. Die drei Kontakte des Impulsdrehgebers müssen vor dem Start der Zusatzfunktion beliebig an die drei Testpins des Transistorstesters angeschlossen werden. Nach dem Start der Funktion muss der Drehknopf nicht zu schnell gedreht werden. Wenn der Test erfolgreich abgeschlossen ist, wird die Pinbelegung der Kontakte symbolisch in Zeile 2 dargestellt. Dabei wird der gemeinsame Anschluss herausgefunden und für etwa zwei Sekunden angezeigt, ob an den Raststellung beide Kontakte offen (‘o’) oder beide Kontakte geschlossen (‘C’) sind. Ein Impulsdrehgeber mit offenen Kontakten an den Raststellungen wird so mit der Zeile 2 „1/-2/-3 o“ zwei Sekunden lang angezeigt. Natürlich wird die richtige Pinnummer des gemeinsamen Kontaktes in der Mitte

anstelle der '2' angezeigt. Wenn auch die geschlossene Schalterstellung an den Raststellungen vorkommt, wird außerdem in Zeile 2 „1—2—3 C“ zwei Sekunden lang angezeigt. Mir ist kein Impulsdrehgeber bekannt, der immer nur geschlossene Kontakte an jeder Raststellung hat. Die Stellungen der Kontakte zwischen den Raststellungen werden nur kurz (< 0,5s) ohne die Kennbuchstaben 'o' oder 'C' in Zeile 2 angezeigt. Wenn der Impulsdrehgeber für die Bedienung des Testers eingesetzt werden soll, muß die Makefile Option WITH_ROTARY_SWITCH=2 für Drehgeber mit nur den offenen Kontakten ('o') und die Option WITH_ROTARY_SWITCH=1 für Drehgeber mit offenen ('o') und geschlossenen ('C') Kontakten an den Raststellungen.

C(μ F)-Korrektur Mit dieser Menüfunktion kann ein Korrekturwert für die Messung grösserer Kapazitätswerte geändert werden. Die Funktion ist die gleiche, die auch mit der Makefile Option C_H_KORR voreingestellt werden kann. Werte über Null reduzieren den Ausgabewert der Kapazität um diesen Prozent Wert, Werte unter Null würden den Ausgabewert anwachsen lassen. Ein kurzer Tastendruck verringert den Korrekturwert um 0.1%, ein längerer Tastendruck vergrößert den Korrekturwert um 0.1%. Ein sehr langer Tastendruck speichert den Wert. Es ist eine Eigenschaft des Meßverfahrens, daß Kondensatoren mit geringer Güte wie Elektrolyt-Kondensatoren mit zu großer Kapazität gemessen werden. Erkennbar ist die Güte über den Parameter Vloss. Gute Kondensatoren haben kein Vloss oder nur 0.1%. Zum Abgleich dieses Parameters sollten also nur Kondensatoren mit über $50\mu F$ hoher Güte verwendet werden. Übrigens halte ich die Bestimmung des exakten Kapazitätswertes von Elektrolyt-Kondensatoren für überflüssig, da die Kapazität sowohl von der Temperatur als auch der DC-Spannung abhängt.

Selbsttest Mit der Zusatzfunktion „Selbsttest“ wird ein vollständiger Selbsttest mit Kalibration durchgeführt. Dabei werden sowohl die Testfunktionen T1 bis T7 (wenn nicht verhindert mit der Option NO_TEST_T1_T7) als auch die Kalibration mit dem externen Kondensator jedes Mal durchgeführt.

Spannung Die Zusatzfunktion „Spannung“ (Spannungsmessung) ist nur möglich, wenn die serielle Ausgabe deaktiviert wurde oder der ATmega mindestens 32 Pinne hat (PLCC) und einer der zusätzlichen Pinne ADC6 oder ADC7 für die Messung benutzt wird. Da am Port PC3 (oder ADC6/7) ein 10:1-Spannungsteiler vorgesehen ist, können Spannungen bis 50V gemessen werden. Ein installierter DC-DC-Wandler für die Zenerdiode-Messung wird durch Tastendruck eingeschaltet. So können auch angeschlossene Zenerdioden gemessen werden. Bei gesetzter POWER_OFF-Option in der Makefile und ohne Bedienung wird die Messung nach 4 Minuten beendet. Die Messung kann aber durch einen besonders langen Tastendruck (> 4 Sekunden) vorher beendet werden.

Kontrast Diese Funktion steht für Controller mit Kontrasteinstellung per Software zur Verfügung. Der Einstellwert kann mit einem sehr kurzen Tastendruck oder einer Linksdrehung des Impulsdrehgebers verringert werden. Ein längerer Tastendruck oder eine Rechtsdrehung des Impulsdrehgebers vergrößert den Einstellwert. Die Funktion wird beendet und der eingestellte Wert wird permanent in den EEPROM geschrieben, wenn die Taste noch länger gedrückt wird.

BackColor Für Farbdisplays kann dieser Menüpunkt mit der Makefile Option LCD_CHANGE_COLOR eingeschaltet werden, um die Hintergrundfarbe einzustellen zu können. Die Impulsdrehgeber-Erweiterung muß dafür installiert sein. Sie können eine der drei Farben rot, grün und blau mit einem längeren Tastendruck wählen. Die Intensität der gewählten Farbe, die mit einem > Zeichen in Spalte 1 gekennzeichnet wird, kann durch Drehen des Impulsdrehgebers verändert werden.

FrontColor Für Farbdisplays kann dieser Menüpunkt mit der Makefile Option LCD_CHANGE_CO-

LOR eingeschaltet werden, um die Vordergrundfarbe einzustellen zu können. Die Impulsdrehgeber-Erweiterung muß dafür installiert sein. Sie können eine der drei Farben rot, grün und blau mit einem längeren Tastendruck wählen. Die Intensität der gewählten Farbe, die mit einem > Zeichen in Spalte 1 gekennzeichnet wird, kann durch Drehen des Impulsdrehgebers verändert werden.

Zeige Daten Die Funktion „Zeige Daten“ zeigt neben der Software-Versionsnummer die Daten des Abgleichs an. Dies sind die Nullwiderstände R0 von Pin 1:3, 2:3 und 1:2. Außerdem werden die Ausgangswiderstände der Ports zur 5V-Seite (RiHi) und zur 0V Seite (RiLo) angezeigt. Die Nullkapazitätswerte (C0) werden ebenfalls in allen Pinkombinationen angezeigt (1:3, 2:3, 1:2 und 3:1, 3:2 2:1). Danach werden auch die Spannungskorrekturen für die Komparatormspannung (REF_C) und für die Referenzspannung (REF_R) angezeigt. Bei Graphikdisplays werden danach noch die verwendeten Symbole für die Bauteile und der Fontsatz angezeigt. Jede Seite wird 15 Sekunden angezeigt. Es kann aber auch durch Tastendruck oder einer Rechtsdrehung des Impulsdrehgebers zur nächsten Seite geblättert werden. Mit einer Linksdrehung des Impulsdrehgebers kann die Ausgabe wiederholt werden oder zur vorigen Seite zurückgeblättert werden.

Schalte aus Mit der Zusatzfunktion „Schalte aus“ kann der Transistortester abgeschaltet werden.

Transistor Natürlich kann mit der Funktion „Transistor“ (Transistortester) wieder zu der normalen Transistortester-Funktion zurückgekehrt werden.

Bei gesetzter POWER_OFF-Option in der Makefile sind alle Zusatzfunktionen zeitbeschränkt, damit die Batterie nicht verbraucht wird.

3.3 Selbsttest und Kalibration

Wenn die Software mit der Selbsttestfunktion konfiguriert ist, kann der Selbsttest durch einen Kurzschluss aller drei Testports und Drücken der Starttaste eingeleitet werden. Für den Start des Selbsttests muss die Starttaste innerhalb von 2 Sekunden noch einmal gedrückt werden, sonst wird mit einer normalen Messung fortgefahrene. Beim Selbsttest werden die im Selbsttest-Kapitel 5.5 beschriebenen Tests ausgeführt. Wenn der Tester mit einer Menüfunktion (Option WITH_MENU) konfiguriert ist, wird der vollständige Selbsttest mit den Tests T1 bis T7 nur bei dem „Selbsttest“ ausgeführt, der als Menüfunktion ausgewählt werden kann. Außerdem wird beim Aufruf über die Menüfunktion der Abgleich mit dem externen Kondensator jedesmal durchgeführt, sonst nur für den ersten Aufruf. Damit kann der durch die kurgeschlossenen Testports automatisch gestartete Abgleich schneller durchgeführt werden. Die viermalige Testwiederholung bei T1 bis T7 kann vermieden werden, wenn der Starttaster gedrückt gehalten wird. So kann man uninteressante Tests schnell beenden und sich durch Loslassen des Starttasters interessante Tests viermal wiederholen lassen. Der Test 4 läuft nur automatisch weiter, wenn die Verbindung zwischen den Testports gelöst wird.

Wenn die Funktion AUTO_CAL in der Makefile gewählt ist, wird beim Selbsttest eine Kalibration der Nullwertes für die Kondensatormessung durchgeführt. Für die Kalibration des Nullwertes für die Kondensatormessung ist wichtig, dass die Verbindung zwischen den Testpins (Kurzschluß) während des Tests 4 wieder gelöst wird! Sie sollten während der Kalibration (nach dem Test 6) weder die Testports noch angeschlossene Kabel berühren. Die Ausrüstung sollte aber die gleiche sein, die später zum Messen verwendet wird. Andernfalls wird der Nullwert der Kondensatormessung nicht richtig bestimmt. Die Kalibration des Innenwiderstandes der Port-Ausgänge wird mit dieser Option vor jeder Messung durchgeführt.

Bei der Kalibration werden zwei Sonderschritte gemacht, wenn die samplingADC Funktion in der Makefile gewählt wurde (WITH_SamplingADC = 1). Nach der normalen Bestimmung der Nullwerte der Kapazitätssmessung werden dann auch die Nullwerte mit der Sampling Methode (C0samp) bestimmt. Als letzter Teil der Kalibration wird der Anschluß eines Testkondensators für die Spulenmessung an Pin 1 und Pin 3 angefordert mit der Meldung 1 —II— 3 10-30nF(L). Der Kapazitätswert sollte hierbei zwischen $10nF$ und $30nF$ liegen, um eine meßbare Resonanzfrequenz bei der späteren Parallelschaltung mit einer Spule ($< 2mH$) zu erreichen. Für Spulen mit mehr als $2mH$ Induktivität sollte die normale Testfunktion ohne die Parallelschaltung eines Kondensators ausreichen. Ein Parallelschalten des Kondensators sollte hier nicht mehr zu einer Verbesserung des Meßergebnisses führen.

Nach der Bestimmung der Nullkapazitäten ist der Anschluss eines Kondensators mit einer beliebiger Kapazität zwischen $100nF$ und $20\mu F$ an Pin 1 und Pin 3 erforderlich. Dazu wird in Zeile 1 ein 1 —II— 3 $>100nF$ angezeigt. Sie sollten den Kondensator erst nach der Ausgabe der C0 Werte oder nach der Ausgabe dieser Aufforderung anschließen. Mit diesem Kondensator wird die Offset-Spannung des analogen Komparators kompensiert, um genauere Kapazitätswerte ermitteln zu können. Die Verstärkung für ADC-Messungen mit der internen Referenz-Spannung wird ebenfalls mit diesem Kondensator abgeglichen, um bessere Widerstands-Messergebnisse mit der AUTOSCALE_ADC-Option zu erreichen. Wenn die Menüfunktion beim Tester ausgewählt wurde (Option WITH_MENU) und der Selbsttest nicht als Menüfunktion gestartet wurde, wird der Abgleich mit dem externen Kondensator nur bei der ersten Kalibration durchgeführt. Die Kalibration mit dem externen Kondensator kann nur wiederholt werden, wenn der Selbsttest als Menüfunktion ausgewählt wird. Der Nullwert für die ESR-Messung wird als Option ESR_ZERO in der Makefile vorbesetzt. Mit jedem Selbsttest wird der ESR Nullwert für alle drei Punkt-Kombinationen neu bestimmt. Das Verfahren der ESR-Messung wird auch für Widerstände mit Werten unter 10Ω benutzt um hier eine Auflösung von $0,01\Omega$ zu erreichen.

3.4 Besondere Benutzungshinweise

Normalerweise wird beim Start des Testers die Batteriespannung angezeigt. Wenn die Spannung eine Grenze unterschreitet, wird eine Warnung hinter der Batteriespannung ausgegeben. Wenn Sie eine aufladbare 9V-Batterie benutzen, sollten Sie den Akku möglichst bald austauschen oder nachladen. Wenn Sie eine Version mit eingebauter 2,5V-Präzisionsreferenz besitzen, wird beim Start für eine Sekunde die gemessene Betriebsspannung in der Zeile 2 mit „VCC=x.xxV“ angezeigt.

Es kann nicht oft genug erwähnt werden, dass Kondensatoren vor dem Messen entladen sein müssen. Sonst kann der Tester schon defekt sein, bevor der Startknopf gedrückt ist. Wenn man versucht, Bauelemente im eingebauten Zustand zu messen, sollte das Gerät immer von der Stromquelle getrennt sein. Außerdem sollte man sicher sein, dass keine Restspannung mehr im Gerät vorhanden ist. Alle Geräte haben Kondensatoren verbaut!

Beim Messen kleiner Widerstandswerte muss man besonders auf die Übergangswiderstände achten. Es spielt die Qualität und der Zustand von Steckverbindern eine große Rolle, genau so wie die Widerstandswerte von Messkabeln. Dasselbe gilt auch für die Messung des ESR-Wertes von Kondensatoren. Bei schlechten Anschlusskabeln mit Krokodilklemmen wird so aus einem ESR von $0,02\Omega$ leicht ein Wert von $0,61\Omega$. Wenn möglich sollte man Kabel mit Testklemmen an die drei Testports parallel zu vorhandenen Sockeln fest anschließen (anlöten). Dann braucht der Tester für kleine Kapazitäten nicht jedesmal neu kalibriert werden, wenn mit oder ohne eingesteckte Testkabel gemessen wird. Für die Kalibration des Nullwiderstandes macht es aber im allgemeinen einen Unterschied, ob die Testpins direkt am Sockel oder am Ende der Kabel mit den Testklemmen verbunden wird. Nur im letzteren Fall ist der Widerstand von Kabel und Klemmen mit kalibriert. Im Zweifelsfall kann man die Kalibration mit dem Kurzschluß am Testsockel durchführen und danach den Widerstand der

kurzgeschlossenen Klemmen mit dem Tester messen.

An die Genauigkeit der Messwerte sollte man keine übertriebenen Erwartungen haben, dies gilt besonders für die ESR Messung und die Induktivitätsmessung. Die Ergebnisse meiner Messreihen kann man im Kapitel 5 ab Seite 72 finden.

3.5 Problemfälle

Bei den Messergebnissen sollten Sie immer im Gedächtnis behalten, dass die Schaltung des Transistortesters für Kleinsignal-Bauelemente ausgelegt ist. Normalerweise beträgt der maximale Messstrom etwa $6mA$. Leistungshalbleiter machen oft wegen hoher Reststöme Probleme bei der Erkennung oder beim Messen der Sperrsicht-Kapazität. Bei Thyristoren und Triacs werden oft die Zündströme oder die Halteströme nicht erreicht. Deswegen kann es vorkommen, dass ein Thyristor als NPN-Transistor oder Diode erkannt wird. Ebenso ist es möglich, dass ein Thyristor oder Triac gar nicht erkannt wird. Probleme bei der Erkennung machen auch Halbleiter mit integrierten Widerständen. So wird die Basis-Emitter-Diode eines BU508D-Transistors wegen eines parallel geschalteten internen 42Ω Widerstandes nicht erkannt. Folglich kann auch die Transistorfunktion nicht geprüft werden. Probleme bei der Erkennung machen oft auch Darlington-Transistoren höherer Leistung. Hier sind auch oft Basis-Emitter-Widerstände verbaut, welche die Erkennung wegen der hier verwendeten kleinen Messströme erschweren.

3.6 Messung von PNP- und NPN-Transistoren

Normalerweise werden die drei Anschlüsse des Transistors in beliebiger Reihenfolge an die Messeingänge des Transistortesters angeschlossen. Nach dem Drücken des Starttasters meldet der Tester in der Zeile 1 den Typ (NPN oder PNP), eine eventuell vorhandene Schutzdiode der Kollektor-Emitter-Strecke und die Anschlussbelegung. Das Diodensymbol wird polungsrichtig angezeigt. In der Zeile 2 wird der Stromverstärkfaktor B oder hFE angegeben und der Strom, bei der die Verstärkung bestimmt wurde. Wenn die Emitterschaltung für die Bestimmung des Verstärkfaktors verwendet wurde, wird der Kollektorstrom Ic genannt. Bei Bestimmung des Verstärkfaktors in Kollektorschaltung (Emitterfolger) wird der Emitterstrom Ie angegeben. Weitere Parameter werden bei einem zweizeiligen Display nacheinander in Zeile 2 angegeben. Bei Displays mit mehr Zeilen werden weitere Zeilen benutzt bis die letzte Zeile schon benutzt war. Erst bei der letzten Zeile erfolgen weitere Ausgaben nacheinander in der letzten Zeile. Wenn weitere Parameter vorhanden sind, wird am Ende der letzten Zeile ein + Zeichen ausgegeben. Der nächste Wert erscheint auf Knopfdruck oder nach einer Wartezeit auch automatisch. Der nächste ausgegebene Parameter ist jedenfalls die Basis-Emitter-Schwellspannung Ube , bei der der Verstärkfaktor bestimmt wurde. Falls meßbar, werden auch der Kollektor-Reststrom mit offener Basis I_{CE0} und mit kurzgeschlossener Basis I_{CES} ausgegeben. Falls eine Schutzdiode vorhanden ist, wird die Flußspannung Uf dieser Schutzdiode als letzter Parameter ausgegeben.

Bei der Emitterschaltung hat der Tester nur zwei Möglichkeiten, den Basisstrom einzustellen:

1. Mit dem 680Ω Widerstand ergibt sich ein Basisstrom von etwa $6,1mA$. Das ist für einen Kleinsignaltransistor mit hohem Verstärkfaktor meist zu viel, weil die Basis gesättigt ist. Da der Kollektorstrom ebenfalls mit einem 680Ω Widerstand gemessen wird, kann der Kollektorstrom den um den Verstärkfaktor höheren Strom gar nicht erreichen. Die Softwareversion vom Markus F. hat in diesem Zustand die Basis-Emitter Spannung gemessen ($Uf=\dots$).
2. Mit dem $470k\Omega$ Widerstand] ergibt sich ein Basisstrom von nur $9,2\mu A$. Das ist für einen Leistungstransistor mit geringem Verstärkfaktor sehr wenig. Die Softwareversion von Markus

F. hat in diesem Zustand den Stromverstärkungsfaktor bestimmt ($hFE = \dots$).

Die Software des Testers bestimmt die Stromverstärkung jetzt auch in der Kollektorschaltung. Ausgegeben wird der höhere Wert der beiden Mess-Methoden. Die Kollektorschaltung hat den Vorteil, dass sich durch die Stromgegenkopplung der Basisstrom abhängig vom Verstärkungsfaktor reduziert. Dadurch kann für Leistungstransistoren mit dem 680Ω und für Darlington-Transistoren mit dem $470k\Omega$ Widerstand meist ein günstigerer Messstrom ergeben. Die ausgegebene Basis-Emitter-Schwellspannung U_{BE} ist jetzt die Spannung, die bei der Bestimmung des Stromverstärkungsfaktors gemessen wurde. Wenn man trotzdem eine Basis-Emitter-Schwellspannung bei circa $6mA$ ermitteln möchte, muss man den Kollektor vom Tester trennen und noch einmal messen. Dann wird die Schwellspannung bei etwa $6mA$ ausgegeben und die Kapazität der Diode in Sperr-Richtung ermittelt. Natürlich kann so auch die Basis-Kollektor-Diode gemessen werden.

Bei Germanium-Transistoren wird meistens ein Kollektor-Emitter-Reststrom I_{CE0} mit stromloser Basis oder ein Kollektor-Emitter-Reststrom I_{CES} mit Basis auf Emitterpotential gemessen. Durch Kühlen kann der Reststrom bei Germanium-Transistoren erheblich gesenkt werden.

3.7 Messung von JFET- und D-MOS-Transistoren

Wegen des symmetrischen Aufbaus von JFET-Transistoren kann Source und Drain nicht unterschieden werden. Normalerweise wird bei JFETs als eine Kenngröße der Strom bei kurzgeschlossenem Gate-Source angegeben. Dieser Strom ist aber oft höher als der, der sich bei der Messschaltung mit dem 680Ω Widerstand erreichen lässt. Deswegen wird der 680Ω Widerstand an den Source-Anschluss geschaltet. Dadurch erhält das Gate stromabhängig eine negative Vorspannung. Als Kenngröße wird sowohl der ermittelte Strom als auch die Gate-Source-Spannung ausgegeben. Damit können verschiedene Typen unterschieden werden. Für D-MOS-Transistoren (Verarmungs-Typ) wird das gleiche Messverfahren verwendet.

3.8 Messung von E-MOS Transistoren und IGBTs

Für Anreicherungs-MOS-Transistoren (P-E-MOS oder N-E-MOS) sollte man wissen, dass die Bestimmung der Gate-Schwellspannung (V_{th}) bei kleiner Gatekapazität schwierig wird. Hier können mit dem Tester genauere Spannungswerte ermittelt werden, wenn ein Kondensator mit einigen nF parallel zum Gate / Source angeschlossen wird. Die Schwellspannung wird bei Drain-Strömen von etwa $3,6mA$ für P-E-MOS und bei etwa $4mA$ für N-E-MOS bestimmt. Der Widerstand RDS or besser R_{DSon} von E-MOS Transistoren wird mit einer Gate - Source Spannung von annähernd $5V$ bestimmt, was wahrscheinlich nicht der niedrigste Wert ist. Außerdem wird der RDS Widerstand bei niedrigem Drain-Strom ermittelt, was die Auflösung des Widerstandwertes begrenzt. Oft bei IGBTs und manchmal auch bei Anreicherungs-MOS Transistoren reichen die verfügbaren $5V$ des Testers nicht aus, um den Transistor über das Gate anzusteuern. In diesem Fall hilft eine Batterie mit etwa $3V$ aus, um eine Erkennung und Messungen mit dem Tester möglich zu machen. Die Batterie wird dazu mit einem Pol an das Gate des Transistors angeschlossen und der andere Pol der Batterie wird dann anstelle des Transistor-Gates an einen Testport (TP) des Testers angeschlossen. Bei richtiger Polung der Batterie addiert sich die Batteriespannung zu der Steuerspannung des Testers und die Erkennung des Transistors gelingt. Natürlich muß dann die Batteriespannung zu der angezeigten Gate-Schwellspannung addiert werden, um die richtige Schwellspannung für dieses Bauteil zu erhalten.

3.9 Messen von Kondensatoren

Die Kapazitätswerte werden immer aus einer Zeitkonstanten berechnet, die sich aus der Serienschaltung von eingebauten Widerständen mit dem Kondensator bei Ladevorgängen ergeben. Bei kleineren Kondensatoren werden die $470k\Omega$ Widerstände für die Messung benutzt und die Zeit bis zum Erreichen einer Schwellspannung gemessen. Bei größeren Kondensatoren mit einigen $10\mu F$ wird die Spannungserhöhung des Kondensators nach Ladepulsen mit den 680Ω Widerständen beobachtet und daraus die Kapazität berechnet. Ganz kleine Kapazitäten können mit der samplingADC Methode gemessen werden. Dabei wird ein Aufladevorgang immer wieder erzeugt und durch Verschiebung der ADC S&H Zeit mit dem Zeitabstand abgetastet, der sich aus der Taktfrequenz des Prozessors ergibt. Eine komplette ADC Wandlung braucht hingegen 1664 Prozessortakte! Bis zu 250 ADC-Werte werden so ermittelt und aus dem Spannungsverlauf die Kapazität berechnet. Wenn die samplingADC Funktion in der Makefile gewählt wurde, werden alle Kondensatoren $< 100pF$ in der Kondensatormeßfunktion [C] mit der samplingADC Methode gemessen. Die Auflösung beträgt hier bis zu $0.01pF$ bei $16MHz$ Taktfrequenz. Der Abgleich der Nullkapazität stellt bei dieser Auflösung eine besondere Herausforderung dar. Die samplingADC Methode des Kapazitätsbestimmung wurde immer angewendet, wenn es Bruchteile von $1pF$ dargestellt werden. Übrigens können auch die Sperrsichtkapazitäten von Einzeldioden mit dieser Methode gemessen werden. Da die Methode die Kapazität sowohl beim Laden als auch beim Entladen messen kann, werden zwei Kapazitätswerte angegeben. Wegen des Kapazitätsdioden-Effektes unterscheiden sich die beiden Werte.

3.10 Messen von Spulen

Das angewandte Verfahren zum Messen der Induktivität besteht aus einer Zeitkonstantenbestimmung des Stromanstiegs. Die Detektionsgrenze ist hier $0.01mH$, wenn der Widerstandswert der Spule kleiner 24Ω beträgt. Bei größeren Widerstandswerten beträgt die Auflösung nur $0.1mH$. Bei Widerstandswerten über $2.1k\Omega$ kann die Methode gar nicht angewendet werden. Die Ausgabewerte der normalen Messung erscheinen in der zweiten Zeile (Widerstand und Induktivität). Mit dem samplingADC Verfahren können bei größeren Spulenwerten Eigenresonanzen festgestellt werden. Wenn diese deutlich genug ist, wird die Frequenz und die Güte Q zusätzlich zu der normalen Messung in Zeile 3 ausgegeben.

Daneben lässt sich diese Methode der Schwingfrequenz-Messung auch für die Bestimmung des Induktivitätswertes benutzen, wenn ein ausreichend großer Kondensator mit bekanntem Kapazitätswert einer Spule mit kleiner Induktivität ($< 2mH$) parallelgeschaltet wird. Durch die Parallelschaltung des Kondensators funktioniert die normale Messung der Zeitkonstante nicht mehr. Die Induktivität der normalen Messung wird gar nicht ausgegeben und der Widerstandswert wird in der ersten Zeile ausgegeben, wenn die Resonanzfrequenz eine Parallelschaltung des Kondensators erwarten lässt. Auch für diesen Fall wird eine Güte Q aus den Schwingverhalten berechnet. Für diesen Fall wird die berechnete Induktivität an erster Stelle in Zeile 2 angegeben. Dahinter steht dann der Text if "gefolgt von dem Kapazitätswert des Parallelkondensators. Der Wert dieses Kondensators kann derzeit nur in der Kalibrationsfunktion durch Messung festgelegt werden (1 — 3 10-30nF(L)). Die dritte Zeile zeigt wieder die Schwingfrequenz und die ermittelte Güte Q. Bei zweizeiligen Displays wird der Inhalt der 3. Zeile zeitverzögert in Zeile 2 ausgegeben.

Kapitel 4

Programmieren des TransistorTesters

4.1 Konfigurieren des TransistorTesters

Die ganze Software des TransistorTesters ist im Quellcode verfügbar. Die Übersetzung der Module wird mit einer `Makefile` gesteuert. Die Entwicklung wurde auf einem Ubuntu Linux mit den GNU-Werkzeugen (GNU toolchain, gcc version 4.5.3) durchgeführt. Es sollte möglich sein, ohne Schwierigkeiten andere Linux-Betriebssysteme zu benutzen. Um die übersetzten Daten in den Flash-Speicher oder den EEPROM-Speicher zu laden, wird das Programm `avrdude` [14] (Version 6.3) von der `Makefile` benutzt, wenn man `make upload` aufruft. Das Programm `avrdude` ist für Linux und Windows verfügbar. Der GNU C-Kompiler wird auch von der AVR-studio-Software unter Windows oder von der WinAVR [18],[19] Software benutzt. Sie können die Programmdaten (.hex und .eep) auch mit anderen Programmen in den ATmega laden, aber nur meine `Makefile`-Version stellt sicher, dass die richtigen Daten in den gewählten Prozessor gelangen. Avrdude lädt Daten nur in den ATmega, wenn die Signaturbytes des angeschlossenen ATmega gleich mit dem ausgewählten sind. Wenn Sie die `Makefile` ändern, wird die Software komplett neu übersetzt, wenn man `make` oder `make upload` aufruft. Die Software, die für einen ATmega8 übersetzt wurde, läuft nicht auf einem ATmega168. Die Software, die für einen ATmega328 übersetzt wurde, läuft nicht auf einem ATmega168. Eine Ausnahme bildet Software, die für einen ATmega168 übersetzt wurde. Diese Programmdateien sind auch für einen ATmega328 brauchbar. Seien Sie vorsichtig, wenn Sie nicht das mitgelieferte `Makefile` benutzen.

Mit den entsprechenden Optionen ist die Software auch auf dem unveränderten Hardware-Entwurf von Markus F. lauffähig (PARTNO=m8 , **keine** NO_AREF_CAP und **keine** PULLUP_DISABLE Option). Die Taktrate kann mit den fuses auch auf 8MHz gestellt werden, dazu ist kein Quarz erforderlich!

Die folgenden Optionen der `Makefile` sind verfügbar, um die Software für den Tester zu konfigurieren:

PARTNO beschreibt den Ziel-Prozessor:

m8 = ATmega8
m168 or m168p = ATmega168
m328 or m328p = ATmega328
m644 or m644p = ATmega644
m1284p = ATmega1284
m1280 = ATmega1280
m2560 = ATmega2560
Beispiel: PARTNO = m168

UI_LANGUAGE gibt die Sprache für den Tester an:

LANG_BRASIL, LANG_CZECH, LANG_DANISH, LANG_DUTCH, LANG_ENGLISH,
LANG_GERMAN, LANG_HUNGARIAN, LANG_ITALIAN, LANG_LITHUANIAN,

LANG_POLISH, LANG_RUSSIAN, LANG_SLOVAK, LANG_SLOVENE, LANG_SPANISH, und LANG_UKRAINIAN sind derzeit verfügbar. Für die russische und ukrainische Sprache ist ein LCD mit kyrillischem Zeichensatz erforderlich.

Beispiel: UI_LANGUAGE = LANG_ENGLISH

LCD_CYRILLIC wird nur gebraucht, wenn man ein LC-Display mit kyrillischem Zeichensatz benutzt. Die Zeichen μ und Ω sind im kyrillischen Zeichensatz nicht enthalten. Wenn Sie diese Option angeben, werden beide Zeichen von der Software in das LCD geladen. Setzen sie diese Option, wenn bei der Ausgabe falsche Zeichen anstelle von μ oder Ω erscheinen.

Beispiel: CFLAGS += -DLCD_CYRILLIC

LCD_DOGM muss angegeben werden, wenn ein LCD mit ST7036-Controller (Typ DOG-M) zur Anzeige verwendet wird. Der LCD-Kontrast wird dann mit Software-Befehlen eingestellt. Falls sie den Kontrast zu weit verstellt haben, daß auf dem Display nichts mehr zu erkennen ist, können sie zunächst versuchen, ob nicht bei seitlicher Sicht auf das Display doch noch etwas zu erkennen ist. sonst muß das EEPROM neu beschrieben werden, um den Kontrastwert zurückzusetzen.

Beispiel: CFLAGS += -DLCD_DOGM

FOUR_LINE_LCD kann bei einem 4x20-Zeichen-Display verwendet werden, um den Platz auf dem Display besser auszunutzen. Zusätzliche Parameter, welche sonst nur kurz in Zeile 2 angezeigt werden, werden dann in den Zeilen 3 und 4 angezeigt. Bei grafischen Displays wird die Zahl der Zeilen aus der Displayhöhe und dem Zeilenabstand berechnet.

Beispiel: CFLAGS += -DFOUR_LINE_LCD

LCD_LINE_LENGTH gibt die Anzahl der Zeichen an, die in einer Zeile dargestellt werden können. Bei grafischen Displays wird die Anzahl der Zeichen aus der Zeichenbreite und der Displaybreite berechnet.

Beispiel: CFLAGS += -DLCD_LINE_LENGTH=20

DD_RAM_OFFSET Bei einigen Text-Displays werden andere DD-RAM Startadressen für den Zeilenanfang benutzt. Normalerweise beginnt die erste Zeile bei der DD-RAM Adresse 0. Bei einigen Displays wie TC1604 oder TC1602 beginnt die Zeile 1 aber bei der Adresse 128 (0x80). Das kann mit dieser Option berücksichtigt werden.

Beispiel: CFLAGS += -DDRAM_OFFSET = 128

WITH_LCD_ST7565 Diese Option muss verwendet werden, wenn ein 128x64 Pixel LCD mit serieller Schnittstelle angeschlossen ist. Für diesen Displaytyp müssen weitere Optionen gesetzt werden, die in Tabelle 4.1 beschrieben werden. Anstelle des ST7565-Controllers kann auch zum Beispiel der ähnliche SSD1306-Controller konfiguriert werden. Dafür muss diese Option auf 1306 gesetzt werden. Ein PCF8812 oder PCF8814 Controller wird ebenfalls unterstützt, wenn diese Option richtig gesetzt wird. Auch ein Display mit ST7920 oder NT7108 Controller kann angeschlossen werden. Für den NT7108 Controller muß ein zusätzlicher Seriell-Parallel Wandler 74HC(T)164 oder 74HC(T)595 verwendet werden.

Beispiel: WITH_LCD_ST7565 = 1

LCD_INTERFACE_MODE Beim SSD1306-Controller kann anstelle des Standard-SPI (4-Wire) Interface auch das I²C-Interface benutzt werden. Dafür muss diese Option auf 2 gesetzt werden. Für den ST7920 Controller kann ein spezieller serieller Anschluß mit dem Setzen dieser Option auf 5 unterstützt werden. Wenn nur eine Anschlußmöglichkeit vorgesehen ist, braucht die Konstante LCD_INTERFACE_MODE nicht gesetzt werden. Alle bisher möglichen Werte für LCD_INTERFACE_MODE und WITH_LCD_ST7565 sind in Tabelle 4.1 aufgeführt.

Display-Typ	Interface	WITH_LCD_ST7565	LCD_INTERFACE_MODE
Character 16x2, Character 20x4	4-Bit parallel 4-Wire SPI I ² C	disabled (0)	disabled (1) 4 2
Graphic ST7565	4-Wire SPI	1 or 7565	disabled (4)
Graphic ST7565	I ² C	1 or 7565	2
Graphic SSD1306	4-Wire SPI	1306	disabled (4)
Graphic SSD1306	I ² C	1306	2
Graphic ST7920	4-Bit parallel	7920	disabled (1)
Graphic ST7920	2-Bit serial	7920	5
Graphic NT7108 oder KS0108	8-Bit parallel + 74HCT164	7108	disabled (6)
Graphic PCF8812	4-Wire SPI	8812	disabled (4)
Graphic PCF8814	4-Wire SPI I ² C 3-line	8814 8814 8814	disabled (4) 2 3
Graphic ILI9163 128x128 Color	4-Wire SPI	9163	disabled (4)
Graphic ST7735 128x160 Color	4-Wire SPI	7735	disabled (4)

Tabelle 4.1. Kennzahlen für Controller und Art der Schnittstelle

Die Werte in Klammern werden nur Software intern benutzt und sind hier nur zur Information genannt. Man sollte die Werte in den Klammern nicht hier in der `Makefile` setzen.
Beispiel: `CFLAGS += -DLCD_INTERFACE_MODE=2`

LCD_SPI_OPEN_COL Mit der Option `LCD_SPI_OPEN_COL` werden die Datensignale der SPI-Schnittstelle nicht nach VCC direkt geschaltet. Die Signale werden nur nach GND geschaltet, für die „high“ Signale werden die „Pull-Up“ Widerstände des ATmega benutzt. Für das Signal „Reset“ wird aber ein externer Pull-Up Widerstand benötigt, wenn die Option `PULLUP_DISABLE` gesetzt ist. Für die anderen SPI Signale werden die internen Pull-Up Widerstände vorübergehend benutzt, wenn die Option `PULLUP_DISABLE` gesetzt ist. Beispiel: `CFLAGS += -DLCD_SPI_OPEN_COL`

LCD_I2C_ADDR Die I²C-Adresse des SSD1306-Controllers kann durch Vorbesetzen der Adresse `LCD_I2C_ADDR` auf 0x3D geändert werden.

Beispiel: `CFLAGS += -DLCD_I2C_ADDR=0x3d`

LCD_ST7565_RESISTOR_RATIO Mit dieser Option wird das Widerstands-Teilverhältnis für den Spannungsregler des ST7565-Controllers eingestellt. Brauchbare Werte liegen im Allgemeinen zwischen 4 und 7. Einstellbar sind Werte zwischen 0 und 7.

Beispiel: `LCD_ST7564_RESISTOR_RATIO = 4`

LCD_ST7565_H_FLIP Mit dieser Option kann die Anzeige horizontal umgedreht werden.

Beispiel: `CFLAGS += LCD_ST7565_H_FLIP = 1`

LCD_ST7565_H_OFFSET Diese Option kann den für die Ausgabe benutzten Speicher an das Anzeigefenster des Displays anpassen. Der Controller benutzt mehr horizontale Pixel (132) als angezeigt (128) werden. Ein Wert von 0, 2 oder 4 kann je nach verwendetem Display-Modul für eine korrekte Darstellung notwendig sein.

Beispiel: `CFLAGS += LCD_ST7565_H_OFFSET = 4`

LCD_ST7565_V_FLIP Mit dieser Option kann die Anzeige vertikal umgedreht werden.

Beispiel: CFLAGS += LCD_ST7565_V_FLIP = 1

VOLUME_VALUE Hier kann man den Kontrastwert für ST7565 oder SSD1306 Controller voreinstellen. Der Wert kann für den ST7565 Controller zwischen 0 und 63 liegen. Für den SSD1306 Controller sind Werte zwischen 0 und 255 erlaubt.

Beispiel: CFLAGS += -DVOLUME_VALUE = 25

LCD_ST7565_Y_START Mit dieser Option kann die erste Zeile vertikal richtig gesetzt werden.

Bei einigen Display-Varianten ist die erste Zeile in die Bildmitte verschoben. Bei diesen Displays wird die erste Zeile wieder an den oberen Rand verschoben, wenn diese Option auf 32 (halbe Bildschirmhöhe) gesetzt wird.

Beispiel: CFLAGS += -DLCD_ST7565_Y_START = 32

LCD_CHANGE_COLOR Diese Option erweitert die Menüfunktionen um die Möglichkeit, die Hintergrundfarbe und die Vordergrundfarbe wählen zu können. Wenn die Option auf den Wert 2 gesetzt wird, werden die Farben blau und rot vertauscht. Diese Option kann nur für Farbdisplays (Controller ST7735 oder ILI9193) gewählt werden.

Beispiel: CFLAGS += -DLCD_CHANGE_COLOR=1

LCD_BG_COLOR Mit dieser 16-Bit Zahl kann eine Farbe für den Hintergrund gewählt werden.

Normalerweise sind die oberen 5 Bit für die Farbe rot, die mittleren 6 Bit für die Farbe grün und die unteren 5 Bit für die Farbe blau vorgesehen. Manchmal sind aber die Bits für die Farben rot und blau vertauscht. Diese Option kann nur für Farbdisplays (Controller ST7735 oder ILI9193) gewählt werden.

Beispiel: CFLAGS += -DLCD_BG_COLOR=0x000f

LCD_FG_COLOR Mit dieser 16-Bit Zahl kann eine Farbe für den Vordergrund gewählt werden.

Das Beispiel wählt weiss als Farbe für Text und Symbole aus. Diese Option kann nur für Farbdisplays (Controller ST7735 oder ILI9193) gewählt werden.

Beispiel: CFLAGS += -DLCD_FG_COLOR=0xffff

FONT_8X16 Für den ST7565 controller sollte ein Font ausgewählt werden. Auswählbar sind die Fonts mit der Variablen FONT_ mit angehängter Größe (Breite x Höhe). iDerzeit sind 6X8, 8X8, 7X12, 8X12, 8X12thin, 8X14 8X15, 8X16 und 8X16thin. Die Fontgröße 8x16 oder 8X16thin ist die effizienteste Wahl für das graphische 128x64-LCD.

Beispiel: CFLAGS += FONT_8X16

BIG_TP Die Pinnummern bei der graphischen Darstellung können mit dieser Option größer dargestellt werden.

Beispiel: CFLAGS += BIG_TP

INVERSE_TP Mit dieser Option werden die Pinnummern in der graphischen Darstellung invers (schwarz auf weiß) dargestellt. Weil für die Darstellung ein Rand benötigt wird, kann diese Option nicht mit der Option BIG_TP kombiniert werden.

Beispiel: CFLAGS += INVERSE_TP

STRIP_GRID_BOARD Diese Option passt die Software an eine andere Pinbelegung von Port D für Streifenleiterplatinen an. Die Einzelheiten findet man im Hardwarekapitel 2.1 auf Seite 10. Mit dieser Option werden auch alternative Belegungen der ATmega Pins für graphische Displays gewählt. Für die chinesische „T5“ Platine muß die Option STRIP_GRID_BOARD auf 5 gesetzt werden. Bei den Alternativen für die graphischen Displays bleibt die Belegung des Tastensignals unverändert.

Beispiel: CFLAGS += -DSTRIP_GRID_BOARD

WITH_MENU aktiviert eine Menüfunktion für einen ATmega328. Man kann einige Zusatzfunktionen über ein Auswahlmenü benutzen, welches man über einen langen Tastendruck (> 0,5s)

erreichen kann. Wenn die Menüfunktion eingeschaltet ist, wird beim automatischen Selbsttest-Start mit kurzgeschlossenen Testpins nur der Kalibrationsteil des Selbsttests ausgeführt. Die Tests T1-T7 werden nur beim Selbsttest ausgeführt, der als Menüfunktion ausgewählt werden kann.

Beispiel: `CFLAGS += -DWITH_MENU`

MAX_MENU_LINES Diese Option gibt eine maximale Zeilenzahl für die gezeigte Auswahl der Menüfunktionen an. Normalerweise ergibt sich die Zahl der Zeilen über die vorhandene Zeilenanzahl des Displays. Da normalerweise mehr Funktionen ausgewählt werden können als auf dem Display Zeilen zur Verfügung stehen, wird die Auswahl zyklisch ausgetauscht. Die Aufbereitung des Displayinhalts bei dem zyklischen Austausch braucht speziell bei großen Farbdisplays mit vielen Zeilen erhebliche Zeit. Durch die Begrenzung der Zeilenanzahl mit dieser Option kann die Ausgabezeit bei der Menüauswahl deutlich reduziert werden und so die Bedienung beschleunigt werden. Der Wert für diese Option ist mit 5 vorbesetzt.

Beispiel: `CFLAGS += -DMAX_MENU_LINES=3`

WITH_ROTARY_SWITCH Die Menüfunktion kann leichter bedient werden, wenn ein Impulsdrehgeber als Erweiterung benutzt wird. Für die Details der notwendigen Erweiterung sehen Sie bitte die Beschreibung 2.6 im Hardwarekapitel. Wenn der Impulsdrehgeber die gleiche Anzahl von Raststellungen wie Impulse für jede Umdrehung hat, muss die Option `WITH_ROTARY_SWITCH` auf 2 gesetzt werden. Wenn der Impulsdrehgeber doppelt so viele Raststellungen hat, muss die Option `WITH_ROTARY_SWITCH` auf 1 gesetzt werden. Das Setzen der Option `WITH_ROTARY_SWITCH` auf 5 wählt die höchste Auflösung für den Impulsdrehgeber. Jeder Zyklus der beiden Schalter-Zustände wird als 4 gezählt. Normalerweise ist diese Einstellung nur für Impulsdrehgeber ohne Raststellung sinnvoll. Ein Setzen der Option `WITH_ROTARY_SWITCH` auf 4 ist notwendig für die korrekte Behandlung von zwei separaten Rauf (Up) und Runter (Down) Tastern, die anstelle der beiden Schalter eines Impulsdrehgebers eingebaut sind. Verwenden Sie nicht die Einstellung 4 mit normalen Impulsdrehgebern!

Beispiel: `CFLAGS += -DWITH_ROTARY_SWITCH=1`

CHANGE_ROTARY_DIRECTION Man kann die erkannte Drehrichtung des Impulsdrehgebers durch Vertauschen der beiden Schalter-Signale oder durch Setzen dieser Option ändern. Beispiel: `CFLAGS += -DCHANGE_ROTARY_DIRECTION`

WITH_SELFTEST Wenn Sie diese Option angeben, baut die Software eine Selbsttest-Funktion ein, die gestartet wird wenn Sie alle drei Prüfspitzen verbinden und eine Messung starten.

Beispiel: `CFLAGS += -DWITH_SELFTEST`

NO_COMMON_COLLECTOR_HFE verhindert die hFE-Messung von Transistoren in der Kollektorschaltung. So können Sie Speicher sparen, um die erweiterten Selbsttest-Routinen T1 bis T7 für den ATmega168-Prozessor zu ermöglichen. Standardmäßig sind beide Schaltungen für die hFE-Messung eingeschaltet, aber dann ist kein Platz im Programmspeicher des ATmega168 für die erweiterten Selbsttests.

Beispiel: `CFLAGS += -DNO_COMMON_COLLECTOR_HFE`

NO_COMMON_EMITTER_HFE schaltet die hFE-Messung von Transistoren in Emitterschaltung ab. So können Sie Speicher sparen, um die erweiterten Selbsttest-Routinen T1 bis T7 für den ATmega168-Prozessor zu ermöglichen. Standardmäßig sind beide Schaltungen für die hFE-Messung eingeschaltet, aber dann ist kein Platz im Programmspeicher des ATmega168 für die erweiterten Selbsttests.

Beispiel: `CFLAGS += -DNO_COMMON_EMITTER_HFE`

NO_TEST_T1_T7 Diese Option verhindert die Ausführung der Selbsttest Teile T1 bis T7. Diese Tests sind nützlich um Fehler in der Schaltung wie falsche Messwiderstände oder Isolationsprobleme zu finden. Wenn Ihre Schaltung fehlerfrei ist, können Sie die Selbsttest-Teile T1 bis

T7 durch das Setzen dieser Option weglassen, um eine schnellere Kalibration zu erreichen. Bei eingeschalteter Menüfunktion werden die Selbsttest-Teile T1 bis T7 nur bei Aufruf der Menüfunktion „Selbsttest“ ausgeführt. Der ATmega168-Prozessor benutzt die Selbsttest Teile T1 bis T7 nicht, wenn beide Messmethoden für die hFE-Bestimmung benutzt werden.

Beispiel: CFLAGS += -DNO_TEST_T1_T7

AUTO_CAL Der Nullabgleich für die Kondensatormessung wird beim Selbsttest zusätzlich ins EEPROM geschrieben und ist damit für die weiteren Messungen abgeglichen. Wenn nach dem Nullabgleich der Kondensatormessung ein Kondensator mit einer Kapazität zwischen $100nF$ und $20\mu F$ an Pin 1 und Pin 3 angeschlossen wird, wird auch der Offset des analogen Komparators und die Skalierung für die AUTOSCALE_ADC Umschaltung auf die interne Spannungsreferenz ermittelt und ins EEPROM geschrieben. Die Port-Ausgangswiderstände werden zu Beginn jeder Messung neu bestimmt.

Beispiel: CFLAGS += -DAUTO_CAL

SHORT_UNCAL_MSG Bei Prozessoren mit mindestens 32K Flash wird nach einem Bauteiletetest ein Hinweis auf einen unkalibrierten Zustand des Testers gegeben. Normalerweise folgt dann eine kurze Anleitung, wie die Kalibration durchzuführen ist. Diese Anleitung wird nicht ausgegeben, wenn die Option SHORT_UNCAL_MSG gesetzt wird. Dann bleibt es bei einem einzeiligen Hinweis auf den unkalibrierten Zustand. Dies spart einerseits etwas Platz in Flash-Speicher und andererseits Ausgabezeit für Benutzer, die ohnehin wissen, wie kalibriert wird.

Beispiel: CFLAGS += -DSHORT_UNCAL_MSG

NO_ICONS_DEMO Diese Option schaltet die zusätzliche Demonstration der Symbole und die Ausgabe des Zeichensatzes bei der Menüfunktion „Zeige Daten“ ab. Dies spart Platz im Flash-Speicher und auch Ausgabezeit für den Nutzer.

Beispiel: CFLAGS += -DNO_ICONS_DEMO

WITH_ROTARY_CHECK Diese Option schaltet die zusätzliche Menüfunktion für den Test eines Impulsdrehgebers frei. Dieser Test testet einen Impulsdrehgeber, der an TP1,TP2 und TP3 angeschlossen wird. Bitte beachten Sie, daß nicht der eingebaute Impulsdrehgeber des Testers getestet wird! Aber ein solcher Impulsdrehgeber kann mit der Option WITH_ROTARY_SWITCH auch zur Bedienung des Testers eingesetzt werden.

Beispiel: CFLAGS += -DWITH_ROTARY_CHECK

NO_FREQ_COUNTER Mit dieser Option wird die Frequenzzähler Funktion des Testers abgewählt. Dies ist besonders dann sinnvoll, wenn der benutzte Pin PD4 (ATmega328) nicht zusammen mit dem Anschluß des Displays benutzt werden kann. Der entsprechende Eintrag in der Liste der Menüfunktionen erscheint dann nicht mehr und es wird auch Flash-Speicherplatz gespart.

Beispiel: CFLAGS += -DNO_FREQ_COUNTER

WITH_FREQUENCY_DIVIDER Mit dieser Option wird das Menü um einen einstellbaren Vorteiler für den Frequenzzähler erweitert. Das Teilverhältnis kann auf 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64 und 1:128 eingestellt werden. Diese Option ist nur sinnvoll, wenn an den Frequenzeingang ein externer Vorteiler angeschlossen wird. Die bei der Messung angezeigten Frequenzen und Perioden berücksichtigen das eingestellte Teilverhältnis.

Beispiel: CFLAGS += -DWITH_FREQUENCY_DIVIDER

NO_FREQUENCY_SWITCH Mit dieser Option wird die Umschaltung des Frequenzeingangs der ATmega644/ATmega1284 Schaltung nicht unterstützt. Die LF- und HF-Quarz Messung wird mit dieser Option nicht unterstützt.

Beispiel: CFLAGS += -DNO_FREQUENCY_SWITCH=1

PWM_SERVO wählt eine spezielle PWM Erzeugung zum Testen von Modellbau Servos anstelle der allgemeinen PWM Erzeugung.

Beispiel: CFLAGS += -DPWM_SERVO=1

WITH_SamplingADC Mit dieser Funktion wird in bestimmten Fällen die Sampling-Methode für den ADC angewendet. Durch Verschieben der Abtastzeit des ADC kann für wiederholbare Signale die Signalform mit einem Zeitabstand von einem Prozessortakt oder auch 4 oder 16 Prozessortakte abgetastet werden. Dadurch kann die Ladefunktion von Kondensatoren unter 100pF so vermessen werden, daß sich eine Auflösung von 0.01pF bei 16MHz Prozessortakt ergibt. Mit den gleichen Methode kann die Schwingfrequenz bei kleinen Spulen unter 2mH mit einem parallelgeschalteten Kondensator gemessen werden. Wenn die Kapazität des Parallel-Kondensators bekannt ist, kann die Induktivität aus der Schwingfrequenz mit hoher Auflösung bestimmt werden. Als Nebenprodukt kann auch die Güte der Spule aus dem Schwingverhalten geschätzt werden. Freigeschaltet werden diese Funktionen mit der Option WITH_SamplingADC. Bei der Kalibration werden dann sowohl zusätzlich die Nullkapazitäten mit der Samplingmethode zusätzlich bestimmt, als auch die Kapazität eines Parallel-Kondensators vermessen für die spätere Bildung eines Schwingkreises mit einer unbekannten Spule.

Beispiel: WITH_SamplingADC = 1

WITH_XTAL Diese Option schaltet zusätzlich Tests für Quarze und Resonatoren frei, wenn die SamplingADC Funktion schon vorhanden ist und ein 16 MHz Quarz verwendet wird (OP_MHZ = 16). Wenn möglich, werden die Frequenzen für Serien- und Parallel-Schaltung bestimmt und dann versucht, aus dem Frequenzversatz die Serienkapazität Cm zu bestimmen.

Beispiel: CFLAGS += -DWITH_XTAL

WITH_UJT Diese Option schaltet zusätzliche Tests für Unijunctiontransistoren frei. Wenn die SamplingADC Funktion freigeschaltet wurde, wird die Schwingfähigkeit des Bauteils untersucht. Richtig erkannt wird das Bauteil aber auch ohne die SamplingADC Funktion. Ohne die Option WITH_UJT werden Unijunctiontransistoren als Doppeldiode erkannt.

Beispiel: CFLAGS += -DWITH_UJT

WITH_PUT Diese Option schaltet einen zusätzlichen Test auf „Programmable Unijunction Transistor“ frei. Ohne den zusätzlichen Test werden PUTs normalerweise als Bipolartransistor (Bipolar Junction Transistor) erkannt.

Beispiel: CFLAGS += -DWITH_PUT

FET_Idss Diese Option bewirkt zusätzliche Messungen, um den Drain-Strom Idss zu berechnen, wenn die Schätzung nicht über 60mA liegt. Die Schätzung und Berechnung wird mit einem angenommenen quadratischen Stromverlauf durchgeführt. Beispiel: CFLAGS += -DFET_Idss

FREQUENCY_50HZ Zum Ende des Selbsttests wird bis zu einer Minute lang ein 50Hz -Signal auf Port 2 und Port 3 erzeugt. Diese Option sollte nur in Ausnahmefällen bebraucht werden.

Beispiel: CFLAGS += -DFREQUENCY_50HZ

CAP_EMPTY_LEVEL Diese Option legt die Spannung (mV) für einen entladenen Kondensator fest. Der Wert kann höher als 3mV gesetzt werden, wenn die Entladung nicht zum Ende kommt. In diesen Fall meldet der Tester nach längerer Zeit „Cell!“.

Beispiel: CFLAGS += -DCAP_EMPTY_LEVEL=3

WITH_AUTO_REF Mit dieser Option wird die Referenzspannung gemessen, um den aktuellen Faktor für die Kapazitätsmessung von kleineren Kapazitäten (unter $40\mu\text{F}$) zu ermitteln.

Beispiel: CFLAGS += -DWITH_AUTO_REF

REF_C_KORR gibt einen Offset für die gelesene Referenz-Spannung in mV-Einheiten an. Das kann benutzt werden, um die Kapazitätsmessung kleiner Kondensatoren abzugleichen. Wenn zusätzlich die AUTO_CAL-Option gewählt wurde, ist diese Angabe nur ein zusätzlicher Offset für den gefundenen Komparator-Offset. Ein Wert von 10 ergibt etwa 1 Prozent kleinere Messergebnisse.

Beispiel: CFLAGS += -DREF_C_KORR=14

REF_L_KORR gibt einen zusätzlichen Offset für die Referenz-Spannung für die Induktivitätsmessung in mV-Einheiten an. Der REF_C_KORR-Offset beziehungsweise der gefundene Offset bei der Kalibration wird bei der Induktivitätsmessung ebenfalls berücksichtigt.

Der REF_L_KORR Wert wird für die Messungen ohne 680Ω Widerstand subtrahiert, bei Messungen mit einem 680Ω Widerstand wird der Wert addiert. Ein Wert von 10 führt zu einer Änderung des Ergebnisses um 1 Prozent.

Beispiel: CFLAGS += -DREF_L_KORR=40

C_H_KORR gibt eine Korrektur der Messergebnisse für grosse Kondensatoren an. Eine Eingabe von 10 führt zu 1 Prozent kleineren Messergebnissen.

Beispiel: CFLAGS += -DC_H_KORR=10

WITH_UART benutzt den Pin PC3 zur Ausgabe der seriellen Texte (V24). Wenn die Option nicht benutzt wird, kann der PC3 Pin zum Anschluss einer externen Spannung mit einem 10:1-Widerstandsteiler benutzt werden. Damit können beispielsweise Zenerdioden mit höherer Durchbruchspannung getestet werden. Diese Messung wird so lange mit etwa 3 Messungen pro Sekunde wiederholt, solange der Starttaster gedrückt bleibt.

Beispiel: CFLAGS += -DWITH_UART

WITH_HARDWARE_SERIAL wird nur für die serielle Ausgabe bei Testern benutzt, die mit einer Arduino UNO Platine gebaut wurden.

Example: CFLAGS += -DWITH_HARDWARE_SERIAL

TQFP_ADC6 Die Option TQFP_ADC6 benutzt anstelle des PC3-Pins (ADC3) den zusätzlichen ADC-Eingang ADC6 des ATmegas im TQFP-Gehäuse oder QFN-Gehäuse. Dadurch kann dieser Eingang unabhängig von der seriellen Ausgabe auf dem PC3 Pin genutzt werden. Dieser Pin wird dann für die Zenerdioden-Messung und für die Messung einer externen Spannung über den Dialog des ATmega328 genutzt.

Beispiel: CFLAGS += -DTQFP_ADC6

TQFP_ADC7 Die Option TQFP_ADC7 benutzt anstelle des PC3-Pins (ADC3) den zusätzlichen ADC-Eingang ADC7 des ATmegas im TQFP-Gehäuse und QFN-Gehäuse. Dadurch kann dieser Pin unabhängig von der seriellen Ausgabe auf den PC3 Pin genutzt werden. Wenn diese Option ohne die Option TQFP_ADC6 genutzt wird, erfolgt sowohl die Zenerdioden-Messung als auch die Messung einer externen Spannung über den Dialog des ATmega328 genutzt. Wenn die Option zusätzlich zur TQFP_ADC6-Option gesetzt wird, erfolgt die Zenerdioden-Messung mit dem ADC6-Pin und bei der über den Dialog wählbaren Spannungsmessung werden beide Eingänge gemessen. Beide Pinne sollten dann an einen 10:1-Spannungsteiler angeschlossen sein.

Beispiel: CFLAGS += -DTQFP_ADC7

WITH_VEXT ermöglicht die Messung einer externen Spannung über einen 10:1-Spannungsteiler. Für den ATmega168 oder ATmega328 wird normalerweise der PC3-Pin benutzt, wenn keine Option TQFP_ADC6 oder TQFP_ADC7 gesetzt ist. Dann ist diese Option aber nur möglich, wenn die WITH_UART Option nicht gesetzt ist.

Beispiel: CFLAGS += -DWITH_VEXT

RMETER_WITH_L wählt für die Widerstandsmeßfunktion, die durch einen Widerstand an TP1 und TP3 gestartet wird, zusätzlich die Messung von Induktivitäten. Der Betriebsmodus wird dann durch ein [RL] am Ende der ersten Displayzeile angezeigt. Durch den zusätzlichen Test auf Induktivität wird die Meßzeit für Widerstände unter 2100Ω deutlich länger. Ohne diese Option werden außerdem Widerstände unter 10Ω nicht mit der ESR-Methode gemessen, da als Bauteil eine Induktivität nicht ausgeschlossen werden kann. Wegen der kurzen Strompulse können mit der ESR-Methode keine Induktivitäten gemessen werden. Weil nur mit der ESR-Methode eine Auflösung von 0.01Ω erreicht wird, beträgt ohne diese Option auch für Widerstände unter 10Ω die Auflösung nur 0.1Ω . Wenn diese Option gesetzt ist, gelten diese Einschränkungen nicht, die

Messung kann aber länger dauern.

Beispiel: `CFLAGS += -DRMETER_WITH_L`

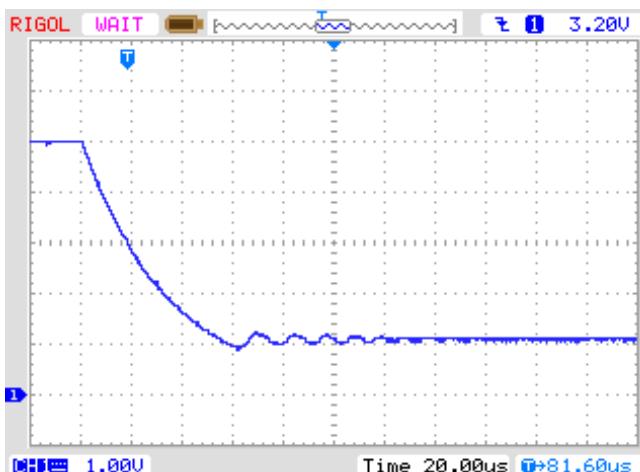
AUTOSCALE_ADC schaltet die automatische Bereichswahl des ADC (entweder VCC oder interne Referenz) ein. Die interne Referenz hat 2,56V für den ATmega8 und 1,1V für die anderen Prozessoren. Beim ATmega8 wird die automatische Bereichswahl nicht mehr benutzt.
Beispiel: `CFLAGS += -DAUTOSCALE_ADC`

ESR_ZERO gibt einen Nullwert für die ESR-Messung von Kondensatoren vor. Der vorgegebene Nullwert wird durch die beim Selbsttest ermittelten Nullwerte für alle drei Pinkombinationen ersetzt. Diese Nullwerte werden von den ermittelten Messwerten abgezogen.

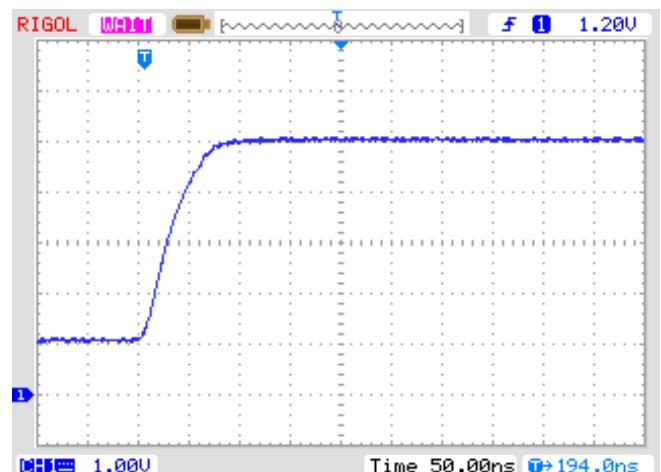
Beispiel: `CFLAGS += -DESR_ZERO=29`

NO_AREF_CAP teilt der Software mit, dass Sie keinen Kondensator am AREF Pin (Pin 21) angeschlossen haben. Dies ermöglicht kürzere Wartezeiten für die AUTOSCALE_ADC Umschaltung des ADC. Ein $1nF$ Kondensator wurde in diesem Modus ohne Fehler getestet. Die Abbildungen 4.1a und 4.1b zeigen die Schaltzeiten mit einem $1nF$ Kondensator. Wie Sie sehen können ist das Schalten von 5V auf 1,1V viel langsamer als das Zurückschalten auf 5V. Wenn Sie noch einen $100nF$ installiert haben, ist die Schaltzeit etwa Faktor 100 länger!

Beispiel: `CFLAGS += -DNO_AREF_CAP`



(a) from 5V to 1.1V



(b) from 1.1V to 5V

Abbildung 4.1. Umschalten von AREF mit einem $1nF$ Kondensator

REF_R_KORR gibt einen Offset für die interne Referenz-Spannung in mV-Einheiten an. Mit diesem Offset kann eine Differenz bei der Umschaltung der Referenzspannung für die Widerstandsmessung abgeglichen werden. Wenn die AUTO_CAL-Option gewählt wurde, ist dieser Wert nur ein Offset zu der gefundenen Spannungs-Differenz in der AUTO_CAL Funktion.
Beispiel: `CFLAGS += -DREF_R_KORR=10`

OP_MHZ gibt der Software an, mit welcher Taktfrequenz in MHz der Tester arbeiten wird. Die Software ist nur mit $1MHz$, $8MHz$ und zusätzlich auch $16MHz$ getestet. Der Betrieb mit $8MHz$ wird wegen der besseren Auflösung der Kondensator- und Spulen-Messung empfohlen.
Beispiel: `OP_MHZ = 8`

RESTART_DELAY_TICS muss auf 6 gesetzt werden, wenn der ATmega168 oder ATmega328 ohne Quarz mit dem RC-Generator betrieben wird. Wenn dieser Wert nicht vorbesetzt wird, wählt die Software die 16384 Takte Startverzögerung für den Quarzbetrieb.

Beispiel: `CFLAGS += -DRESTART_DELAY_TICS = 6`

USE_EEPROM gibt an, ob feste Texte und Tabellen im EEPROM-Speicher abgelegt werden sollen. Andernfalls wird der Programmspeicher (Flash) benutzt. Es wird empfohlen, den EEPROM-Speicher zu benutzen (Option gesetzt).

Beispiel: `CFLAGS += -DUSE_EEPROM`

EBC_STYLE gibt an, dass die Ausgabe der Transistor-Pinbelegung im Format „EBC=...“ bzw. „GDS=...“ erfolgen soll. Diese Darstellungsweise spart Programmplatz. Ohne diese Option wird die Belegung im Format „123=...“ angezeigt, wobei jeder Punkt ein E (Emitter), B (Basis) oder K (Kollektor) sein kann. Bei FETs kann jeder Punkt entsprechend ein G (Gate), D (Drain) oder S (Source) sein. Wenn die Reihenfolge der Testpins nicht 1,2 und 3 in Leserichtung ist, kann die Reihenfolge mit der Option `EBC_STYLE=321` umgedreht werden. Dann wird die Pinbelegung in der Form „321=...“, was der gewohnten Leserichtung von links nach rechts entgegen kommt, wenn die Testpins die Reihenfolge 3,2,1 haben.

Beispiel: `CFLAGS += EBC_STYLE`

NO_NANO gibt an, dass der Dezimalpräfix Nano nicht zur Darstellung von Messergebnissen benutzt werden soll. So werden Kapazitätswerte in μF statt in nF angegeben.

Beispiel: `CFLAGS += NO_NANO`

NO_LONG_PINLAYOUT kann gesetzt werden, um die lange Form der Pinbelegung bei grafischen Displays zu verhindern wie „Pin 1=E 2=B 3=C“. Wenn die Option gesetzt ist, wird die kurze Form wie „Pin 123=EBC“ benutzt.

Example: `CFLAGS += NO_LONG_PINLAYOUT`

PULLUP_DISABLE gibt an, dass man die internen „Pull-Up“-Widerstände nicht benötigt. Sie müssen einen externen „Pull-Up“ Widerstand an Pin 13 (PD7) und VCC angeschlossen haben, um diese Option benutzen zu können. Mit dieser Option wird ein möglicher Einfluss der „Pull-Up“ Widerstände auf die Mess-Ports (Port B und Port C) verhindert.

Beispiel: `CFLAGS += -DPULLUP_DISABLE`

ANZ_MESS diese Option gibt an, wie oft der ADC-Wert eingelesen und addiert werden soll. Sie können einen Wert zwischen 5 und 200 wählen um einen Mittelwert für eine ADC-Messung zu bilden. Höhere Werte ergeben eine bessere Genauigkeit, aber brauchen längere Messzeit. Eine ADC-Messung mit dem Wert 44 braucht etwa 5ms.

Beispiel: `CFLAGS += -DANZ_MESS=44`

POWER_OFF Diese Option schaltet die automatische Abschaltfunktion ein. Wenn Sie diese Option weglassen, werden die Messungen in einer Schleife endlos wiederholt, bis die Betriebsspannung unterbrochen wird (Ein/Aus-Schalter). Wenn Sie einen Tester ohne die Schalttransistoren haben, können Sie diese Option weglassen.

Wenn Sie mit den eingebauten Schalttransistoren die Option `POWER_OFF` weggelassen haben, gibt es dennoch eine Möglichkeit für eine Abschaltung, wenn Sie die `WITH_MENU`-Option gewählt haben.

Sie können mit der `POWER_OFF`-Option auch angeben, nach wie vielen Messungen ohne gefundenes Bauteil der Tester ausschaltet. Bei doppelt so viel aufeinanderfolgenden Messungen mit gefundenem Bauteil schaltet der Tester auch ab, wenn nicht zwischendurch eine Messung ohne gefundenes Bauteil war. Wenn Sie vergessen haben, ein angeschlossenes Bauteil abzuklemmen, wird so eine vollständige Batterie-Entladung verhindert. Bei einer Options-Angabe in der Form von `CFLAGS += -DPOWER_OFF=5` wird nach 5 aufeinanderfolgenden Messungen ohne gefundenes Bauteil abschaltet. Aufeinanderfolgende 10 Messungen mit gefundenem Bauteil schalten ebenfalls aus. Nur wenn die jeweilige Mess-Serie durch den anderen Typ unterbrochen wird, wird die Messung fortgesetzt. Die Messresultate für eine Einzelmessung werden 28 Sekunden angezeigt, bei der Mehrfachmessung wird die Anzeigezeit auf 5 Sekunden reduziert (wird in config.h gesetzt). Wenn der Startknopf beim ersten Einschalten lange gedrückt wird, wird das

Messergebnis auch bei der Mehrfachmessung 28 Sekunden angezeigt. Der Maximalwert für die Wiederholungen ist 255 (CFLAGS += -DPOWER_OFF=255).

Beispiel 1: CFLAGS += -DPOWER_OFF=5

Beispiel 2: CFLAGS += -DPOWER_OFF

BAT_CHECK schaltet die Batterie-Spannungsprüfung ein. Wenn Sie diese Option nicht angeben, wird die Versionsnummer der Software angezeigt. Diese Option ist hilfreich um bei batteriebetriebenen Tester-Versionen an den Batteriewechsel zu erinnern.

Beispiel: CFLAGS += -DBAT_CHECK

BAT_OUT schaltet die Batterie-Spannungsanzeige auf dem LCD ein, wenn BAT_CHECK gewählt wurde. Wenn Ihre 9V-Versorgung eine Diode als Verpolungsschutz installiert hat, können Sie die Form BAT_OUT=600 angeben, um die Dioden-Schwellspannung bei der Spannungsanzeige zu berücksichtigen. Auch der Spannungsverlust am Transistor T3 kann so mit dieser Option berücksichtigt werden. Die Angabe der Schwellspannung in mV beeinflusst nicht die Prüfspannungs Werte (BAT_POOR).

Beispiel 1: CFLAGS += -DBAT_OUT=300

Beispiel 2: CFLAGS += -DBAT_OUT

BAT_POOR setzt die Leer-Spannung für die Batteriespannungs-Prüfung auf den angegebenen Wert in Einheiten von 1mV. Die Warn-Spannung ist 0,8V höher als die angegebene Leer-Spannung, wenn die Leer-Spannung mehr als 5,3V beträgt. Sonst wird eine 0,4V höhere Warn-Spannung gewählt, bei unter 3,25V sogar nur eine 0,2V höhere Warn-Spannung und bei unter 1,3V nur eine 0,1V höhere Warnspannung als die angegebene Leer-Spannung. Das Setzen der Leer-Spannung auf Werte wie 5,4V wird für wiederaufladbare 9V Batterien nicht empfohlen, weil das die Gefahr von Batterie-Schäden aufgrund von Tiefentladung erhöht! Wenn Sie wiederaufladbare 9V-Batterien einsetzen, werden „Ready to Use“-Typen wegen der geringeren Selbstentladung empfohlen.

Beispiel für low-drop-Regler (5,4V): CFLAGS += -DBAT_POOR=5400

Beispiel für 7805-Regler (6,4V): CFLAGS += -DBAT_POOR=6400

DC_PWR Dieser Spannungspegel in mV Einheiten gibt eine Grenze für die Batteriespannung an, oberhalb derer der Tester in den „DC_Pwr_Mode“ wechselt. Normalerweise läuft der Tester in einem Batterie-Modus, wo alle Zusatzfunktionen zeitlich beschränkt laufen. Mit dem „DC_Pwr_Mode“ laufen die Zusatzfunktionen zeitlich unbeschränkt. Weil es keinen DC-DC Konverter gibt, der mit einer Eingangsspannung von 0.9V läuft, wird der „DC_Pwr_Mode“ auch gestartet, wenn eine Batteriespannung unterhalb 0.9V entdeckt wird.

Beispiel: CFLAGS += -DDC_PWR=9500

BAT_NUMERATOR Definiert den Zähler eines Bruchs an, mit dem die Spannung bewertet werden muß, um die richtige Batteriespannung zu erhalten. Für den Standard-Spannungsteiler mit einem $10k\Omega$ und einem $3.3k\Omega$ Widerstand ist der Quotient $(10000 + 3300)/3300$. Der über die Widerstandswerte erhaltene Quotient sollte gekürzt werden. Für das Beispiel ergibt sich $133/33$.

Beispiel: CFLAGS += -DBAT_NUMERATOR=133

BAT_DENOMINATOR Gibt den Nenner eines Bruches an, mit dem die Spannung bewertet werden muß.

Beispiel: CFLAGS += -DBAT_DENOMINATOR=33

EXT_NUMERATOR Definiert den Zähler eines Bruchs an, mit dem die externe Spannung bewertet werden muß, um die richtige Spannung zu erhalten. Für den Standard-Spannungsteiler mit einem $180k\Omega$ und einem $20k\Omega$ Widerstand ist der Quotient $(180000+20000)/20000$. Der Quotient sollte auf 10/1 gekürzt werden.

Beispiel: CFLAGS += -DEXT_NUMERATOR=10

EXT_DENOMINATOR Gibt den Nenner eines Bruches an, mit dem die externe Spannung bewertet werden muß.

Beispiel: CFLAGS += -DEXT_DENOMINATOR=1

INHIBIT_SLEEP_MODE sperrt die Benutzung des „Sleep Mode“ (Schlafzustand) des Prozessors. Normalerweise wird von der Software für längere Pausen der Schlafzustand des Prozessors benutzt, um Strom zu sparen. Die Benutzung dieses Schlafzustandes mit dem Wiederaufwachen spart zwar Batteriekapazität, stellt eine zusätzliche Anforderung für den Spannungsregler dar. Beispiel: DINHIBIT_SLEEP_MODE = 1

PROGRAMMER stellt den Programmer-Typ für das `avrdude` Schnittstellenprogramm ein.

Eine richtige Einstellung des Programmer-Typs (und Ports) ist notwendig.

In `Makefile` ist standardmäßig Programmer der Firma Diamex eingestellt.

Vorbereitet sind aber auch USBasp von Fischler und Arduino Mega.

Falls ein anderer Programmer verwendet sein sollte, muss er in `Makefile` aufgenommen werden und der, bis jetzt aktuelle, mit # am Anfang der Zeile abgewählt werden.

Ein Beispiel für die Benutzung von USBtiny Programmer:

```
# setting for USBtiny ISP  
PROGRAMMER=usbtiny
```

BitClock=10

PORT=usb

und ein weiterer Beispiel:

```
# setting for Pololu programmer  
# PROGRAMMER=stk500v2  
# BitClock=1.0  
# PORT = /dev/ttyACM0
```

, wenn Sie den `make upload` - oder `make fuses` -Aufruf dieser `Makefile` benutzen. Beispiel:
PROGRAMMER=avrisp2

BitClock stellt die Bit-Taktperiode für den Programmer ein. Siehe dazu die Beschreibung des -B Parameters von `avrdude`.

Beispiel: BitClock=5.0

PORT stellt die verwendete Schnittstelle ein, wo `avrdude` den Mikrocontroller (ATmega) erreichen kann. Für weitere Informationen schauen Sie bitte in das Handbuch von `avrdude` oder in die Online-Dokumentation [14].

Beispiel: PORT=usb

Zusätzliche Parameter können in den Dateien TransistorTester.h und config.h gesetzt werden. Die Datei config.h enthält globale Variablen und Tabellen, definiert die Port- / Pin-Konstellation, die ADC-Taktfrequenz sowie die Widerstandswerte, die für die Messung benutzt werden. Die Datei TransistorTester.h enthält die globalen Variablen und Tabellen sowie die Texte für die LCD-Anzeige. Normalerweise brauchen diese Werte nicht ohne Grund geändert werden.

4.2 Programmierung des Mikrocontrollers

Ich gebe die Software für den Mikrocontroller in Quelltext heraus. Die Entwicklung wurde mit dem Linux-Betriebssystem (Ubuntu bzw. Mint) gemacht und wird gesteuert mit einer `Makefile`.

Die `Makefile` stellt sicher, dass die Software entsprechend der vorher in der `Makefile` eingestellten Optionen übersetzt wird. Schauen Sie bitte in die Datei LiesMich.txt im Verzeichnis default und in das Konfigurations-Kapitel 4.1 ab Seite 49.

Das Ergebnis der Übersetzung hat die Dateierweiterung `.hex` und `.EEP`. Die `.hex`-Datei enthält die Daten für den Programmspeicher (Flash) des ATmega-Prozessors. Die `.EEP`-Datei enthält die Daten

für den EEPROM des ATmega.

Beide Dateien müssen in den richtigen Speicher geladen werden.

Zusätzlich muss der ATmega mit den „fuses“ richtig konfiguriert werden.

Wenn Sie meine Makefile zusammen mit dem Programm avrdude [14] benutzen, brauchen Sie keine genaue Kenntnis über die Einzelheiten der fuses.

Sie brauchen nur make fuses aufrufen, wenn Sie keinen Quarz benutzen oder Sie müssen make fuses-crystal aufrufen, wenn Sie einen 8MHz Quarz auf der Baugruppe installiert haben.

Bei der ATmega168-Serie der Mikrocontroller können Sie alternativ auch make fuses-crystal-lp aufrufen für den low power Quarz-Betrieb.

Benutzen Sie niemals die Quarz-Varianten, wenn Sie keinen 8MHz oder 16MHz Quarz installiert haben.

Wenn Sie sich nicht sicher mit den fuses sind, lassen Sie diese erst einmal wie vom Werk gesetzt und bringen Sie den Tester in diesem Zustand zum Laufen.

Es kann sein, dass das Programm zu langsam läuft, wenn Sie die für den 8MHz-Betrieb erzeugten Programmdaten benutzen, aber das kann man später korrigieren! Aber falsch gesetzte fuses können die spätere ISP-Programmierung verhindern.

Vielleicht meldet das Programm avrdude einen Fehler beim Setzen der extendet Fuse efuse. Das Lesen von unbgenutzten Fuse Bits ist beim ATmega als „1“ spezifiziert, aber das avrdude Programm maskiert die unbgenutzten Bits, so daß es eine „0“ für alle unbgenutzten Bits erwartet. Normalerweise sollte die efuse auf 0xfc gesetzt werden, aber avrdude liest eine 0x04 mit der Maske zurück. Man kann die Datei avrdude.conf ändern, um das Verhalten von avrdude zu ändern oder die efuse auf 0x04 setzen. Alle efuse-Werte können mit dem Bezeichner EFUSE_VAL am Anfang der Datei setup.mk im Quellcode-Verzeichnis gesetzt werden. Wahrscheinlich sind die Fuses aber auch mit der Fehlermeldung richtig gesetzt.

4.2.1 Benutzung von Linux

Um meine Erfahrung mit Verzweiflung und „schlaflose Nächten“ anderen Kollegen zu ersparen, wurde dieses Unterkapitel geschrieben. Ich hatte ohne jegliche AVR Erfahrung einen Clonetester erworben und wollte diesem die deutsche Sprache „beibringen“.

Die dabei erworbenen Erfahrungen sollen anderen „willigen“ Unerfahrenen helfen, ERFOLGREICH ihren Tester zu programmieren. Diese Gelegenheit wird benutzt, dem Entwickler des Transistortester und Autor dieses Dokuments, Karl-Heinz Kübbeler siehe [16] zu danken für sein Engagement und Geduld, denn die folgenden Seiten wären ohne seine Hilfe nie entstanden.

Damit das Übersetzen der Firmware und Brennen ins MCU gelingt und gleichzeitig . . . „das Rad nicht neu erfunden werden muss“, wurde ein Teil der folgenden Seiten aus dem ursprünglichen Original übernommen.

Also noch einmal . . . **EINEN RIESEN DANK** an Karl-Heinz Kübbeler.

Betriebssystem Linux

Die Programmierung unten Linux bringt viele Vorteile, weil dieses OS von Experten entwickelt wurde, die sich auf den Wünschen der Benutzer orientieren. Zudem ist die Umgebung kostenlos erhältlich und perfekt gewartet. Ein weiterer Vorteil ist die Sicherheit des OS selbst aber auch beim Nutzen des Internets. Die heutigen Editionen lassen sich noch viel leichter bedienen als die Mitbewerber OS. Es gibt auch sehr leistungsfähige Editoren wie vim oder emacs, die aber etwas Einarbeitungszeit erfordern. Speziell vim hat den Vorteil, daß er praktisch auf jedem System vorinstalliert ist. Aber gerade bei diesem Editor ist der Wechsel zwischen Eingabe-Modus und Befehls-Modus gewöhnungsbedürfig. Diese Anleitung soll alle „nicht“ Linux Benutzer dazu animieren es NUN zu testen indem sie ihr Tester damit programmieren.

Die nachfolgenden Beispiele wurden mit Linux Mint getestet, das mit drei verschiedenen Desktop Umgebungen angeboten wird (cinnamon, MATE und Xfce). Die nachfolgenden Hinweise sollten aber bei allen drei Desktops funktionieren. Die Installation ist auf verschiedene Arten möglich und bringt seinen eigenen Bootmanager mit, damit man sein vorhandenes OS weiter parallel nutzen kann.

Tipps für die Linux Nutzung

Zuerst möchte ich einen Hinweis geben für alle, die nicht gerne Texte abschreiben. Sie können dieses Handbuch auf einen USB-Stick kopieren und im Dateibrowser mit Doppelclick der linken Maustaste **[● ○ ○]** öffnen. Alternativ zum Doppelklick kann mit der **[○ ○ ●]** Taste ein anderes als das vorgewählte PDF Betrachtungsprogramm ausgewählt werden. Das geöffnete Fenster kann auf dem Bildschirm verschoben werden und auch in der Größe angepasst werden. Dazu gibt es in der Regel verschiedene Möglichkeiten. Eine davon ist die **[○ ○ ●]** Taste zu drücken, wenn der Mauszeiger auf die Kopfzeile des Fensters zeigt und dann die Funktion **Verschieben** auszuwählen. So kann man das Fenster mit der PDF-Dokumentation beispielsweise an die linke Bildschirmseite verschieben. Durch Drücken der linken Maustaste rastet das Fenster an der neuen Position ein. Zum Verschieben kann man auch mit der **[● ○ ○]** Taste die Kopfleiste des Fensters packen und bei weiter gedrückter **[● ○ ○]** Taste verschieben. Dann rastet das Fenster auf der Position ein, wenn man die Maustaste loslässt.

Im nächsten Schritt wird die Tastenkombination **[Strg]+[Alt]+[T]** gleichzeitig gedrückt, um ein (neues) Befehlsfenster zu öffnen. Dies wird nun auf die schon beschriebene Weise zur rechten Hälfte des Bildschirmes verschoben und kann auch in der Größe verändert werden. Die Größenänderung kann mit der **[● ○ ○]** Taste an den Rändern oder Ecken des Fenstern gemacht werden, oder auch mit der **[○ ○ ●]** Taste an der Kopfleiste mit der **Größe ändern** Funktion. Die Bedienung ist sonst gleich wie beim Verschieben.

In der Regel besitzen die graphischen Oberflächen bei Linux mehr als eine Arbeitsfläche, die mit der Tastenkombination **Strg**+**Alt**+→ beziehungsweise **Strg**+**Alt**+← umgeschaltet werden können. Mit der ⎵ Taste können Sie auf der Kopfzeile der Fenster eine der Arbeitsflächen wählen, auf der das Fenster angezeigt wird.

So kann man die für die Arbeit benötigten Fenster auf einer Arbeitfläche zusammen sammeln und erstickt nicht in all den Fenstern für verschiedene Arbeitsbereiche. Bei allen Befehlseingaben und Dateibezeichnern ist übrigens darauf zu achten, daß bei Linux zwischen Groß und Kleinschreibung unterschieden wird.

Programm Pakete installieren

Zum Installieren von Software-Paketen braucht ihr Rechner einen Internetzugang. Bevor Sie den Tester programmieren können, müssen zuerst die Programmpakete binutils-avr, avrdude, avr-libc und gcc-avr installiert werden. Jetzt können sie den unten angegebenen Befehl durch Abtippen mit der Tastatur ausführen lassen. Sie können aber auch im geöffneten PDF Dokument an diese Stelle blättern und den nachfolgenden Text mit der gedückten linken Maustaste ⎵ markieren:

```
sudo apt-get install avrdude avr-libc binutils-avr gcc-avr git
```

Am Ende des Textes muß die Maustaste ⎵ losgelassen werden, um die Markierung abzuschließen. Wenn der Text in einer eigenen Zeile steht, wie in diesem Beispiel, können Sie den Text auch mit dreimaligem ⎵ Maus Click irgendwo auf der Zeile markieren. Danach kann man den Mauszeiger in das rechte Befehlsfenster führen und durch Drücken der mittleren Maustaste weiter als ⎵ abgekürzt den vorhin markierten Text in die Befehlszeile einfügen. Bei vielen Mäusen ist das Scroll-Rad gleichzeitig die mittlere Maustaste. Bei Mäusen ohne mittlere Maustaste ist es möglich die mittlere Maustaste durch gleichzeitiges Drücken der ⎵ Maustasten zu ersetzen. Egal, wie der Befehlstext nun in die Befehlszeile gekommen ist, Sie sollten den Text noch einmal kontrollieren, bevor Sie den Befehl durch Drücken der **Enter** Taste abschicken. In der Regel brauchen Sie keine Angst zu haben, daß durch die Installation von Paketen etwas schlimmes durch Fehlbedienung passiert. Das Programm **apt-get** prüft vor dem Ausführen der Operation, ob die aufgeführten Pakete schon installiert sind und die Abhängigkeiten erfüllt sind. Gegebenenfalls wird aber ein schon vorher installiertes Paket durch ein neueres ersetzt. Dann wird das **sudo** Programm Sie zunächst nach dem Benutzer Passwort fragen, bevor es den Rest der Befehlszeile ausführt. Sie sollten das Passwort mit Drücken der **Enter** oder ↲ Taste abschließen und bestätigen. Damit werden nun alle Software Pakete durch **apt-get** herunter geladen und installiert.

Es kann sein, daß **apt-get** bei der Installation der Pakete Fragen stellt, die man in der Regel mit der Taste **J** beantworten kann.

Natürlich gibt es auch andere Wege, die Pakete zu installieren, welche eine graphische Oberfläche benutzen wie **synaptic** oder **dpkg**. Aber es nicht einfacher eine gemischte Gruppe von Paketen anzugeben. Wenn Sie irgendeine dieser Paketmanager, sollten Sie über alle installierten Pakete Bescheid wissen, egal auf welchem Wege Sie installiert wurden. Die graphischen Paketmanager können auch dabei helfen, den Namen der Pakete herauszufinden.

Download der Quellen

Ob die Versionsverwaltung **git** erfolgreich installiert wurde, kann man mit dem folgenden Kommando prüfen:

```
git version
```

Das `git` Programm sollte mit der Ausgabe seiner Versionsnummer antworten. Falls in Ihrem Heimatverzeichnis schon ein Ordner `TransistorTester-source` existiert, sollte man den umbenennen oder löschen. Das Programm `git` wird für den Download der Quellen und Dokumentation benutzt. Mit dem Befehl:

```
git clone https://github.com/kubi48/TransistorTester-source
```

wird das aktuelle Transistortester Quellcode-Archiv heruntergeladen. Die Dateien sind nun in dem Linux [Persönlicher Ordner] , üblicherweise `/home/` gefolgt von Ihrem Benutzerkürzel, mit dem Verzeichnisnamen `TransistorTester-source`. Mit der Terminaleingabe `ls [l] [s] [d]` kann man das Vorhandensein überprüfen. Mehr Informationen über Dateien und Verzeichnisse erhält man mit dem Kommando `ls -lh`. Hier wurden jetzt zwei Optionen für das Kommando `ls` zusammengefasst, die `-l` Option und die `-h` Option. Diese Eingabeform ist also eine Kurzform für das Kommando `ls -l -h` oder auch `ls -l --human-readable`. Einige der Optionen sind in zwei Versionen vorhanden, einer kurzen mit einem `-` (`-h`) und einer langen Form mit zwei `-` (`--human-readable`). Die Reihenfolge der Optionen spielt übrigens keine Rolle, genau wie die Anzahl der trennenden Leerzeichen . Bei fast allen Kommandos erfährt man mehr über die Bedienung und die Optionen durch Anhängen der `--help` Option. Das gilt natürlich auch für das Kommando `git`.

Um neue Updates heruntergeladen, reicht es in der Zukunft,

```
git pull
```

im Arbeitsverzeichnis `TransistorTester-source` einzugeben. Das kann je nach Kommandointerpreter auch mit

```
(cd ~/TransistorTester-source ; git pull)
```

aus jedem Arbeitsverzeichnis erfolgen. Damit der `git pull` einwandfrei funktioniert, sollte man den Verzeichnisbaum besser nicht verändern.

Übersetzen der Transistortester Quellen

Zum Übersetzen der Transistortester-Quellen ist nur ein `make` Aufruf im richtigen Arbeitsverzeichnis erforderlich. Da aber auch mit `make upload` der ATmega mit einem angeschlossenen ISP-Programmer programmiert werden kann, ist es sinnvoll, sich zunächst mit den Schnittstellen für den Anschluß des ISP-Programmer zu beschäftigen. Der weitere Arbeitsablauf ist dann unter Punkt 4.2.1 auf Seite 67 beschrieben.

Benutzung der Schnittstellen

Alle modernen ISP-Programmer mit serieller Schnittstelle benutzen gerne die USB Schnittstelle, da diese Schnittstelle auch gleich die Stromversorgung sicherstellt. Für diese Geräte sollten Sie überprüfen, welche Gerätbezeichnung diesem Gerät zugeordnet ist. Beim Einsticken eines USB Gerätes wird bei Linux ein Eintrag in das Systemprotokoll vorgenommen. Da das Systemprotokoll eine Textdatei ist, kann man diese einfach auf dem Bildschirm anzeigen. Dazu kann man in Konsolenfenster den Befehl `dmesg` benutzen und da wir nach dem Einsticken des Programmers nur an den letzten Zeilen des Protokolls interessiert sind, benutzt man:

```
dmesg|tail
```

Das `dmesg` Programm zeigt das gesamte Systemprotokoll und das Kommando `tail` zeigt nur die letzten 10 Zeilen der Ausgabe. Für einen Pololu Programmer sieht das Ergebnis so aus:

```
usb 1-3: new full-speed USB device number 3 using xhci_hcd
usb 1-3: New USB device found, idVendor=1ffb, idProduct=00bb, bcdDevice= 1.02
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: Pololu USB AVR Programmer v2.1
usb 1-3: Manufacturer: Pololu Corporation
usb 1-3: SerialNumber: 00227484
cdc_acm 1-3:1.1: ttyACM0: USB ACM device
cdc_acm 1-3:1.3: ttyACM1: USB ACM device
usbcore: registered new interface driver cdc_acm
cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
```

Wichtig sind hier die Zeilen 7 und 8 mit den Einträgen `ttyACM0` und `ttyACM1`. Das sind die zugewiesenen Gerätbezeichnungen für zwei serielle Schnittstellen. Bei Linux sind alle Geräte auch Bestandteil des Dateibaums und im Ordner `/dev/` eingetragen. Mit vollständigem Namen heißen die beiden seriellen Schnittstellen also `/dev/ttyACM0` und `/dev/ttyACM1`. Für den Pololu Programmer muß man wissen, daß die erste serielle Schnittstelle für den ISP-Programmer benutzt wird und die zweite serielle Schnittstelle für andere Zwecke frei benutzt werden kann. Sie können die Existenz der Geräteeintragungen mit dem Kommando

```
ls -l /dev/ttyACM*
```

überprüfen. Das Ergebnis sollte dann so aussehen:

```
crw-rw---- 1 root dialout 166, 0 Mär 11 09:57 /dev/ttyACM0
crw-rw---- 1 root dialout 166, 1 Mär 11 09:57 /dev/ttyACM1
```

Bei dieser Ausgabe kann man lesen, daß der Zugriff für den Benutzer `root` und für die Benutzergruppe `dialout` erlaubt ist. Mit dem Kommando `id` können Sie ihre Gruppenzugehörigkeit überprüfen. Hier sollte die Gruppe `dialout` in der Liste auftauchen, sonst wird im nächsten Arbeitspunkt die Änderung der Gruppenzugehörigkeit beschrieben. Ein anderes Beispiel für das Systemprotokoll eines anderen ISP-Programmers „Diamex ISP-PRoG NG“ sehen sie hier:

```
usb 1-6: new full-speed USB device number 8 using xhci_hcd
usb 1-6: New USB device found, idVendor=16c0, idProduct=2a9b, bcdDevice=43.40
usb 1-6: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-6: Product: AVR-ISP2
usb 1-6: Manufacturer: ERFOS
usb 1-6: SerialNumber: 19377-43111-757
cdc_acm 1-6:1.0: ttyACM0: USB ACM device
```

Bei diesem Beispiel ist der Gerätename identisch (`ttyACM0`).

Einen Überblick über die angeschlossenen USB-seriell Wandler geben auch die Eintragungen in Verzeichnis `/dev/serial/by-id/`, die man mit dem Kommando abfragen kann:

```
ls -og /dev/serial/by-id/* | cut -d' ' -f 7-
```

Die Ausgabe kann beispielsweise so aussehen:

```
/dev/serial/by-id/usb-Arduino_www.arduino.cc_0043_954323131383519062F0-if00 -> ../../ttyACM2
/dev/serial/by-id/usb-Pololu_Corporation_Pololu_USB_AVR_Programmer_v2.1_00227484-if01 -> ../../ttyACM0
/dev/serial/by-id/usb-Pololu_Corporation_Pololu_USB_AVR_Programmer_v2.1_00227484-if03 -> ../../ttyACM1
```

Wenn die Beschreibung ausführlich genug ist, kann man so die Schnittstellen zuordnen.

Wenn bis zu diesem Schritt alles in Ordnung ist, muß für dieses Beispiel nur der PORT Eintrag in der Makefile auf „PORT=/dev/ttyACM0“ geändert werden, damit das avrdude Programm auf den Programmer zugreifen kann. Wenn bei Ihrem System noch andere USB-seriell Schnittstellen benutzt werden, können die letzten Ziffern der Gerätebezeichnung abweichen. Erwähnen möchte ich auch, daß eine andere Gruppe von USB-seriell Schnittstellen statt ttyACM den Namen ttyUSB verwendet.

Wenn Ihr Programmer eine spezielle USB-Schnittstelle benutzt und keinen USB-seriell Typ, ist wahrscheinlich noch zusätzliche Arbeit erforderlich. Als PORT kann hier fest „usb“ in der Makefile eingetragen werden. Es ist aber wahrscheinlich, daß Sie trotzdem keinen Zugriff auf das Gerät haben. Alle angeschlossenen USB-Geräte können durch Eingabe von lsusb im Befehlsfenster angezeigt werden. Geben Sie lsusb zuerst ohne und dann mit angeschlossenem USB-Programmer ein. Ein Vergleich der Ergebnisse lokalisiert den USB-Programmer.

Das Ergebnis von lsusb kann so aussehen:

```
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 003: ID 046d:c050 Logitech, Inc. RX 250 Optical Mouse
Bus 002 Device 058: ID 03eb:2104 Atmel Corp. AVR ISP mkII
Bus 002 Device 059: ID 2341:0042 Arduino SA Mega 2560 R3 (CDC ACM)
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub}
```

Hier wurde als Device 58 ein AVR ISP mkII erkannt (DIAMEX ALL-AVR). Das ebenfalls entdeckte Gerät mit der Nummer 59 gehört zur Gruppe der „USB-seriell“ Geräte. Die ID 03eb ist eine Herstellerkennung und die ID 2104 eine Produktkennung für diesen ISP-Programmer. Diese beiden Kennungen werden der Datei /etc/udev/rules.d/90-atmel.rules benötigt und erstellt mit Hilfe von:

```
sudo xed /etc/udev/rules.d/90-atmel.rules
```

Natürlich können Sie auch einen anderen Editor als xed benutzen, wenn Sie möchten. In diesem Beispiel besteht die Datei 90-atmel.rules aus einer Zeile:

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2104", MODE="0660", GROUP="plugdev"
```

Dieser Eintrag erlaubt den Zugriff auf das Gerät für Mitglieder der Gruppe plugdev. Man kann den Eintrag auch ohne Editor direkt mit einem Kommando erzeugen:

```
sudo echo 'SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2104"
, MODE="0660", GROUP="plugdev"' >> /etc/udev/rules.d/90-atmel.rules
```

Die beiden Zeilen müssen sie zu einer langen Zeile in der Kommandozeile zusammenfügen! Um die meisten Programmer verwenden zu können, wird folgender Text in 90-atmel.rules empfohlen:

```
# Copy this file to /etc/udev/rules.d/90-atmel.rules
# AVR ISP mkII - DIAMEX ALL-AVR
SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2104", GROUP="plugdev", MODE="0660",
# Atmel AVR Dragon
ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2107", GROUP="plugdev", MODE="0660"
# Atmel-ICE
SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2141", GROUP="plugdev", MODE="0660",
# xpained-mini
SUBSYSTEM=="usb", ATTRS{idVendor}=="03eb", ATTRS{idProduct}=="2145", GROUP="plugdev", MODE="0660",
# avrftdi
#SUBSYSTEM=="usb", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6010", GROUP="plugdev", MODE="0660",
# UM232H
#SUBSYSTEM=="usb", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6014", GROUP="plugdev", MODE="0660",
# USBasp programmer
ATTRS{idVendor}=="16c0", ATTRS{idProduct}=="05dc", GROUP="plugdev", MODE="0660"
# USB NIBObee-Programmer
ATTRS{idVendor}=="16c0", ATTRS{idProduct}=="092f", GROUP="plugdev", MODE="0660"
# USBtiny programmer
ATTRS{idVendor}=="1781", ATTRS{idProduct}=="0c9f", GROUP="plugdev", MODE="0660"
# USB ISP-programmer für Atmel AVR
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", SYSFS{idVendor}=="16c0", SYSFS{idProduct}=="05dc", MODE="0660",
```

Nach dem die Datei erstellt wurde, kann die Erstellung und Inhalt kontrollieren mit:

```
less /etc/udev/rules.d/90-atmel.rules
```

Danach sollten Sie das System veranlassen, die udev-Regeln neu einzulesen mit:

```
sudo udevadm control --reload-rules
```

Dann sollten Sie den ISP-Programmer ausstecken und wieder einstecken. Jetzt sollte das System Ihnen Zugriff auf das Gerät geben, wenn Sie Mitglied der Gruppe `plugdev` sind. Daher sollten Sie Mitglied der Gruppe `plugdev` und auch der Gruppe `dialout` sein, damit Sie beide ISP-Programmertypen nutzen können.

Gruppen Mitgliedschaft

Deswegen sollte die eigene Benutzerkennung sowohl Mitglied der Gruppe `plugdev` als auch der Gruppe `dialout` sein. Das Kommando:

```
sudo usermod -a -G dialout,plugdev $USER
```

sollte die Zugehörigkeit sicherstellen. Kontrollieren kann man es mit dem Kommando: `id`. Wenn die Gruppenmitgliedschaft richtig gemeldet wird, sollte jetzt ein Zugriff mit `avrdude` auf ISP-Programmer mit beiden Schnittstellentypen möglich sein.

Mann kann sich die Gruppenmitgliedschaft auch mit einem Programm mit graphischer Bedienoberfläche ansehen: `Menü > Systemverwaltung > Benutzer und Gruppen > ?Passwort`. Das `Menü` kann man auch mit der `Esc` Taste (zwischen `Strg` und `Alt`) öffnen. Aber das Layout der Funktion ist unterschiedlich je nach Desktop Umgebung.

Arbeitsumgebung und Übersetzen der Quellen

Damit das Original erhalten bleibt, wird empfohlen, ein Duplikat der Quellen mit Namen `Mytester` anzulegen. Üblicherweise heißt das Home-Verzeichnis `/home/` gefolgt von ihrem Benutzerkürzel. Der Name ihres Home-Verzeichnisses ist in der System-Variablen `$HOME` abgelegt. Sie können statt des Namens in Befehlen auch kurz `~/` schreiben. Vergessen sie nicht den `/` nach dem `~` Zeichen, sonst würde der Kommandointerpreter nach einem Benutzer mit dem nachfolgenden Namen suchen! Dazu legen Sie erst einmal ein leeres Verzeichnis an mit:

```
mkdir ~/Mytester
```

Wenn Sie den Download vom Archiv in das Verzeichnis `~/TransistorTester-source` durchgeführt haben, können Sie den nachfolgenden Befehl ausführen, um die Quelldateien und dessen Unterverzeichnisse in das Mytester Verzeichnis zu kopieren.

```
rsync -auv ~/TransistorTester-source/trunk ./Mytester
```

Wegen der `-v` Option protokolliert `rsync` alle Kopiervorgänge. Wenn Sie statt der `-auv` Optionen also `-au` als Optionen angeben, bleibt der Kopiervorgang stumm. Sie können sich aber vom gefüllten Unterverzeichnis mit folgendem Befehl überzeugen:

```
ls -lh ~/Mytester/trunk
```

Eine übersichtliche Darstellung der Verzeichnisstruktur und der Dateien ist mit dem Kommando

```
tree ~/Mytester
```

möglich. Das Kommando ist aber nicht als Standard installiert, was sie aber mit

```
sudo apt-get install tree
```

leicht nachholen können.

Wenn Sie jetzt wissen, welches Unterverzeichnis für Ihren Tester paßt, können Sie in dieses Unterverzeichnis wechseln. Nehmen wir mal an, Sie besäßen einen chinesischen Bausatz mit SW-Display, dann wäre das Unterverzeichnis `mega328_st7565_kit` richtig. Der Befehl dazu wäre dann:

```
cd ~/Mytester/mega328_st7565_kit
```

Mit einem simplen `make` Aufruf würde der Quellcode mit den voreingestellten Einstellungen in der `Makefile` neu übersetzt.

Methode mit grafische Oberfläche

Wenn das MyTester Verzeichnis schon angelegt ist, können Sie auch mit dem Dateien Fenster in das gewünschte Verzeichnis wechseln. Sie können die Dateien in `~/Mytester/` auch mit dem Dateien Fenster ansehen **Menü > Zubehör > Dateien**, dabei muß das Verzeichnis `Mytester` durch **•○○** Doppelclick ausgewählt werden. Hier erscheinen jetzt viele Unterverzeichnisse, darunter auch beispielsweise `mega328_st7565_kit`. Wenn Sie dieses jetzt auch dieses Verzeichnis mit **•○○** Doppelclick auswählen, sehen Sie unter den Files auch die `Makefile`. Wenn Sie das geschafft haben, können sie jetzt mit der **○○●** Taste die Funktion **Im Terminal öffnen** auswählen und schon öffnet sich ein Kommandofenster mit dem richtigen Arbeitsverzeichnis, welches auch direkt aktiv ist. Jetzt brauchen sie nur noch `make` einzutippen und schon wird das Transistortester Programm neu übersetzt. Sie haben jetzt zwei Fenster, die schon auf dem richtigen Arbeitsverzeichnis stehen. Das Dateien-Fenster und das Terminal-Fenster für die Befehls-Eingabe `make`.

Bearbeitung von Makefile

Durch **•○○** Doppelclick auf die `Makefile` Datei im Dateien-Fenster öffnet sich ein weiteres Fenster mit dem voreingestellten Editor. Wenn Sie hier einen anderen Editor bevorzugen, können Sie diesen Editor entweder direkt über **○○● Öffnen mit** auswählen. Wenn Sie immer einen anderen Editor für Text-Dateien einstellen möchten, ist das ebenfalls über **○○● Eigenschaften** möglich. Dann öffnet sich ein neues **Eigenschaften von Makefile** Fenster, wo sie die Funktion **Öffnen Mit** durch **○○●** anklicken auswählen können. Hier sehen sie jetzt die aktuelle Einstellung und eine Auswahl an Möglichkeiten zum Bearbeiten der `Makefile`. Hier können Sie durch **●○○** anklicken eine andere Anwendung auswählen. Durch **●○○** Anklicken können sie jetzt auswählen, ob sie diese Anwendung **Zur Liste hinzufügen** oder **Als Vorgabe festlegen** möchten. Sie können die Einstellung aber auch wieder **Auf Systemvorgabe zurücksetzen**. Für den nächsten Schritt ist es wichtig, daß die Einstellungen für **ihren ISP-Programmer** in der `Makefile` richtig eingestellt sind. Siehe dazu im Unterkapitel 4.1, auf der Seite 60, Thema **PROGRAMMER** und **PORT**. Nach dem sie diese Einstellungen überprüft haben, speichern sie die Makefile. Vor dem Eintippen eines Kommandos muß das Terminal-Fenster durch einen **●○○** Klick aktiviert werden. Ein aktiviertes Fenster wird durch einen höheren Kontrast in der Kopfzeile angezeigt. Sie haben jetzt zwei Fenster, die schon auf dem richtigen Arbeitsverzeichnis stehen. Das Dateien-Fenster und das Terminal-Fenster für die Befehls-Eingabe `make`. Natürlich läßt sich ein Editor zum Bearbeiten der `Makefile` auch direkt aus dem Terminal-Fenster aufrufen. Dann wäre das Dateienfenster nicht erforderlich.

Endlich ist es so weit:

Tester programmieren

Wenn Sie das geschafft haben, und ihr ISP-Programmer komplett angeschlossen ist, also eine Verbindung zum Rechner und zum ATmega hat, brauchen sie nur das Kommando eingeben:

```
make upload
```

Das Transistortester Programm wird jetzt noch einmal neu übersetzt und dann sofort mit dem Programm `avrdude` in den ATmega geladen. Falls der Erfolg nicht so aussieht, wie soll, können sie nun sofort zum `Makefile` Editor wechseln, nötige Änderungen machen und den Prozess wiederholen.

Tipps zum Terminal

Sie brauchen die Befehle nicht jedes mal neu eintippen. Bereits abgegebene Befehle in dem Terminal Fenster lassen sich mit der Taste wieder anzeigen. Den angezeigten Befehl können sie durch Drücken der `Enter` oder Taste wiederholen. Sie können aber auch der angezeigten Befehl vor dem Abschicken editieren, oder mit der wieder zu neueren Befehlen wechseln.

Mögliche Makefile Aufrufe

Zum Schluß sind hier noch einmal die wichtigsten `Makefile` Aufrufe aufgelistet:

Befehl	Bedeutung
<code>make clean</code>	zum reinigen des Arbeitsumgebung
<code>make</code>	zum Übersetzen des Programms
<code>make fuses</code>	zum setzen von ATmega „fuses“ ohne Quarz
<code>make fuses-crystal</code>	zum Setzen von ATmega „fuses“ NUR mit Version mit Quarz!
<code>make upload</code>	zum Laden des kompletten Programms über die ISP Schnittstelle in den ATmega.

Hinweise zum Update der Transistortester-Quellen

Die Kopie der transistortester-Quellen kann mit dem Kommando

`(cd ~/TransistorTester-source ; git pull)`

auf dem aktuellen Stand gehalten werden. Wenn man wie hier empfohlen in einer Kopie unter `~/Mytester` arbeitet, werden die Änderungen erst übertragen, wenn man auch das Kommando

`rsync -auv ~/TransistorTester-source/trunk ~/Mytester`

ausführt. Dabei kann es passieren, daß die `Makefile` auf dem github Server ein neueres Datum hat als die hier lokal geänderte Kopie. Dann würden die hier lokal im Mytester Ordner durchgeföhrten Änderungen der Optionen in der `Makefile` verlorengehen. Daher ist es eine gute Idee, die erfolgreich geänderte `Makefile` als Kopie zu sichern. Das kann beispielsweise mit dem Befehl `cp Makefile Makefile.bak` erfolgen. Dann kann man mit dem Befehl `diff Makefile Makefile.bak` die Dateien nach einem Update vom Rechner vergleichen lassen. Ein besserer Überblick der Änderungen ist auch mit `kdiff3 Makefile Makefile.bak` möglich. Das Programm muß aber wahrscheinlich mit `sudo apt-get install kdiff3` erst installiert werden, da es eigentlich für die KDE Desktopumgebung entwickelt wurde.

4.2.2 Erstellen der Software unter Windows

Wahrscheinlich ist derzeit der einfachste Weg, zuerst die Arduino IDE (Integrierte Entwicklungsumgebung) zu installieren. Dieses Softwarepaket installiert eine aktuelle Version des avr-gcc Kompilers, der dann auch außerhalb der IDE benutzbar ist, weil der Pfad zu den Programmen in der PATH Variablen von Windows eingetragen ist. Normalerweise werden die Programme in C:\Arduino\hardware\tools\avr\bin installiert. Dort ist dann auch das avrdude Programm für die Bedienung des IDE-Programmers zu finden. Lediglich ein GNU make Programm fehlt, um die Makefile nutzen zu können. Dieses Programm habe ich mit dem Cygwin64 Paket installiert. Für den Download der Transistortester Software wird das Programm git benutzt, das ebenfalls im Cygwin64 Paket vorhanden ist und sich bei der Installation auswählen lässt. Das Cygwin64 Paket benutzt einen eigenen Installer setup-x86_64.exe, der sowohl für Neuinstallationen oder Updates als auch für Nachinstallation fehlender Tools benutzt wird. Das ehemals gesamte TransistorTester Archiv ist jetzt aufgeteilt in drei verschiedene github Archive. Die Dokumentation und Quellen alter Versionen befindet sich auf <http://github.com/kubi48/TransistorTester-old-versions>. Die aktuelle Dokumentation findet man unter <http://github.com/kubi48/TransistorTester-documentation>. Die letzten Quellen findet man unter <http://github.com/kubi48/TransistorTester-source>. In einem Kommando-Fenster kann man die Archive mit „git clone“ und der jeweiligen Adresse des github Archivs in das lokale Verzeichnis kopieren. Die Quellen von github können mit dem folgenden Kommando auf den eigenen Rechner kopiert werden:

```
git clone https://github.com/kubi48/TransistorTester-source
```

Wenn das Kommando fehlerfrei abgearbeitet wurde, sollte sich im Verzeichnis TransistorTester-source eine Kopie des letzten Quellcodes befinden. Die Quellen der hier beschriebene k-Version befinden sich im Unterverzeichnis trunk. Außerdem findet man komprimierte tar Archive (.tgz) der letzten m-Versionen im Unterverzeichnis Markus.

Der Quellcode der k-Version im Verzeichnis trunk hat mehrere Unterverzeichnisse mit je einer Makefile. Die verschiedenen Makefiles dienen dazu, Anpassungen an die verschiedenen Testermodelle vorzunehmen. Um den Namen des passenden Unterverzeichnisses zu finden können Sie die Datei picture-link.pdf im TransistorTester-source Verzeichnis mit dem Acrobat Reader öffnen. Unter den abgebildeten Testern befindet sich jeweils ein Link oder mehrere Links zu einem passenden Unterverzeichnis. Es sind aber derzeit nicht von allen Testern Fotos eingetragen. Zum Übersetzen der Quellen müssen Sie in einem Kommandofenster öffnen und in das zu Ihrem Tester passende Unterverzeichnis wechseln. Für den chinesischen TransistorTester Kit mit dem grafischen Display (ST7565 Kontroller) ist dazu die Kommandofolge erforderlich:

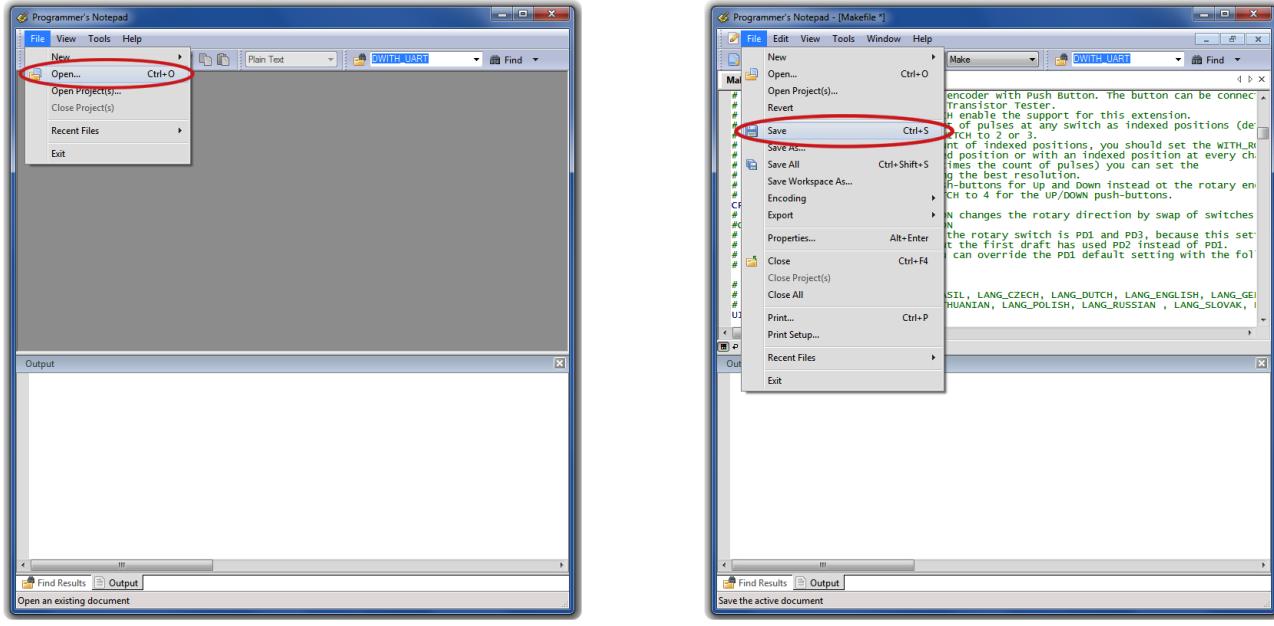
```
cd TransistorTester-source
cd trunk
cd mega328_st7565_kit
make
```

Natürlich können Sie die Optionen in der Makefile mit einem Editor Ihrer Wahl vor dem make Aufruf verändern. Wenn Sie in der Makefile die Einstellungen PROGRAMMER und PORT an Ihren ISP-Programmer angepasst haben, können Sie das übersetzte Programm mit einem „make upload“ zum angeschlossenen TransistorTester übertragen. Bei neuen AVR-Prozessoren oder wenn die richtige Einstellung der Fuses nicht sicher ist, sollten Sie die Fuses mit „make fuses-crystal“ passend einstellen.

4.2.3 Benutzung des WinAVR-Paketes unter Windows

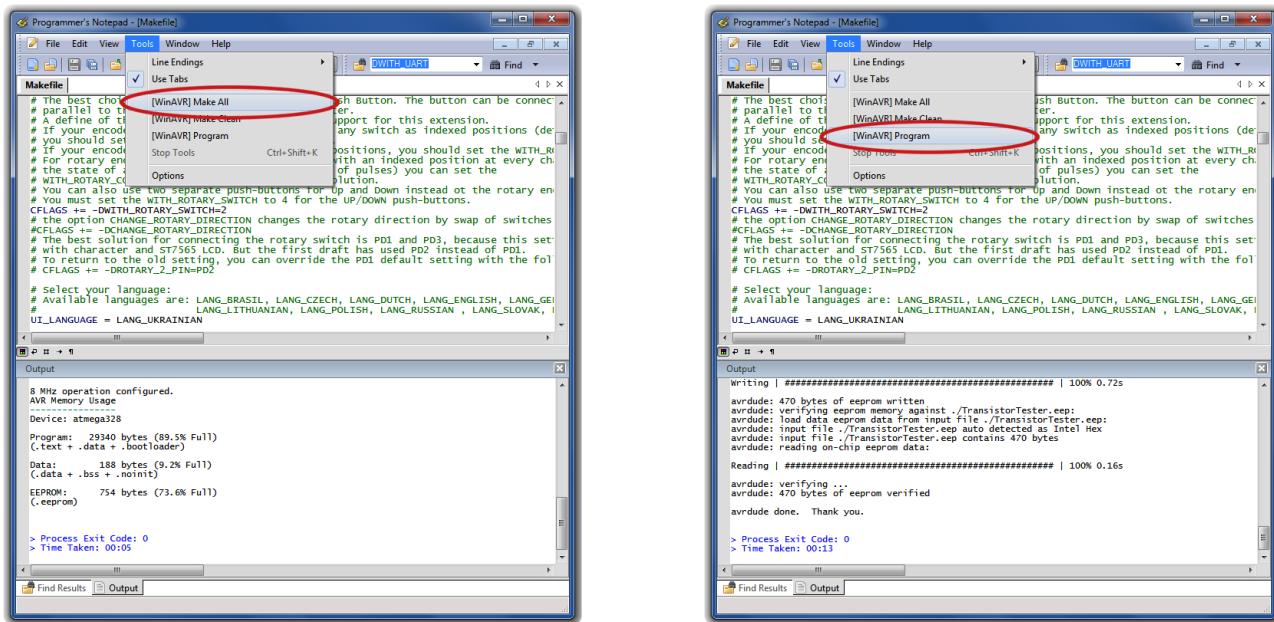
Derzeit kann das WinAVR Paket nicht mehr empfohlen werden, da das Paket sehr alt ist und nicht gewartet wurde. Der integrierte avr-gcc Kompiler ist damit auch sehr alt. Die neueren Versionen des

avr-gcc Kompilers optimieren das Programm deutlich besser. Weil oft der Speicher des ATmege schon bei gut optimierenden Kompiler randvoll ist, kann man WinAVR nur benutzen, wenn man Funktionen in der Makefile abwählt. Dann können Sie das WinAVR-Paket [18],[19] benutzen. Mit meinem Patch [20] können Sie auch die Fuses mit der Makefile setzen. Natürlich muss das avrdude Programm Ihren Programmer unterstützen und die Konfiguration muss in der Makefile richtig angepaßt sein.



(a) open Makefile
 Abbildung 4.2. Bedienung der WinAVR-Oberfläche Programmer's Notepad

Die Abbildungen 4.2 zeigen das File-Menü der Bedienoberfläche von WinAVR zum Öffnen der Datei Makefile und zum Abspeichern der Makefile nach den Änderungen (save).



(a) Erzeuge Programmdaten (.hex/.eep)
 Abbildung 4.3. Bedienung der WinAVR-Oberfläche Programmer's Notepad

Die nächsten Abbildungen 4.3 zeigen das Tools-Menü von Programmer's Notepad zum Übersetzen des Programms (Make All) und zum Programmieren des ATmega (Program) mit avrdude .

Kapitel 5

Beschreibung des Messverfahrens

Ein vereinfachtes Schaltbild eines Eingangs-/Ausgangs-Pin des ATmega wird in Abbildung 5.1 gezeigt. Der Schalter PUD schaltet die Versorgung für alle „Pull Up“-Widerstände des ATmega ab. Mit dem Schalter DD kann der Ausgang abgeschaltet werden, der Eingang funktioniert sowohl im Ausgabe- wie im Eingabe-Modus. Im Eingabe-Modus wird mit dem Ausgabewert (PORT) der „Pull Up“-Widerstand des Eingangs mit geschaltet. Die beiden Schalter PORT und DD können nicht gleichzeitig, sondern nur nacheinander geschaltet werden. Weil beim Umschalten der „Pull Up“-Widerstand die Messung stören könnte, bevorzuge ich die komplette Abschaltung aller „Pull Up“-Widerstände mit dem PUD-Schalter. Natürlich sind die Schalter elektronisch und die Widerstände 19Ω und 22Ω sind angenäherte Werte.

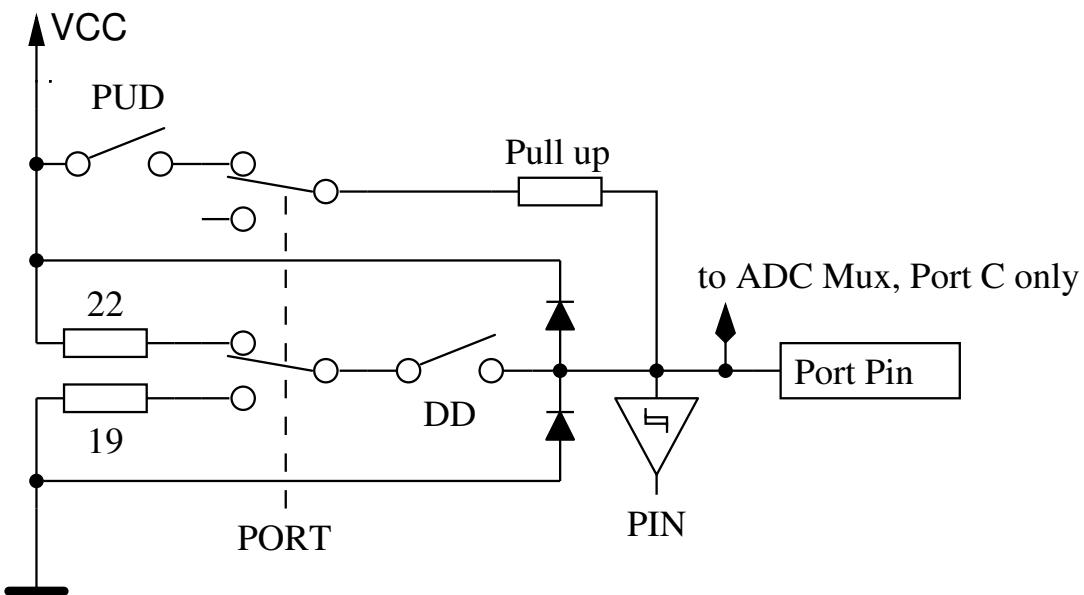


Abbildung 5.1. Vereinfachtes Schaltbild jedes ATmega-Portpins

Jeder der drei Testpins Ihres TransistorTesters wird aus drei ATmega-Portpins gebildet, was im vereinfachten Schaltbild des Testpins TP2 (mittlerer der drei Pins) in Abbildung 5.2 gezeigt wird.

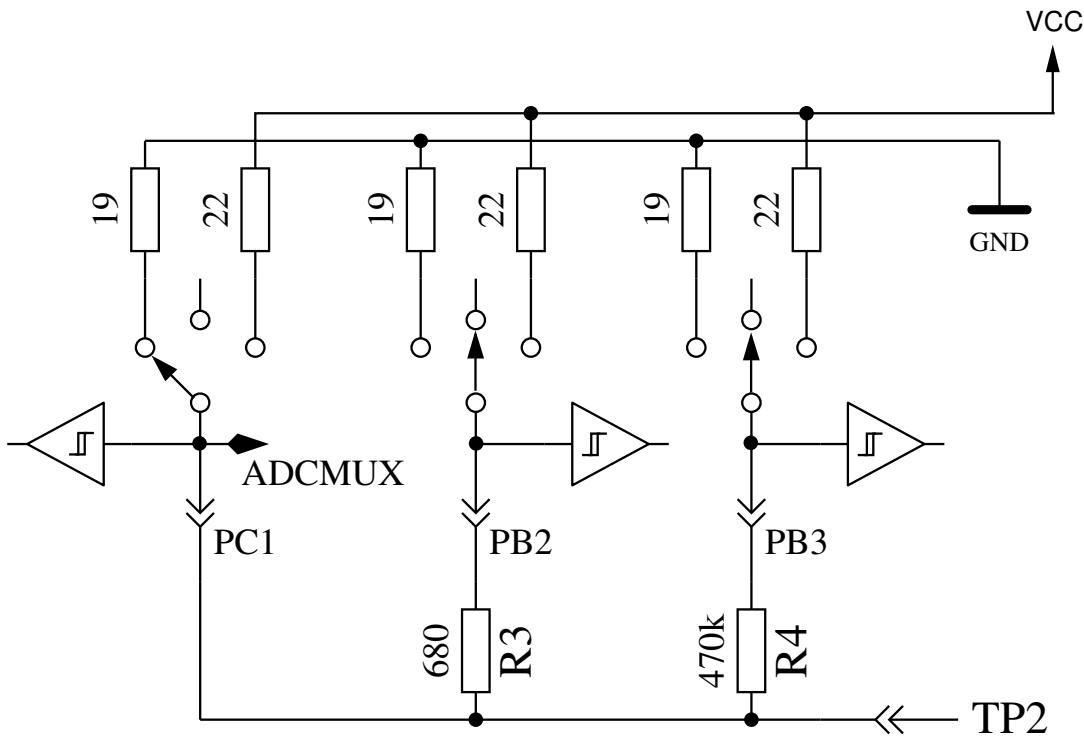


Abbildung 5.2. Vereinfachtes Schaltbild des Testpins TP2

Jeder Testpin (Messport) kann als digitaler oder analoger Eingang benutzt werden. Diese Messfähigkeit ist unabhängig von der Verwendung des Ports als Ausgang. Jeder Testpin kann als Ausgang verwendet werden und in diesem Zustand mit GND (0V) oder VCC (5V) verbunden werden, oder er kann über die Widerstände (680Ω oder $470k\Omega$) mit entweder GND oder VCC verbunden werden. Tabelle 5.1 zeigt alle denkbaren Messmöglichkeiten. Beachten Sie, dass der positive Zustand durch direktes Verbinden mit VCC (Port C) oder durch Verbinden mit dem 680Ω Widerstand mit VCC (Port B) erreicht werden kann. Die gleiche Möglichkeit hat der negative Zustand des Testpins zu der GND-Seite. Der Test-Zustand meint, dass der Pin offen sein kann (Eingang), verbunden über den $470k\Omega$ -Widerstand mit VCC oder GND, oder der Pin kann über den 680Ω -Widerstand mit VCC oder GND verbunden sein.

	Zustand Pin 1	Zustand Pin 2	Zustand Pin 3
1.	positiv	negativ	test
2.	positiv	test	negativ
3.	test	negativ	positiv
4.	test	positiv	negativ
5.	negativ	test	positiv
6.	negativ	positiv	test

Tabelle 5.1. alle Messmöglichkeiten

Wenn die Kondensatormessung des Testers konfiguriert ist, versucht der Tester vor allen Messungen erst einmal, die Kondensatoren an allen Anschlusspins zu entladen. Wenn das nicht gelingt, also die Restspannung zu hoch bleibt, wird das Entladen nach etwa 12 Sekunden mit der Meldung „Cell!“ abgebrochen. Dies kann auch dann vorkommen, wenn gar kein Kondensator angeschlossen ist. Die Ursache kann in diesem Fall sein, dass die Entlade-Grenzspannung für diesen ATmega zu niedrig gewählt ist. Man kann eine höhere Restspannung mit der Makefile-Option CAP_EMPTY_LEVEL wählen.

5.1 Messung von Halbleitern

Als erster Test soll der Stromfluß des Bauteils bei stromlosen Steuerpin (dritter Pin, auch TriState-Pin genannt) untersucht werden. Der Steuerpin ist beispielsweise das Gitter oder die Basis des Testobjektes. Ein Testpin wird die positive Seite des Bauteils angenommen und direkt mit VCC verbunden. Ein anderer Pin wird als negative Seite des Bauteils angenommen. Die negative Seite wird mit dem 680Ω Widerstand nach GND verbunden. Bei Feldeffekttransistoren ist der Zustand des Transistors von der Spannung des Gitters abhängig. Der TriState-Pin wird zuerst mit dem 680Ω -Widerstand für $5ms$ mit GND verbunden und die Spannung an der negativen Seite gemessen. Danach wird die Spannung des negativen Testpins wieder gemessen, während der TriState-Pin auf Eingang (hochohmig) geschaltet ist. Danach wird das angenommene Gate für $5ms$ mit dem 680Ω -Widerstand auf VCC geschaltet und die Spannung an der negativen Seite noch einmal gemessen. Wenn die gemessene Spannung jetzt niedriger ist als bei der ersten Messung, wird diese Schaltung als richtig angenommen. Dann wird die Spannung noch einmal mit stromlosem Tristate-Pin gemessen.

Wenn die Spannung des negativen Pins mit festgehaltenem Pegel größer als $115mV$ ist und dieser Pegel nicht $100mV$ niedriger als der Pegel mit stromlosen Tristate-Pin ist, wird ein Verarmungs-Typ angenommen. Bei bipolaren Transistoren mit hohem Reststrom ist der Kollektor-Reststrom bei stromloser Basis deutlich höher. Durch die Überprüfung beider Pegel wird eine Falschdetektion von Germanium-Transistoren mit höheren Kollektor-Restströmen als Verarmungs-Transistoren (JFET) vermieden. Es werden dann weitere Tests gemacht, um N-Kanal JFET oder D-MOSFET und P-Kanal JFET oder P-MOSFET zu unterscheiden. Die MOSFET-Versionen können erkannt werden durch das Fehlen von Steuerstrom in jedem TriState-Pin-Zustand.

Um Parameter der Verarmungstypen messen zu können, werden sie mit einem 680Ω -Widerstand am Source-Pin vermessen, wie in Abbildung 5.3 gezeigt wird. Diese Messung wird anstelle der üblichen Messung des Stromes bei einer Gate-Spannung auf Source-Potential gemacht, da wegen des relativ hohen 680Ω Widerstandes in vielen Fällen der Kennstrom I_{DSS} des FETs nicht erreicht würde.

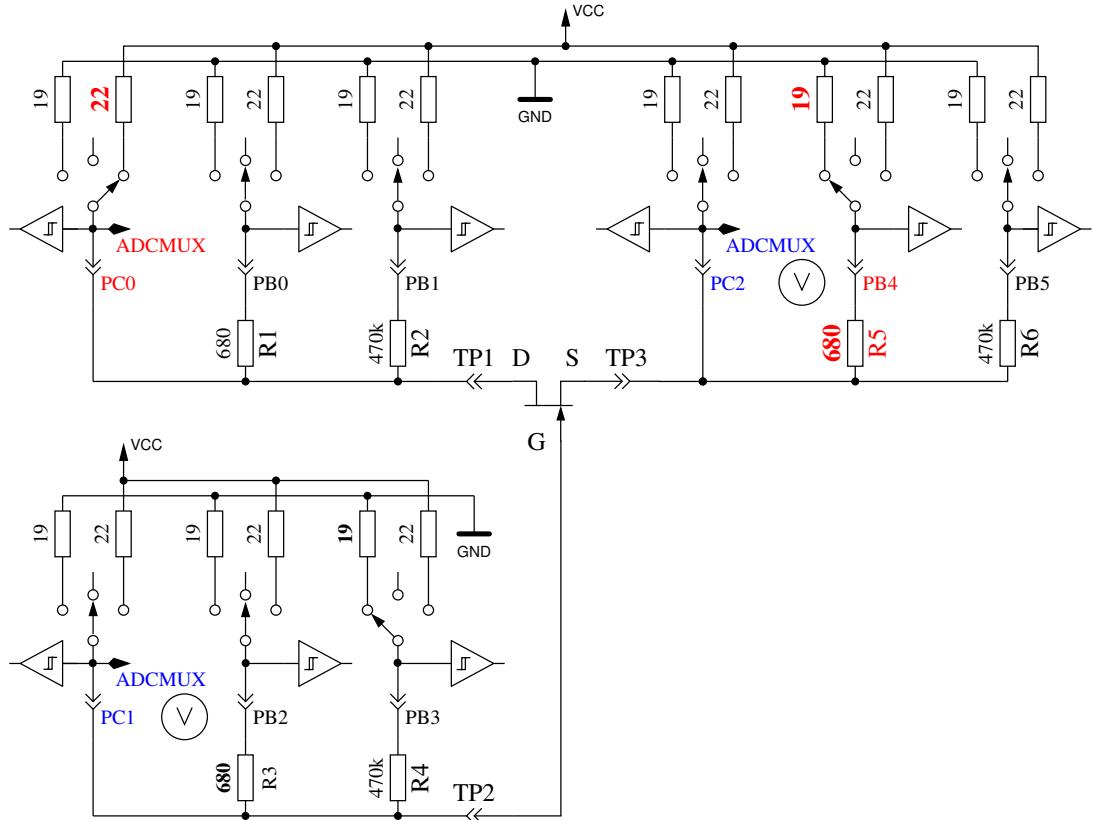


Abbildung 5.3. Messung von Gate-Source-Spannung und Source-Strom eines N-JFET-Transistors

Wenn das Bauteil keinen Strom zwischen dem positiven Pin und dem negativen Pin ohne ein Signal auf dem Tristate-Pin hat, sind die nächsten Tests im nächsten Unterkapitel 5.1.1 beschrieben. Wenn Strom festgestellt wird, sind die nächsten Tests in dem Dioden-Unterkapitel 5.1.4 beschrieben.

5.1.1 Messung eines PNP-Transistors oder eines P-Kanal MOSFETs

Zuerst wird der Stromverstärkungsfaktor in der Kollektor-Schaltung (Emitter-Folger) für den angenommenen PNP-Transistor gemessen. Die Messsituation wird in Abbildung 5.4 gezeigt. Wenn die gemessene Basis-Spannung (UB) über $9mV$ mit dem 680Ω Widerstand liegt, wird die Stromverstärkung hFE berechnet mit $hFE = \frac{UE - UB}{UB}$. Die Spannung UE ist die Differenz der Emitter-Spannung zu VCC. Die Differenz des 22Ω und 19Ω -Widerstandes wird nicht berücksichtigt. Wenn die Spannung UB unter $10mV$ liegt, wird die Messung mit dem $470k\Omega$ -Widerstand an der Basis gemacht. Für diesen Fall wird der Stromverstärkungsfaktor mit $hFE = \frac{UE \cdot 470000}{UB \cdot (680 + 22)}$ gebildet.

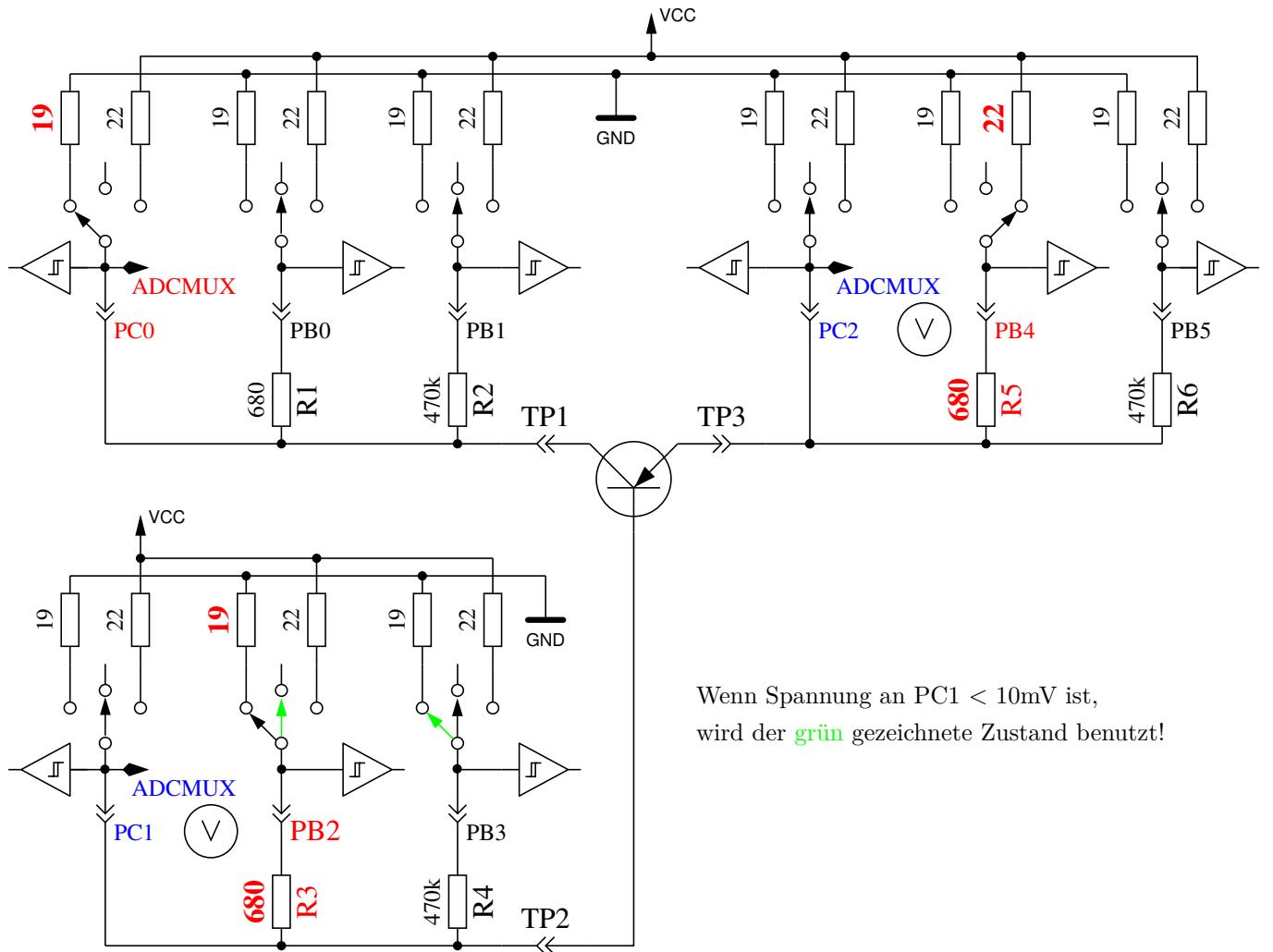


Abbildung 5.4. hFE-Messung eines PNP-Transistors in Kollektor-Schaltung

Als Nächstes werden die Tests in Emitter-Schaltung für den angenommenen PNP-Transistor gemacht. Die positive Seite wird jetzt direkt mit VCC verbunden, der 680Ω -Widerstand der negativen Seite wird mit GND verbunden, wie es in Abbildung 5.5 gezeigt wird. Wenn die negative Seite des Bauteils eine Spannung über $3,4V$ hat, wenn der 680Ω -Widerstand auf der Basis-Seite mit GND verbunden ist, muss es ein PNP-Transistor oder ein P-Kanal-FET sein. Das kann einfach unterschieden werden durch Prüfen der Basis-Spannung: Wenn sie grösser als $0,97V$ ist, muss es ein PNP sein. Für die Messung des Stromverstärkungsfaktors wird anstelle des 680Ω -Widerstandes der $470k\Omega$ -

Widerstand als Basis-Widerstand genommen. Der Stromverstärkungsfaktor wird berechnet mit $hFE = \frac{(UC - UC0) \cdot 470000}{UB \cdot (680 + 19)}$. Die Spannung $UC0$ ist die Spannung am Kollektorwiderstand ohne Basisstrom. Der höhere Stromverstärkungsfaktor wird als der richtige angenommen, dieser hier oder der mit der Kollektor-Schaltung bestimmte.

Die Werte, die für den PNP-Transistor herausgefunden wurden, sind nur gültig, wenn ein zweiter Satz von Messungen gemacht wurde. Um zu verhindern, dass der PNP-Transistor in der inversen Schaltung (Kollektor und Emitter vertauscht) erkannt wird, wird dann die Messung mit dem höheren Stromverstärkungsfaktor als richtige Messung genommen. Wenn die Basis-Spannung kleiner als $0,97V$ ist, muss es ein P-E-MOS sein. In diesem Fall wird die Gate-Schwellwertspannung dadurch bestimmt, dass die Spannung am Gate langsam mit dem $470k\Omega$ -Widerstand rau auf und runter gezogen wird, bis die Drain-Seite schaltet und dann die Spannung am Gate gemessen wird.

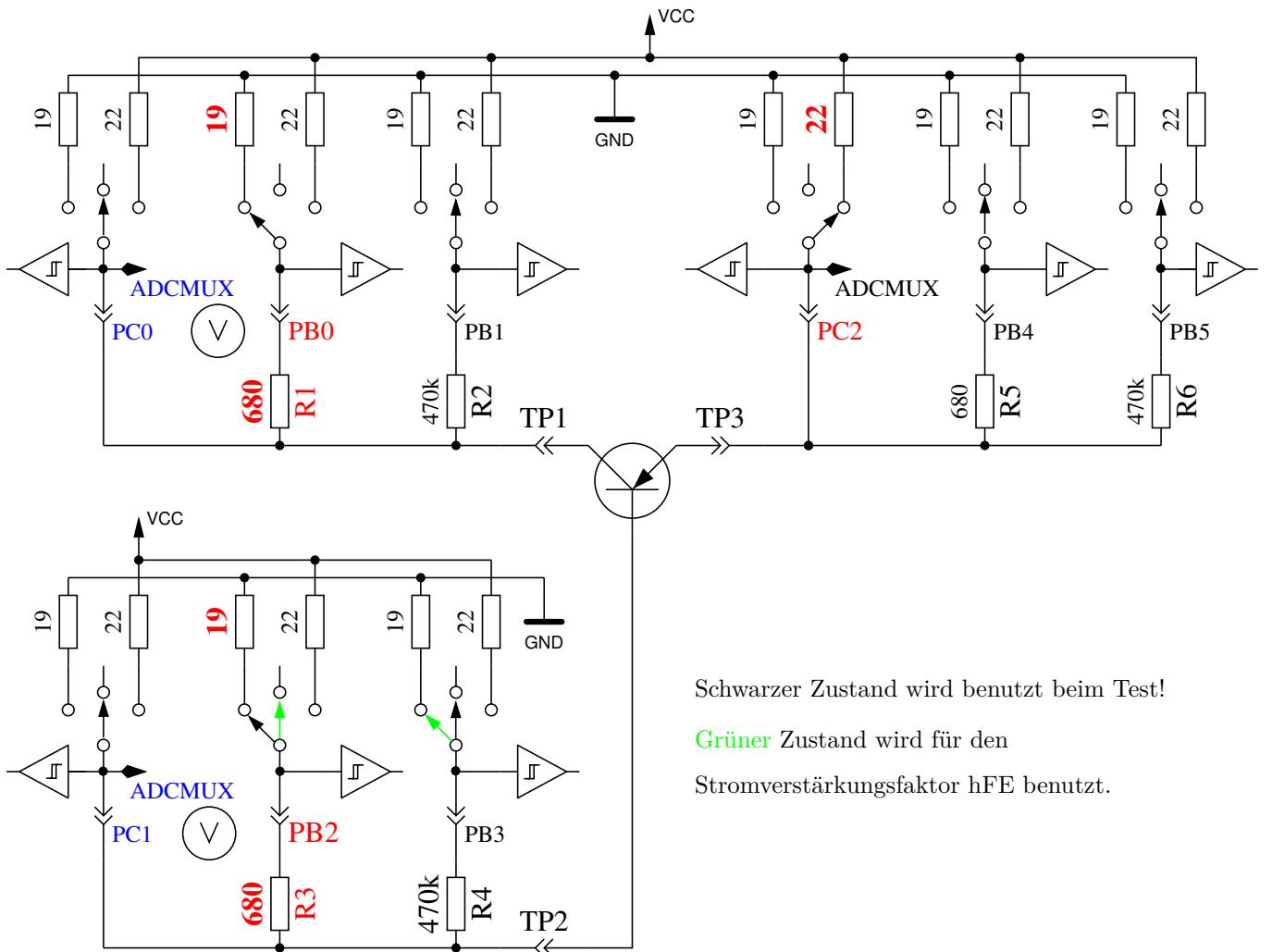


Abbildung 5.5. Prüfung und hFE -Messung eines PNP-Transistors in der Emitter-Schaltung

5.1.2 Messung eines NPN-Transistors oder eines N-Kanal-MOSFET

Die Messung eines NPN-Transistors beginnt auf gleiche Weise wie die PNP-Transistor-Messung, nämlich mit der Messung des Stromverstärkungsfaktors in der Kollektor-Schaltung. Zuerst wird die Messung mit einem nach VCC geschalteten 680Ω -Basiswiderstand gemacht. Wenn die Spannung am Basis-Widerstand zu klein ist, wird stattdessen der $470k\Omega$ -Widerstand genommen. Die Messungen werden dann in der Emitter-Schaltung fortgeführt, wie in Abbildung 5.6 gezeigt.

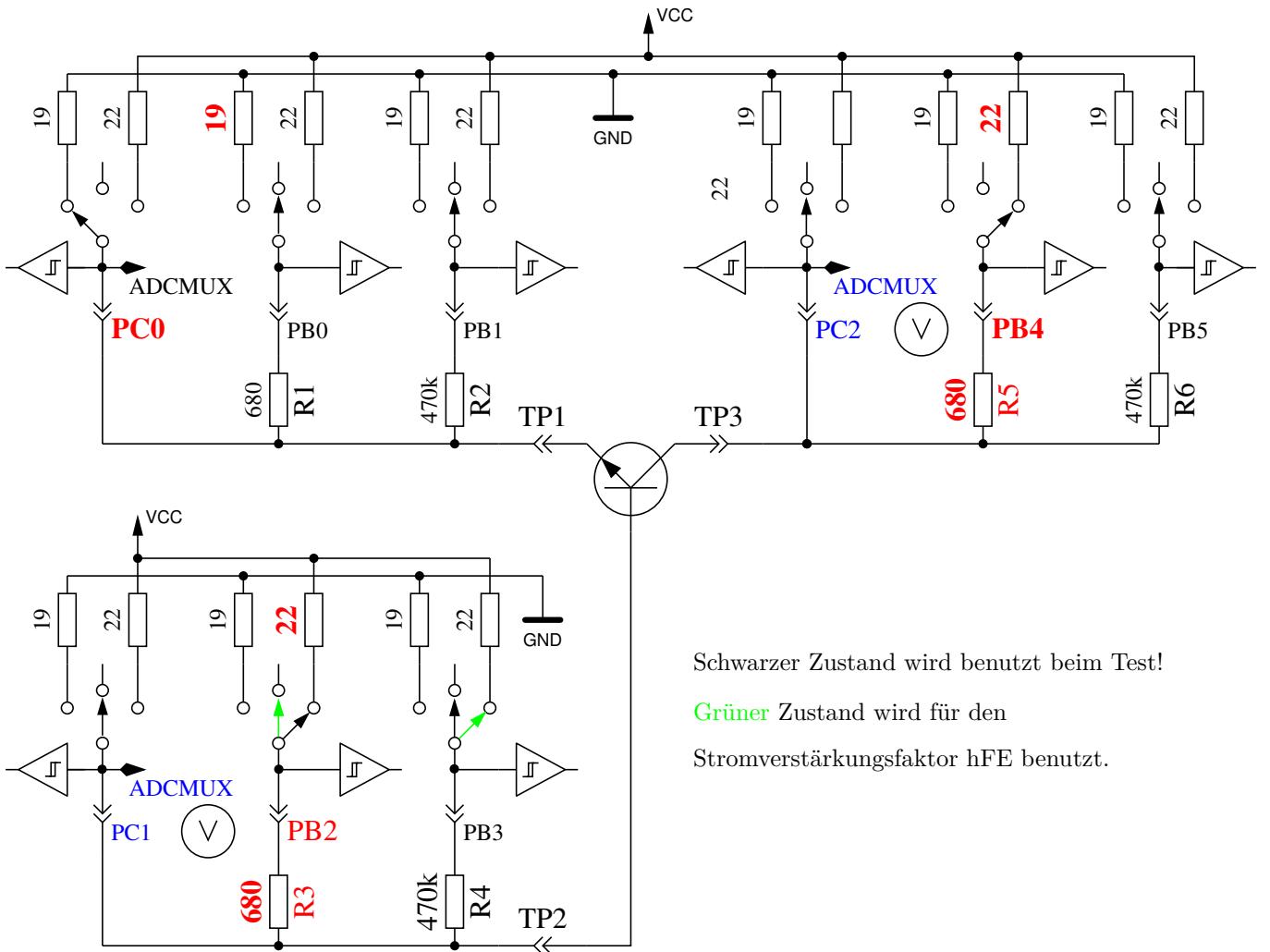


Abbildung 5.6. Prüfung und hFE-Messung eines NPN-Transistor in Emitter-Schaltung

Wenn die Spannung auf der Kollektor Seite unter 1,6V liegt, während der 680Ω-Basiswiderstand mit VCC verbunden ist, muss es ein NPN, ein N-Kanal MOSFET oder ein Thyristor (TRIAC) sein. Mit zwei einfachen Tests kann ein Thyristor oder TRIAC erkannt werden. Wenn der Gate-Pin für 10ms mit GND verbunden wird und dann stromlos geschaltet wird, sollte der Strom an der Anode bleiben. Wenn jetzt der Anoden-Widerstand kurz auf GND geschaltet und dann auf VCC zurückgeschaltet wird, sollte der Thyristor nicht erneut zünden (stromlos bleiben). Beachten Sie, dass nur Kleinleistungs-Thyristoren getestet werden können, weil der Haltestrom des Testers nur 6mA erreichen kann. Wenn beide Tests einen Thyristor bestätigen, werden weitere Tests in umgekehrter Polarität gemacht, um ein TRIAC auszuschliessen oder zu bestätigen.

Wenn weder Thyristor noch TRIAC bestätigt wurden, kann es ein NPN oder ein N-Kanal E-MOSFET sein. Die Basis-Spannung von einem NPN-Transistor wird nahe bei der Emitter-Spannung liegen, so dass dieser Typ sicher erkannt werden kann. Der Stromverstärkungsfaktor in der Emitter-Schaltung wird durch $hFE = \frac{(VCC - UC - UCO) \cdot 470000}{(VCC - UB) \cdot (680 + 22)}$ gebildet. Wenn die Spannung an der Basis zeigt, dass kein oder wenig Strom fließt, wird das Bauteil ein N-Kanal E-MOS (Anreicherungs-MOSFET) sein. In diesem Fall wird die Schwellspannung gemessen, indem die Spannung des Gates langsam mit dem 470kΩ-Widerstand nach VCC und GND gezogen wird, darauf wartend, dass das digitale Eingangs-Signal auf der Drain-Seite schaltet, wobei dann die Gate-Spannung gelesen wird. Die Messung wird elf Mal wiederholt wie in Abbildung 5.7 gezeigt und die Ergebnisse addiert. Diese Summe wird mit Vier multipliziert und durch Neun geteilt, um eine Auflösung in mV zu erhalten.

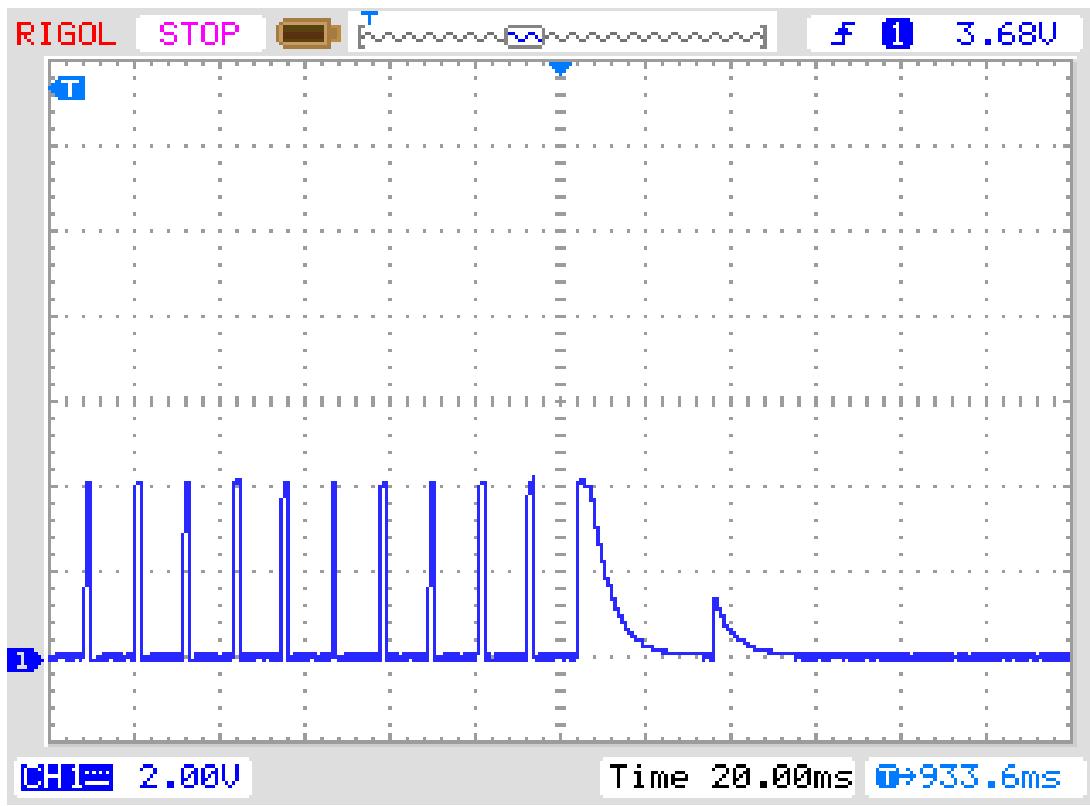


Abbildung 5.7. Messung der Schwellspannung eines N-Kanal-MOSFET

5.1.3 Vereinfachter Ablauf der Transistorerkennung

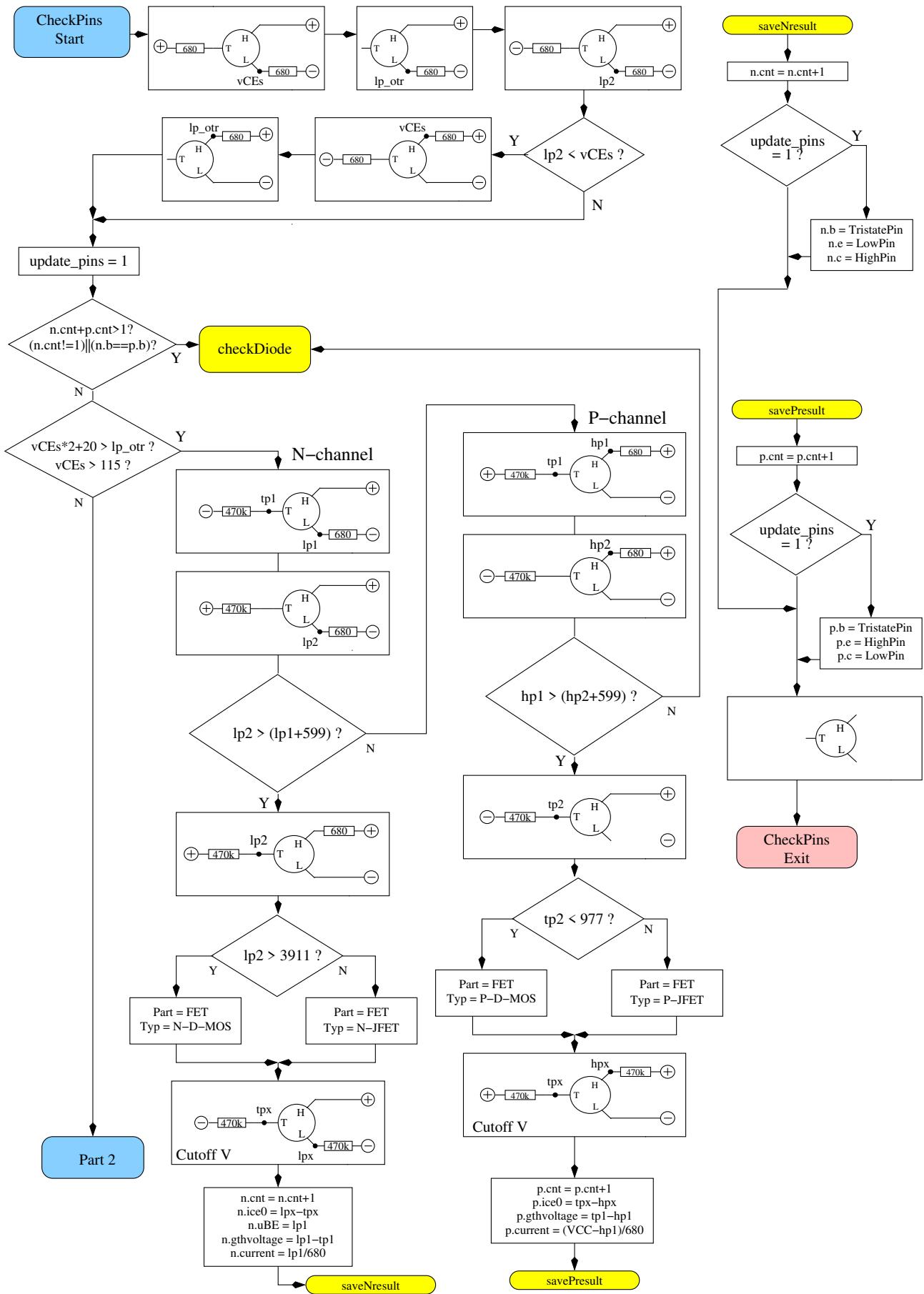


Abbildung 5.8. Ablaufplan der Transistorprüfung Teil 1, JFET und D-MOS

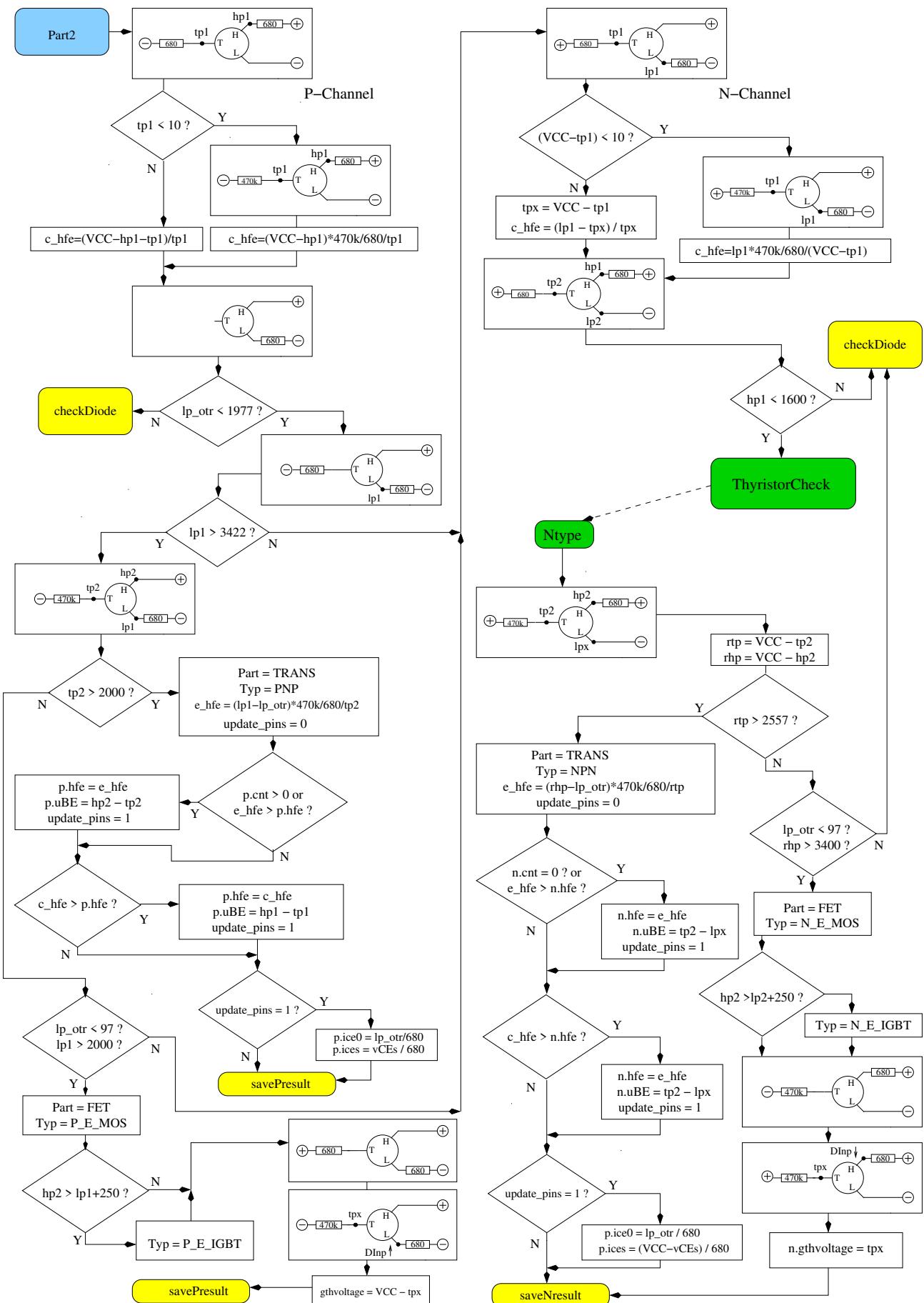


Abbildung 5.9. Ablaufplan der Transistorprüfung Teil 2, BJT und E-MOS

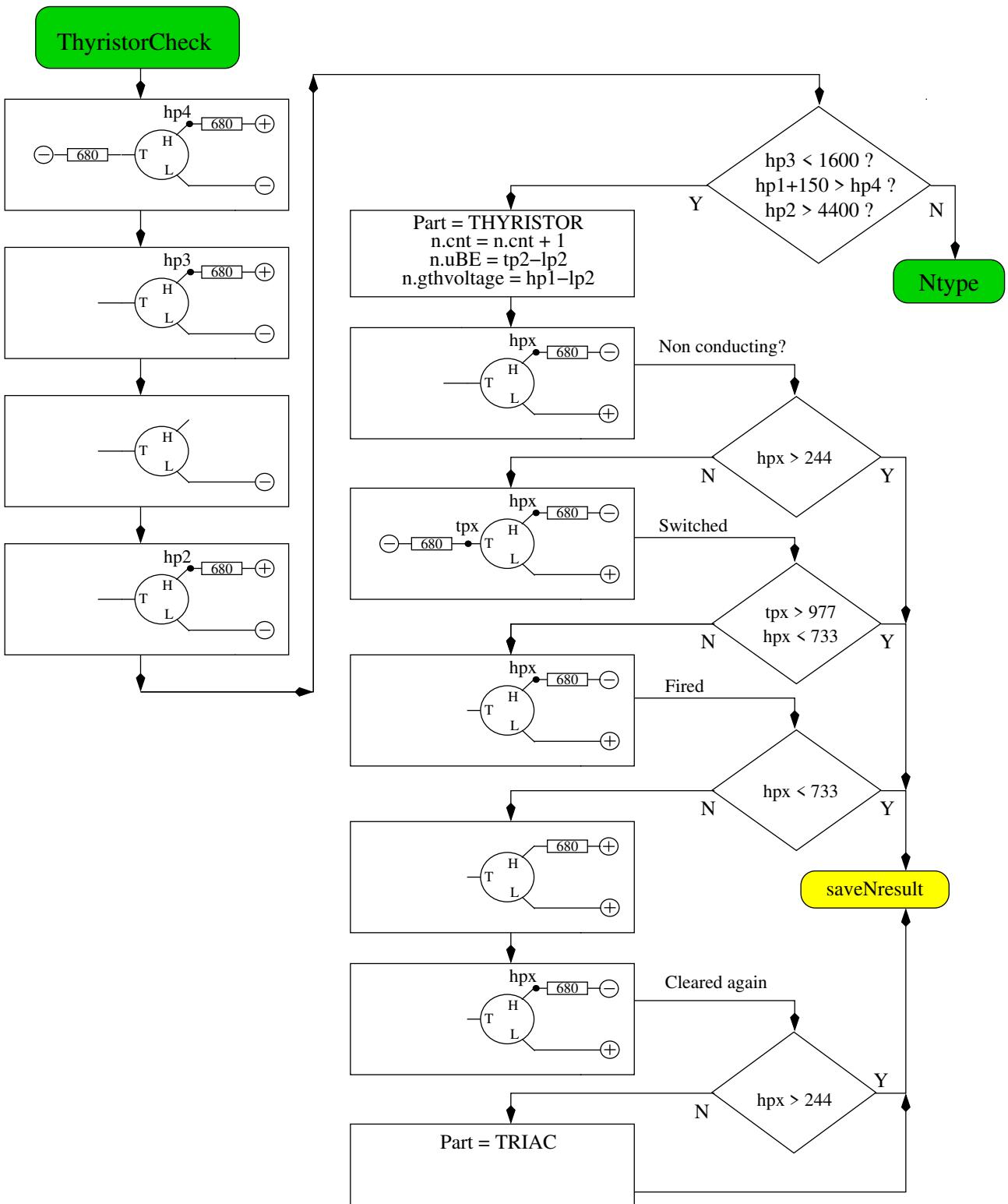


Abbildung 5.10. Ablaufplan der Transistorprüfung Teil 3, Thyristor und Triac

5.1.4 Messung von Dioden

Wenn Strom bei den Vortests festgestellt wurde, wird das Bauteil auf Diodenverhalten geprüft. Die Flussspannung mit dem 680Ω -Widerstand muss zwischen $0,15V$ und $4,64V$ liegen. Die Flussspannung mit dem 680Ω -Widerstand muss grösser als $1,125$ Mal der Flussspannung mit dem $470k\Omega$ -Widerstand sein und sechzehn Mal die Flussspannung mit dem $470k\Omega$ -Widerstand muss grösser als die Flussspannung mit dem 680Ω -Widerstand sein. Zusätzlich darf die anschließende nochmalige Messung mit dem $470k\Omega$ -Widerstand keine höhere Spannung als die Messung mit dem 680Ω -Widerstand

ergeben. Ich hoffe, dass ein Bauteil mit diesem Verhalten immer eine Diode ist. Die Erkennung des Diodenverhaltens durch den fehlenden Stromfluß in der Gegenrichtung ist nicht möglich bei antiparallelen Dioden. Bei einer Einzeldiode wird zusätzlich der Sperrstrom der Diode bei 5V mit dem $470\text{k}\Omega$ Widerstand gemessen. Die Auflösung beträgt etwa $2nA$. Bei größeren Restströmen als $5,3\mu\text{A}$ (Spannung am Widerstand größer als 2,5V) wird mit dem 680Ω Widerstand gemessen. Dann beträgt die Auflösung nur etwa $1\mu\text{A}$. Außerdem wird bei Einzeldioden eine Kapazitätsmessung in Sperr-Richtung durchgeführt.

5.1.5 Ergebnisse der verschiedenen Messungen

Die folgenden drei Tabelle zeigen die Ergebnisse verschiedener Bauteile eines ATmega8-, ATmega168- und ATmega328-Prozessors. Die Messung der Sperrsichtkapazität für die Doppeldiode MBR4045PT gelingt nur gekühlt. Die Ursache hierfür ist der hohe Reststrom der 40A-Diode. Ebenso kann für die Basis-Emitter-Strecke des Germanium-Transistors AC128 die Sperrsichtkapazität nur im gekühlten Zustand gemessen werden.

Diode Typ	Mega8@8MHz	Mega168 @8MHz	Mega328 @8MHz
1N4148	Diode, 715mV, 1pF	Diode, 718mV, 0pF, 2nA	Diode, 715mV, 1pF, 4nA
1N4150	Diode, 665mV, 1pF	Diode, 672mV, 1pF, 4nA	Diode, 666V, 2pF, 6nA
BA157	Diode, 619mV, 19pF	Diode, 621V, 17pF, 12nA	Diode, 615mV, 18pF, 12nA
BY398	Diode, 538mV, 16pF	Diode, 541mV, 14pF, 63nA	Diode, 537mV, 15pF, 63nA
1N4007	Diode, 650mV, 13pF	Diode, 655mV, 10pF, 6nA	Diode, 650mV, 13pF, 6nA
LED green	Diode, 1.96V, 5pF	Diode, 1.95V, 4pF	Diode, 1.95V, 4pF
ZPD2,7	2xDi, 743mV, 2.53V	2xDi, 737mV, 2.52V	2xDi, 733mV, 2.51V
BU508A B+E	Diode, 609mV, 5.15nF	Diode, 611mV, 5.20nF, 0.39uA	Diode, 606mV, 5.25nF, 0.4uA
BU508A B+C	Diode, 582mV, 256pF	Diode, 586mV, 255pF, 21nA	Diode, 587mV, 259pF, 19nA
AC128 B+E	Diode, 272mV, 0pF	Diode, 277mV, 0pF, 2.2uA	Diode, 273mV, 0pF, 2.3uA
AC128 B+E gekühlt			Diode, 349mV, 140pF, 0.57uA
MBR20100CT	2xDi, 337mV, 337mV	2xDi, 338mV, 338mV	2xDi, 336mV, 335mV
MBR20100CT	Diode, 337mV, 345pF	Diode, 339mV, 351pF, 29nA	Diode, 337mV, 350pF, 25nA
MBR4045PT gekühlt	Diode, 243mV, 1.80nF	Diode, 233mV, 1.94nF, 1.7uA	Diode, 235mV, 1.95nF, 1.8uA
SK14	Diode, mV, 0pF	Diode, mV, pF, nA	Diode, 263mV, 0pF, 0.57uA
SK14 gekühlt	Diode, mV, nF	Diode, mV, pF, nA	Diode, 334mV, 88pF, 4nA
SF38G	Diode, 519mV, 107pF	Diode, 521mV, 105pF, 2nA	Diode, 516mV, 106pF, 2nA

Tabelle 5.2. Messergebnisse der Dioden-Tests

Transistor Typ	Typ	Mega8 common-collector	Mega328 maximum	Mega328 common-collector	Mega328 common-emitter
BU508A	NPN	B=9, 601mV	B=9, 597mV	B=9, 598mV	B=4, 484mV
2N3055	NPN	B=20, 557mV	B=21, 550mV	B=21, 550mV	B=6, 442mV
BC639	NPN	B=148, 636mV	B=172, 629mV	B=172, 629mV	B=158, 605mV
BC640	PNP	B=226, 650mV	B=176, 609mV	B=171, 655mV	B=177, 608mV
BC517	NPN	B=23.9k, 1.23V	B=24.8k, 1.22V	B=25.1k, 1.22V	B=764, 1.23V
BC516	PNP	B=75.9k, 1.21V	B=76.2k, 1.20V	B=76.2k, 1.20V	B=760, 1.23V
BC546B	NPN	B=285, 694mV	B=427, 687mV	B=427, 687mV	B=369, 683mV
BC556B	PNP	B=304, 704mV	B=254, 668mV	B=235, 709mV	B=255, 668mV
AC128 (Ge.)	PNP	B=63, 191mV	B=59, 191mV	B=57, 193mV	B=43, 117mV
BUL38D parasitär	NPNp PNPn	B= 37, 627mV B= 11, 654mV	B=41, 617mV B=81, 543mV	B=40, 624mV B=10, 656mV	B=36, 562mV B=83, 541mV
BRY55/200	Thyrist.	0.84V	0.81V	0.82V	0.82V
MAC97A6	Triac	0.92V	0.90V	0.91V	0.90V

Tabelle 5.3. Messergebnisse der Tests mit bipolaren Transistoren

Die Ergebnisse der Transistormessungen unterscheiden sich teilweise erheblich von den Werten der Version von Markus Frejek. Zum Beispiel wird für den Darlington-Transistor BC517 von der früheren Software ein hFE von nur 797 statt 77200 gemessen. Dies hängt damit zusammen, dass die Stromverstärkung bei der neuen Version auch mit der Kollektorschaltung gemessen wird. Dies zeigen auch die Ergebnisse der neuen Version in der Emitterschaltung (common emitter), wie man in der letzten Spalte der Tabelle 5.3 sehen kann. Die Basis-Emitter-Spannung wurde früher mit einem separaten Diodentest mit 1438mV ermittelt. Jetzt wird die angegebene Basis-Emitter-Spannung im Zustand der Verstärkungsmessung (1, 20V) ermittelt. Der BUL38D-Transistor enthält eine Schutzdiode über der Anode und dem Kollektor des NPN-Transistors, wodurch ein parasitärer PNP-Transistor mit vertauschtem Basis-Kollektor Anschluss entsteht. In der Softwareversion 1.10k werden beide Transistoren erkannt und durch das angehängte p auf den weiteren Transistor hingewiesen. Der richtige Transistor (NPN) wird durch einen Vergleich der Sperrsichtkapazitäten herausgefunden. Es wird angenommen, dass der mit der höheren Sperrsichtkapazität der richtige Transistor ist. Wenn während der Ergebnisanzeige die Start-Taste gedrückt ist, werden die Parameter des parasitären Transistors angezeigt. Dabei wird wieder mit PNPn auf die andere Transistorstruktur hingewiesen. Die weitere Transistorstruktur entsteht nur bei der Integration der Schutzdiode in unmittelbarer Nachbarschaft des Transistors in das gleiche Halbleitermaterial, nicht bei einer externen Diode.

In der folgenden Tabelle 5.4 werden die Messergebnisse von Germanium-Transistoren gezeigt, die wegen den stark temperaturabhängigen Kollektor-Restströmen besonders problematisch sind. Es werden die Ergebnisse der Urversion von Markus F. und die Ergebnisse der 1.10k Version miteinander verglichen. Die 1.10k Version misst die Stromverstärkung sowohl in der Kollektorschaltung als auch in der Emitterschaltung mit Berücksichtigung des Kollektor-Ruhestroms, wobei die höhere Stromverstärkung ausgegeben wird. Der Kollektor-Ruhestrom wurde in älteren Versionen nicht berücksichtigt.

Transistor Typ	Mega8 @1MHz Ur-Version Markus F.	Mega168 @8MHz Version 1.10k	Mega328 @8MHz Version 1.10k
AC128	PNP, B=52, 279mV	PNP, B=59, 184mV	PNP, B=59, 191mV
AC116-65	PNP, B=505, 378mV	PNP, B=72, 146mV	PNP, B=72, 149mV
AC116-145	PNP, B=485, 294mV	PNP, B=146, 161mV	PNP, B=146, 163mV
AC176-65	NPN, B=98, 235mV	NPN, B=58, 94mV	NPN, B=56, 96mV
GC122	PNP, B=84, 368mV	PNP, B=55, 117mV	PNP, B=56, 117mV
GC301	PNP, B=48, 289mV	PNP, B=39, 184mV	PNP, B=39, 188mV
AD161	NPN, B=360, 230mV	NPN, B=296, 126mV	NPN, B=298, 128mV
AD162	PNP, B=2127, 280mV	PNP, B=89, 107mV	PNP, B=89, 107mV

Tabelle 5.4. Messergebnisse der Tests mit bipolaren Germanium-Transistoren

In der Tabelle 5.5 werden die Ergebnisse einiger Feldeffekttransistoren-Messungen gezeigt. Ein gemessenen Parameter der E-MOS-Typen ist Gate-Source-Schaltspannung, bei der das digitale Eingangssignal des ATmega eines am 680Ω Drain-Widerstand angeschlossenen Signals schaltet. Bei sehr schneller Änderung der Gatespannung wegen einer kleinen Gatekapazität ist die ermittelte Spannung etwas ungenau. Beim BS250 ändert sich die Gatespannung von 2,6V auf 2,5V, wenn man einen zusätzlichen $10nF$ Kondensator an Gate-Source anschließt.

Ein anderer gemessener Parameter ist die Gatekapazität. Die Gatekapazität wird ermittelt indem sowohl Source als auch Drain auf GND Potential gelegt wird. Bei IGBTs reicht oft die Gatespannung von 5V des Testers nicht zur Ansteuerung aus. Meistens wird dann nur die Emitter-Kollektor Schutzdiode erkannt. In diesem Fall kann eine an den Gatepin angeschlossene Batterie mit etwa 3V reichen, um die Erkennung zu ermöglichen. Der andere Pol der Batterie wird dann anstelle des Gatepins an den Testpin (TP) des Testers angeschlossen. Bei richtiger Polarität der Batterie wird dann die Erkennung des IGBTs ermöglicht. Die angezeigte Gate-Emitter Schaltspannung muß dann um die Batteriespannung erhöht werden, um die wahre Schaltspannung zu erhalten.

Bei JFET-Transistoren wird in den Datenblättern oft der Idss-Kennstrom genannt, der Strom im Drain bei einer Gate-Source-Spannung von 0V. Hier wird aber der Strom angegeben, der sich durch einen 680Ω Lastwiderstand auf der Source-Seite des JFET ergibt. Der Lastwiderstand erzeugt für das Gate eine Gegenspannung Vgs, die ebenfalls angegeben wird. Mit einem $470k\Omega$ Lastwiderstand auf der Source-Seite des JFET ist der Source-Drain Strom fast 0. Damit läßt sich die Gate-Source Cutoff Spannung Vgs_off hinreichend genau bestimmen, sofern sie unter 5V liegt. Mit diesen beiden Arbeitspunkten läßt sich aufgrund der quadratischen Strom-Kennlinie ein Igss schätzen. Sofern der geschätzte Strom unter $40mA$ liegt, wird noch eine zusätzliche Messung ohne Widerstand am Source Anschluß gemacht. Über die Spannung am Source-Anschluß kann ein weiterer Stromwert bestimmt werden. Mit diesem höheren Stromwert und der Gate-Source Spannung wird der Strom Idss noch einmal mit der quadratischen Strom-Kennlinie berechnet, sofern ein Wert von $40mA$ nicht überschritten wird. Wegen des symmetrischen Aufbaus der JFETs kann Drain und Source nicht unterschieden werden.

Transistor	Typ	Mega8 @8MHz	Mega168 @8MHz	Mega328 @8MHz
ZVNL120A	N-E-MOS	D, 1.6V, 147pF	D, 1.5V, 141pF	D, 1.5V, 140pF
IRF530N	N-E-MOS	D, 3.6V, 1.55nF	D, 3.6V, 1.54nF	D, 3.6V, 1.54nF
BS170	N-E-MOS	D, 2.6V, 78pF	D, 2.6V, 68pF	D, 2.6V, 68pF
IRL3803	N-E-MOS	D, 2.3V, 9.81nF	D, 2.3V, 9.71nF	D, 2.3V, 9.74nF
IRFU120N	N-E-MOS	D, 4.2V, 909pF	D, 4.2V, 913pF	D, 4.2V, 911pF
BUZ71A	N-E-MOS	D, 3.2V, 714pF	D, 3.2V, 708pF	D, 3.2V, 705pF
ZVP2106A	P-E-MOS	D, 3.2V, 122pF	D, 3.2V, 115pF	D, 3.2V, 116pF
IRF5305	P-E-MOS	D, 3.6V, 2.22nF	D, 3.6V, 2.22nF	D, 3.6V, 2.22nF
BS250	P-E-MOS	D, 2.6V, 53pF	D, 2.6V, 43pF	D, 2.6V, 44pF
IRFU9024	P-E-MOS	D, 3.5V, 937pF	D, 3.6V, 945pF	D, 3.5V, 933pF
J310 Idss=24-60mA	N-JFET	3.1mA Vgs=2.2V	3.1mA Vgs=2.2V	3.1mA Vgs=2.2V Idss=35mA
2N5459 Idss=4-16mA	N-JFET	2.1mA Vgs=1.5V	2.1mA Vgs=1.5V	2.1mA Vgs=1.5V Idss=8.2mA
BF256C Idss=11-18mA	N-JFET	3.4mA Vgs=2.4V	3.4mA Vgs=2.4V	3.4mA Vgs=2.4V Idss=14mA
BF245A Idss=2-6mA	N-JFET	1.1mA Vgs=.75V	1.1mA Vgs=0.75V	1.1mA Vgs=0.75V Idss=3.6mA
BF245B Idss=6-15mA	N-JFET	2.5mA Vgs=1.7V	2.5mA Vgs=1.7V	2.5mA Vgs=1.7V Idss=10mA
BF245C Idss=12-25mA	N-JFET	3.9mA Vgs=2.7V	3.9mA Vgs=2.7V	3.9mA Vgs=2.7V Idss=17mA
J175 Idss=7-60mA	P-JFET	3.2mA Vgs=2.2V	3.2mA Vgs=2.2V	3.2mA Vgs=2.2V Idss=26mA
2N5460 Idss=1-5mA	P-JFET	0.78mA Vgs=0.54V	0.77mA Vgs=0.54V	0.78mA Vgs=0.54V Idss=2.6mA
BSS139	N-D-MOS	1.7mA Vgs=1.2V	D, 1.7mA Vgs=1.2V	D, 1.7mA Vgs=1.2V
BSS169	N-D-MOS	2.6mA Vgs=1.8V	D, 2.6mA Vgs=1.8V	D, 2.6mA Vgs=1.8V
GP07N120	N-E-IGBT	C=3.81nF Vt=4.2V	C=3.76nF Vt=4.2V	C=3.74nF Vt=4.2V
IRG4PC30 mit Bat.	N-E-IGBT			C=2.22nF Vt=2.0V+3.2V

Tabelle 5.5. Messergebnisse der FET-Tests

5.2 Widerstands-Messung

Jeder Widerstand wird mit vier verschiedenen Messmethoden in einer Stromrichtung vermessen. Der gleiche Widerstand wird auch mit den gleichen vier Messmethoden in die andere Stromrichtung vermessen. Die Messung in die Gegenrichtung wird nur für die Erkennung auf Widerstand benutzt. Wenn die Abweichung dieser beiden Messungen zu groß ist, ist es kein Widerstand.

5.2.1 Widerstandsmessung mit den 680-Ohm-Widerständen

Die Messung des unbekannten Widerstandes R_x wird in zwei verschiedenen Wegen mit den 680Ω -Präzisionswiderständen durchgeführt. Das Schaltbild dieser Messungen mit Testpin 1 (TP1) und Testpin 3 (TP3) werden vereinfacht in Abbildung 5.11 und Abbildung 5.12 als ein Beispiel von den sechs Kombinationsmöglichkeiten gezeigt.

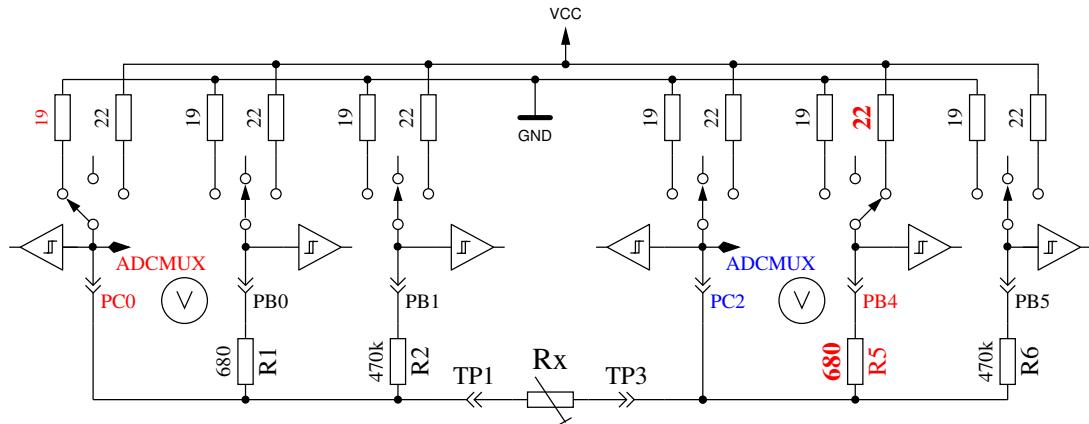


Abbildung 5.11. Messung Type 1 mit 680Ω

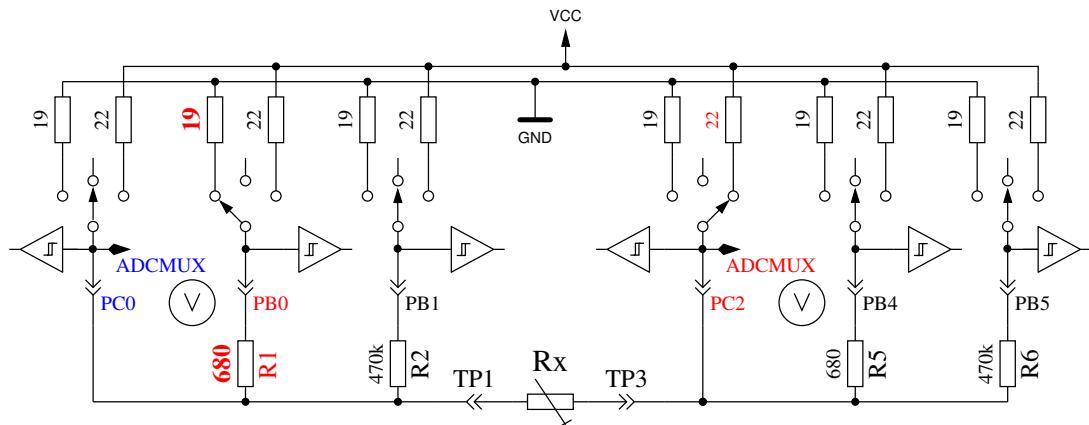


Abbildung 5.12. Messung Type 2 mit 680Ω

Auf der linken Seite wird der Testpin 1 und auf der rechten Seite der Testpin 3 gezeigt. In beiden Schaltungen kann man erkennen, dass der Anschluss 3 (TP3) mit VCC und die linke Seite (TP1) mit GND verbunden ist. Die Stromrichtung durch den Widerstand R_x ist immer die gleiche. Die Werte für auf Ausgang geschaltete Ports werden mit roter Farbe dargestellt, die Werte für die Eingänge werden mit blauer Farbe dargestellt, inaktive Ports sind schwarz. In beiden gezeigten Messmethoden sollte der Strom den gleichen Wert haben, weil die Summe der Widerstände zwischen VCC und GND gleich ist, vorausgesetzt die eingebauten Widerstände sind gleich. Normalerweise sind die gemessenen Spannungen aber nicht gleich, weil die Reihenfolge der Widerstände vertauscht ist.

Das V-Symbol innerhalb eines Kreises markiert die Ports, die für die Spannungsmessung benutzt werden. In beiden Konfigurationen kann der Wert des Widerstandes Rx aus den bekannten Widerstandswerten und den gemessenen Spannungen berechnet werden, wenn das Verhältnis des Widerstands Rx und den 680Ω -Widerständen nicht zu hoch ist. Der theoretische Spannungsverlauf wird in Abbildung 5.13 gezeigt, wobei die Widerstandswerte in logarithmischer Skalierung dargestellt sind.

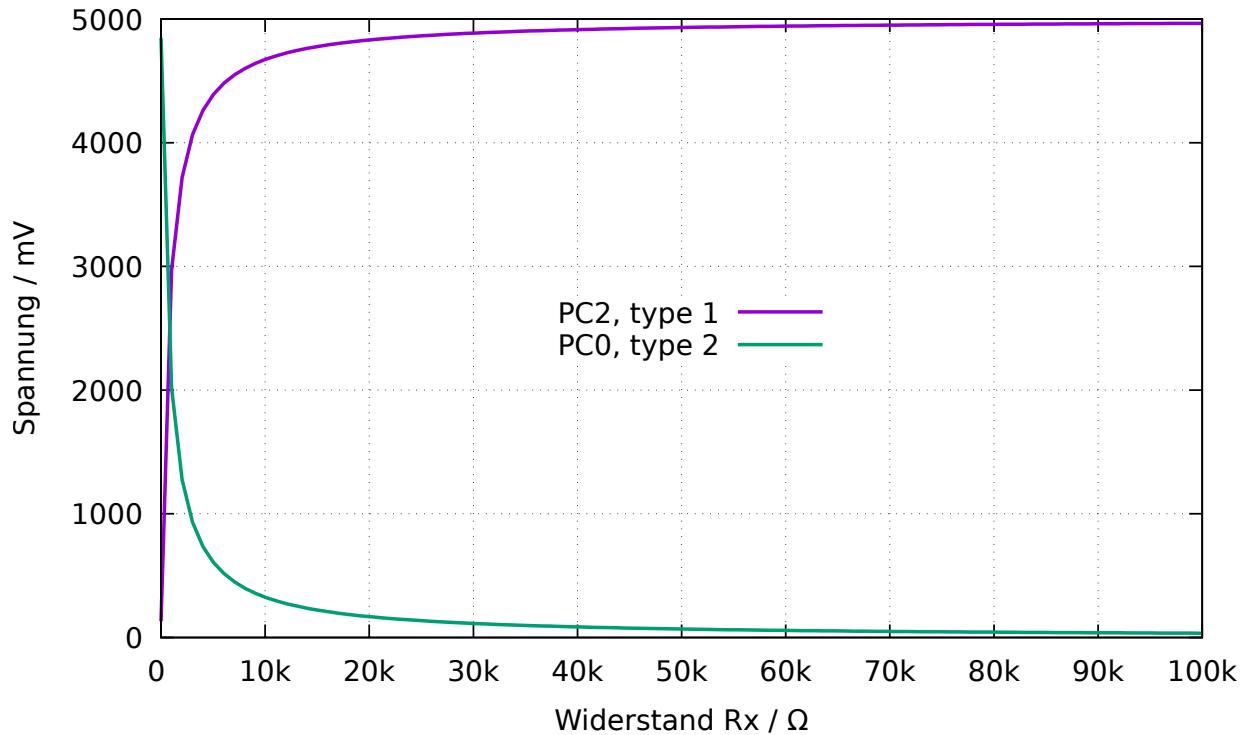
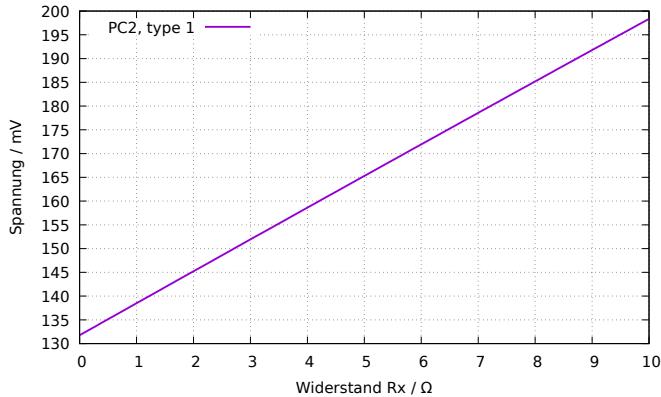


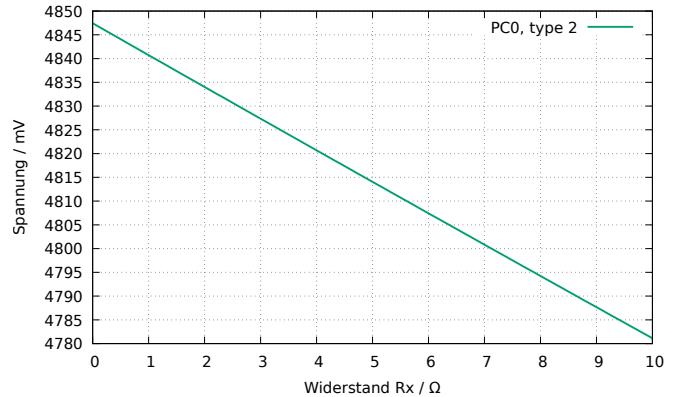
Abbildung 5.13. Spannung von Type 1 und Type 2 mit Messwiderstand 680Ω

Die Verlauf für Messung Type 1 wird in Abbildung 5.14a mit gespreizter Darstellung für die unteren Widerstandswerte gezeigt. Wie man sehen kann, braucht man eine bessere ADC-Auflösung als die möglichen $4,9mV$ bei der $5V$ ADC-Referenz, um richtige Widerstandswerte von der gemessenen Spannung unter 2Ω zu erhalten. Es gibt nur drei ADC-Stufen mit der $5V$ -Referenz zwischen 0Ω und 2Ω . Hier kann die Bereichsumschaltung mit der AUTOSCALE_ADC-Option helfen. Der gleiche gespreizte Bereich für die Type-2-Messung wird in Abbildung 5.14b gezeigt. Unglücklicherweise kann man nicht die höhere ADC-Auflösung für Messmethode Type 2 benutzen, weil die Spannung zu hoch ist und unser ATmega keine differentiellen ADC-Eingänge besitzt. Die Messungen mit den 680Ω -Widerständen werden bis zu einem Widerstandswert von $20k\Omega$ (Spannung ist unter $169mV$) zur Bestimmung des Messergebnisses verwendet.

Für höhere Widerstandswerte werden Messungen mit den $470k\Omega$ -Widerständen benutzt. Der Mittelwert von beiden Messungen wird für den angezeigten Widerstandswert benutzt, wenn alle Messungen ergeben, dass es kein anderes Bauteil ist. Wenn die AUTOSCALE_ADC-Funktion benutzt wird und eine der gemessenen Spannungen für beide Versionen unter $0,98V$ liegt, wird ein gewichteter Mittelwert mit Faktor Vier für die Messung mit der Spannung unter $0,98V$ benutzt. Der andere Wert wird mit Faktor Eins bewertet. Das wird wegen der Faktor Vier besseren Auflösung dieser Messung gemacht. Faktor Vier wird nur für ATmega168- und ATmega328-Prozessoren verwendet, für ATmega8 wird ein Faktor Zwei als Wichtung benutzt wenn die Spannung unter $0,98V$ ist, weil die ADC-Referenzspannung hier $2,56V$ statt $1,1V$ beträgt. Wenn der ATmega mehr als 8 KByte Flashspeicher besitzt, wird die Spannungsmessung an den Widerständen so lange verzögert, bis keine Änderung mehr festgestellt wird oder eine Zeitgrenze überschritten wird. Durch diese Maßnahme werden auch große Kondensatoren nicht mehr irrtümlich als Widerstände erkannt und der Gleichstrom-Widerstand großer Induktivitäten wird richtig gemessen.



(a) Type 1 Messung



(b) Type 2 Messung

Abbildung 5.14. Ausschnitt des theoretischen Spannungsverlauf von 0Ω bis 10Ω

5.2.2 Widerstandsmessung mit den 470-kOhm-Widerständen

Die nächsten Abbildungen 5.15 und 5.16 zeigen die gleichen Messmethoden für die Messungen mit $470k\Omega$ -Präzisionswiderständen. Weil $470k\Omega$ in Relation zu den Port-Widerständen 22Ω und 19Ω sehr groß ist, können die Port-Widerstände für die Berechnung des Widerstandswertes R_x vernachlässigt werden.

Für beide Messmethoden mit den $470k\Omega$ -Widerständen wird nur eine Spannung gemessen, weil der Strom so niedrig ist, dass keine Spannungsdifferenz an den internen Port Widerständen gemessen werden kann (wie zu erwarten). Der theoretische Spannungsverlauf wird in Abbildung 5.17 gezeigt, wobei die Widerstandswerte wieder in logarithmischer Skalierung gezeigt werden. Der theoretische Verlauf in diesem Diagramm endet bei $100M\Omega$, aber das Ergebnis des Testers wird auf $60M\Omega$ begrenzt, anderenfalls nimmt der Tester an, dass kein Widerstand angeschlossen ist. Als Ergebnis wird der gewichtete Mittelwert von beiden Messmethoden verwendet. Dies geschieht nach den gleichen Regeln, die schon bei den Messungen mit den 680Ω Widerständen beschrieben wurden. Ich habe beobachtet, dass die Messergebnisse für alle ATmega-Typen näher am wahren Wert liegen, wenn zum Messergebnis ein konstanter Offset von 350Ω addiert wird. Dieser Offset kann mit der Konstante RH_OFFSET (define) in der Datei config.h angepasst werden.

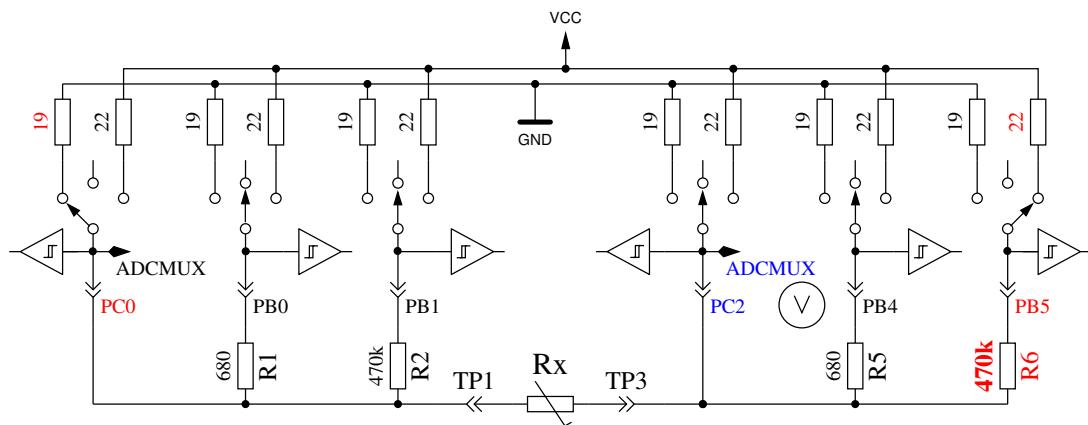


Abbildung 5.15. Messung Type 3 mit $470k\Omega$

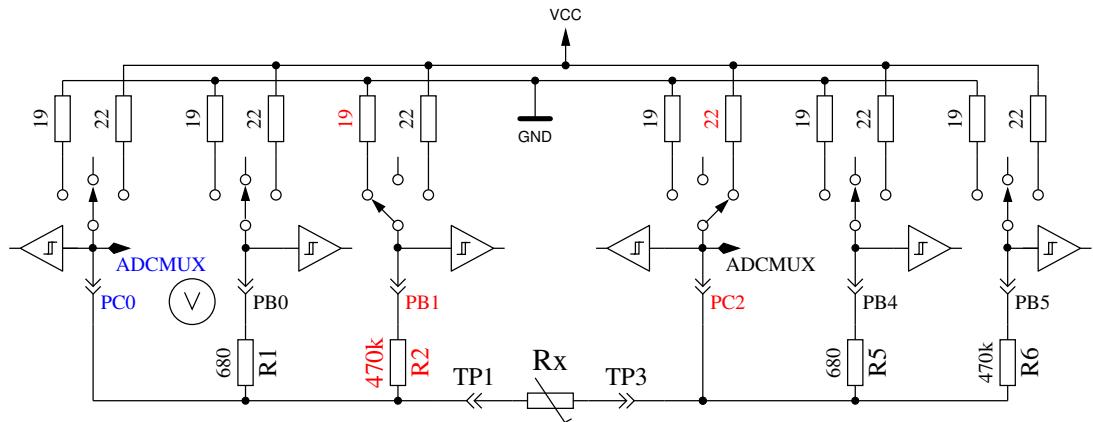


Abbildung 5.16. Messung Type 4 mit $470\text{k}\Omega$

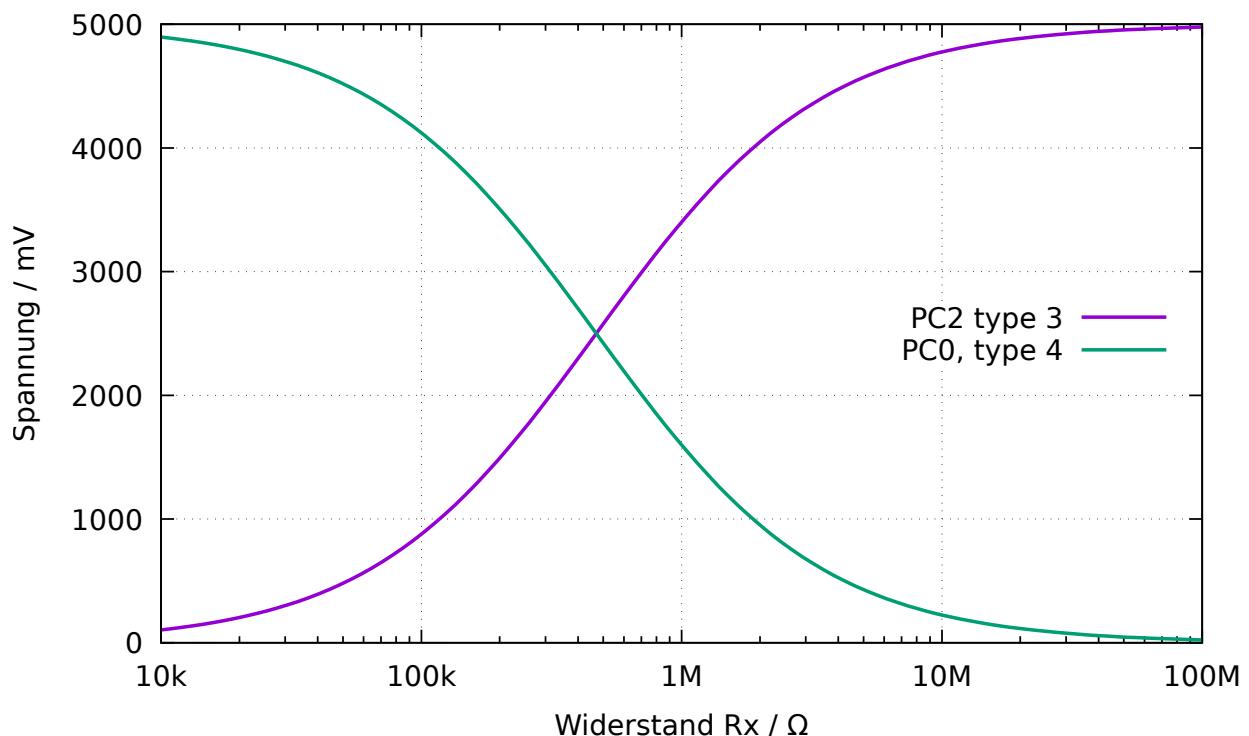


Abbildung 5.17. Spannungen bei Messungen von Type 3 und Type 4 mit $470\text{k}\Omega$

5.2.3 Ergebnisse der Widerstands-Messung

Abbildung 5.18 zeigt den relativen Fehler der Widerstandsmessung mit drei verschiedenen ATmega8. Zusätzlich werden die Messergebnisse einiger Widerstände mit der Originalsoftware von Markus F. als „Mega8orig“ gezeigt. Die Messergebnisse der gleichen Widerstände mit je drei ATmega8A und drei ATmega8L werden in den Abbildungen 5.19a und 5.19b gezeigt. Abbildung 5.20 zeigt die gleichen Messungen mit einem ATmega168. „Mega168“ sind die Ergebnisse ohne die AUTOSCALE_ADC Option, „Mega168as“ die mit der AUTOSCALE_ADC-Option.

Mit dem ATmega168 scheint es möglich zu sein, Messungen von Widerständen im Bereich von 20Ω bis $20M\Omega$ mit einem Messfehler von unter $\pm 1\%$ durchzuführen. Für Messungen unterhalb von 100Ω sollte man berücksichtigen, dass jede Prüfklemme mit Kabel ebenfalls einen Widerstandswert hat. Es ist besser, den Widerstand direkt mit den Anschlussklemmen zu verbinden. Wenn das nicht möglich ist, sollte man den Widerstandswert der kurzgeschlossenen Prüfklemmen vom Messergebnis abziehen. Zeigt beispielsweise der Tester einen Wert von $30,6\Omega$ an, wenn der Präzisionswiderstand

einen aufgedruckten Wert von 30Ω hat, und die im Kurzschluss gemessenen Prüfklemmen haben einen Wert von $0,5\Omega$, dann wird der Widerstand vom Tester mit $30,1\Omega$ gemessen. Unterhalb von einem Widerstandswert von 10Ω macht ein Auflösungs-Schritt von $0,1\Omega$ schon einen Fehler von mehr als 1%!

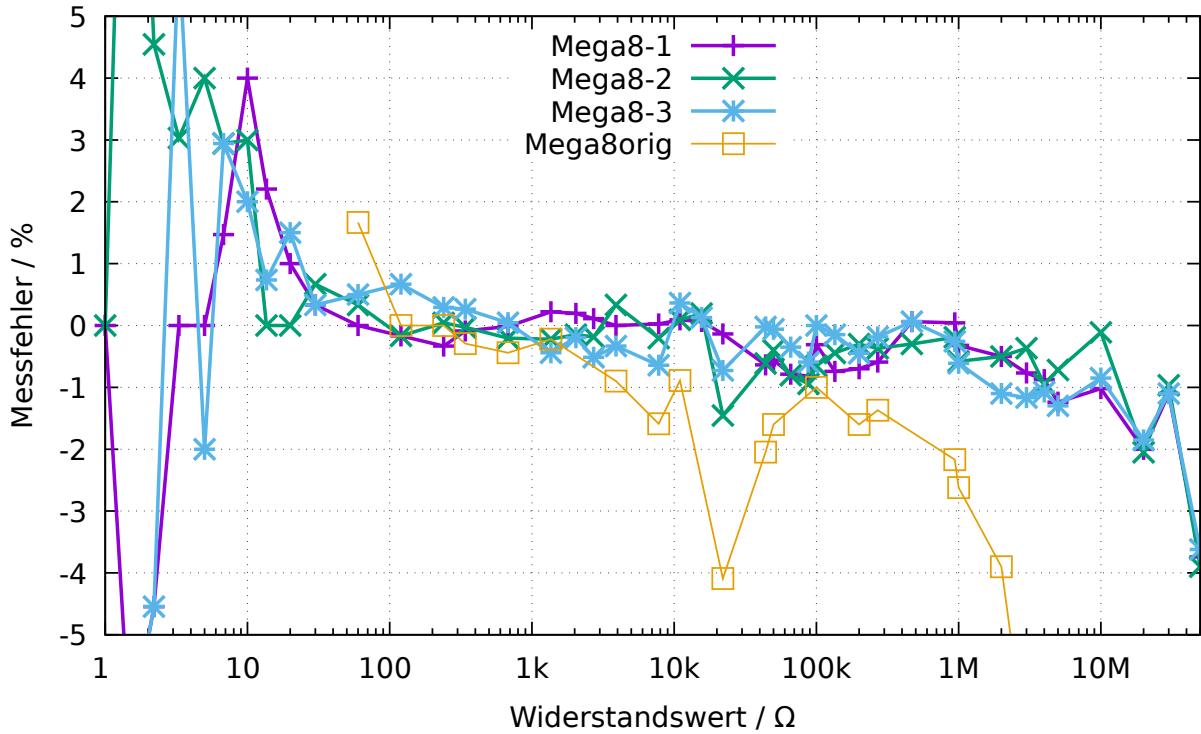
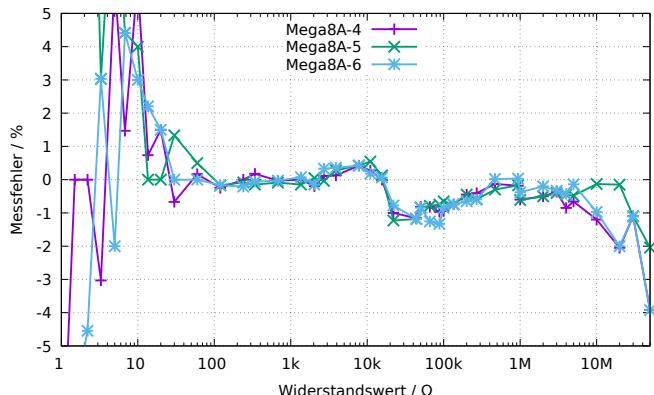
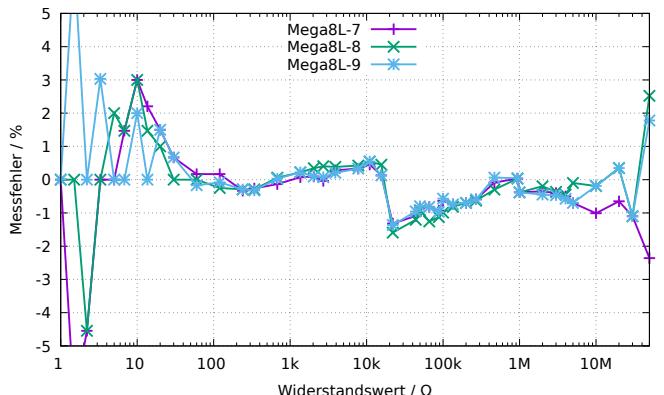


Abbildung 5.18. Relativer Fehler für Widerstands-Messungen mit ATmega8



(a) mit drei ATmega8A



(b) mit drei ATmega8L

Abbildung 5.19. Relativer Fehler für Widerstands-Messungen

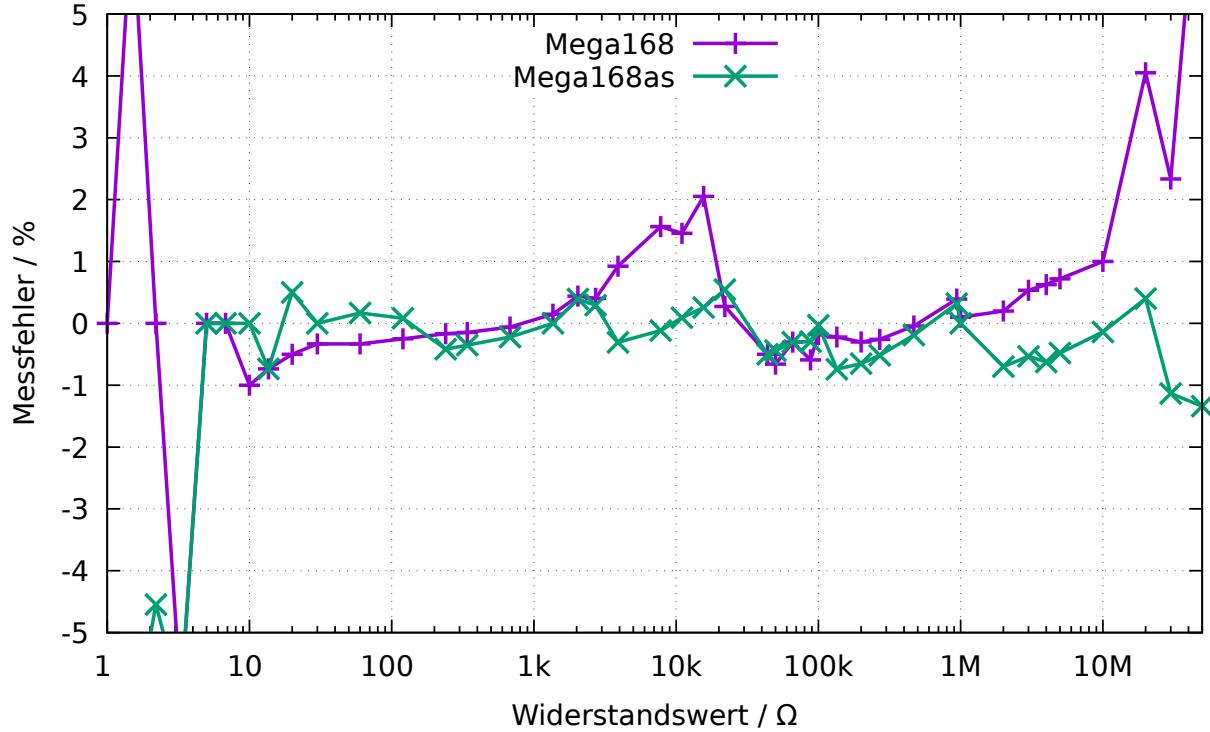


Abbildung 5.20. Relativer Fehler für Widerstand-Messungen mit ATmega168

Das Diagramm 5.21a zeigt die Messfehler von drei ATmega168-Prozessoren vor der Kalibration als Punkte, nach der Kalibration als Linie. Entsprechend werden die Messfehler von drei ATmega168A in Abbildung 5.21b und die Messfehler von drei ATmega328P in Abbildung 5.22 gezeigt. Die Messfehler der ATmega328 werden in den Abbildungen 5.23a und 5.23b gezeigt. Nach der automatischen Kalibration bleibt der Messfehler mit einer Ausnahme (ATmega328P-13, 22kΩ) im Widerstandsbereich 10Ω – 20MΩ im Bereich $\pm 1\%$. Vor der Kalibration können die Messfehler bei einigen Prozessoren bis $\pm 3\%$ betragen. Der Fehler wird verursacht durch die AUTOSCALE_ADC-Umschaltung der ADC-Referenz. Durch den direkten Vergleich einer Kondensatorspannung von unter 1V, einmal mit der VCC-Referenz und nochmal mit der internen ADC-Referenz gemessen, kann der Fehler ausgeglichen werden. In diesem Fall wird die Spannung mit dem gleichen Multiplexer-Kanal gemessen und die Bandgap-Referenz ist auf den AREF-Pin aufgeschaltet. Die direkte Vermessung der Bandgap-Referenz durch die direkte Wahl des Multiplexer-Eingangs führt leider zu diesem Offset, der entweder manuell mit der Option REF_R_KORR oder automatisch mit der Option AUTO_CAL des Selbsttestes beseitigt werden kann. Im AUTO_CAL-Modus ist REF_R_KORR ein zusätzlicher Offset zur automatisch gefundenen Spannungsdifferenz.

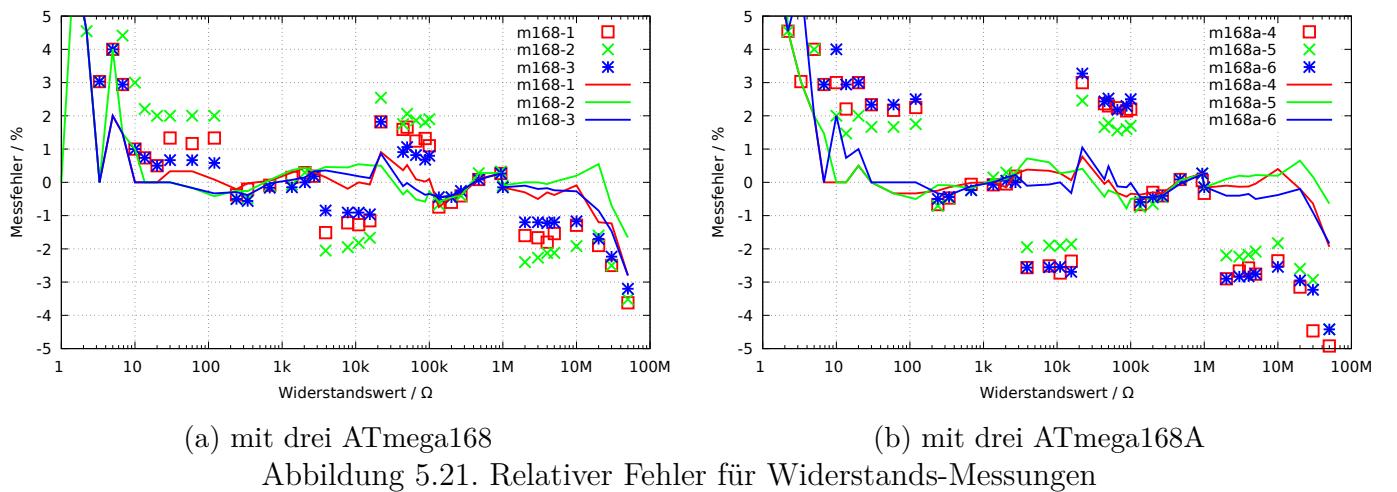


Abbildung 5.21. Relativer Fehler für Widerstands-Messungen

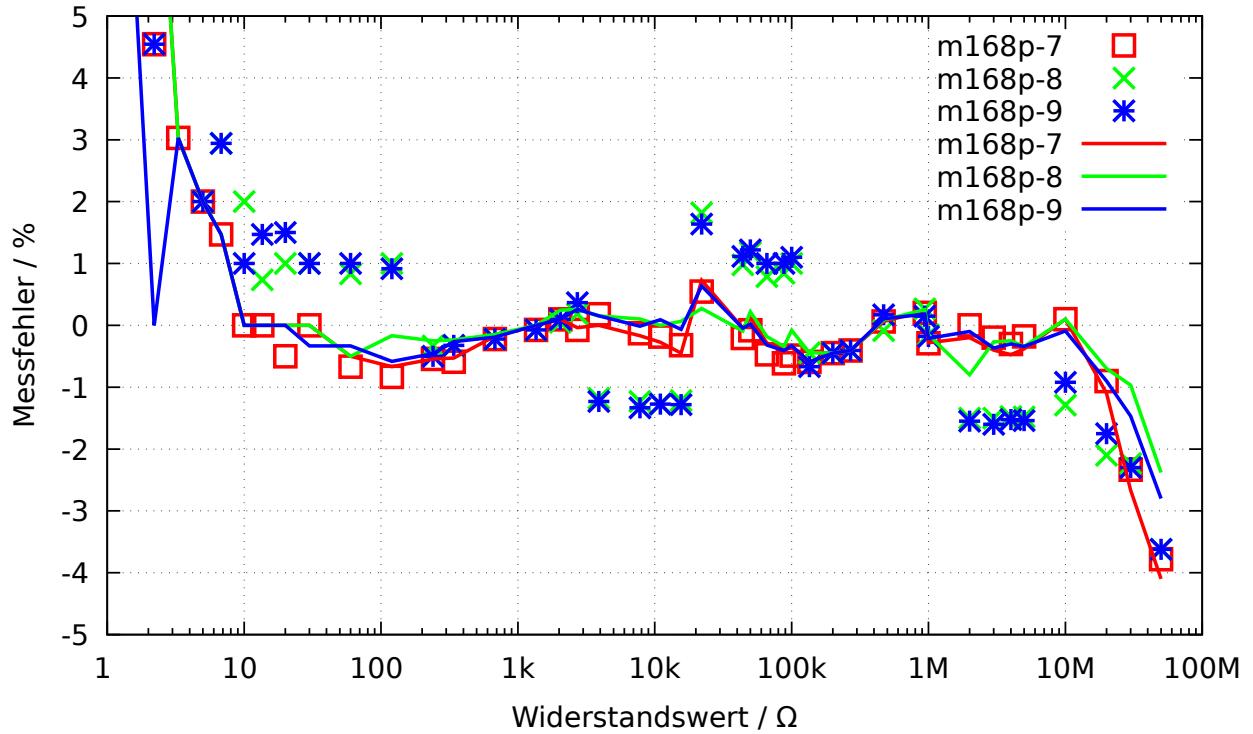
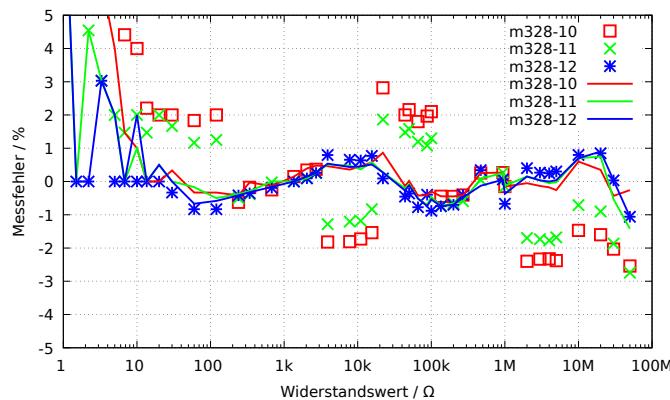
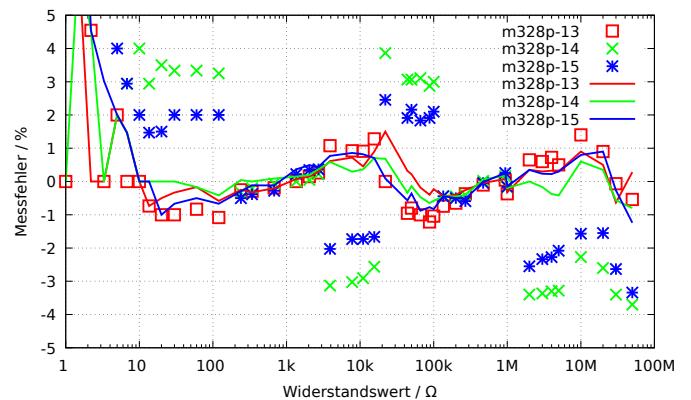


Abbildung 5.22. Relativer Fehler für Widerstands-Messungen mit drei ATmega168P



(a) mit drei ATmega328



(b) mit drei ATmega328P

Abbildung 5.23. Relativer Fehler für Widerstands-Messungen

5.3 Messen von Kondensatoren

Die Messung von Kapazitätswerten wird nach allen anderen Messungen als separater Teil mit einer Ladezeitmessung durchgeführt. Die Originalsoftware vom Markus F. hat das mit einer Programm-schleife gemacht, die den betreffenden digitalen Eingangs-Pin bis zu einer Signaländerung gelesen hat und dabei die Schleifendurchläufe gezählt. Dies hat den Nachteil, dass die Auflösung der Zeitmessung begrenzt ist durch die Gesamtzeit eines Schleifendurchlaufs. Dies wurde üblicherweise in allen sechs Kombinationsmöglichkeiten für die drei Testpins durchgeführt. Die aktuelle Software benutzt zwei verschiedene Möglichkeiten, um die Ladezeit in nur drei Kombinationsmöglichkeiten für die drei Testpins zu erhalten. Die positive Seite ist nun immer die höhere Testpin-Nummer. Nur wenn die Kapazität zusammen mit einer parallel geschalteten Diode gemessen wird, kann die Polarität die andere Richtung haben.

5.3.1 Entladen der Kondensatoren

Sie sollten die Kapazitäten immer entladen, bevor sie mit dem Tester verbunden werden. Bevor irgendein Test gestartet wird, wird der Kondensator vom Tester immer noch einmal entladen. Wenn die Spannung unter $1300mV$ ist, wird der Kondensator dafür mit den angeschlossenen ADC-Ports (Port C) kurzgeschlossen. Ich glaube das ist in Ordnung, weil jeder Portausgang einen Innenwiderstand von ungefähr 20Ω hat. Die Abbildung 149 (Seite 258) im ATmega8-Datenblatt [4] zeigt einen Spannungsabfall an Ausgabe-Pins von bis zu $2V$. Natürlich kann ich nicht garantieren, dass kein Schaden auftreten kann. Ich habe die Funktion mit Kondensatoren von mehr als $15mF$ sehr oft getestet und ich habe noch nie ein Problem bemerkt. Der Strom sollte unter der spezifizierten Grenze von $40mA$ bleiben und reduziert sich schnell durch die Entladung. Natürlich kann ein Schaden entstehen, wenn Sie einen (Hochvolt-) Kondensator nicht vollständig entladen, bevor Sie ihn an den Tester anschließen.

5.3.2 Messung von großen Kapazitäten

Eine Seite des Kondensators ist mit GND verbunden. Die andere Seite wird über den 680Ω -Widerstand für $10ms$ mit VCC verbunden. Danach wird dieser Messpin auf Eingang geschaltet (hochohmig). Nach diesem Strompuls wird die Spannung am Kondensator stromlos gemessen. Wenn die Spannung noch nicht den Minimalwert von $300mV$ erreicht hat, wird dieser Ladepuls bis zu weiteren 499 Mal wiederholt. Wenn nach 127 Pulsen (ungefähr $2s$) noch nicht eine Minimalspannung von $75mV$ erreicht ist, wird der Ladevorgang abgebrochen, weil die $300mV$ mit den verbleibenden Ladepulsen nie erreicht werden wird. Abbildung 5.24 zeigt die drei Phasen der Kapazitätsbestimmung eines Kondensators. Der Kapazitätswert wird dann berechnet aus der Ladepuls-Anzahl und der erreichten Ladespannung über eine Tabelle. Die Tabelle enthält mit einem Spannungsabstand von $25mV$ die Faktoren, um aus der Ladezeit und der Spannung den Kapazitätswert zu bestimmen. Zwischenwerte der Spannung werden interpoliert.

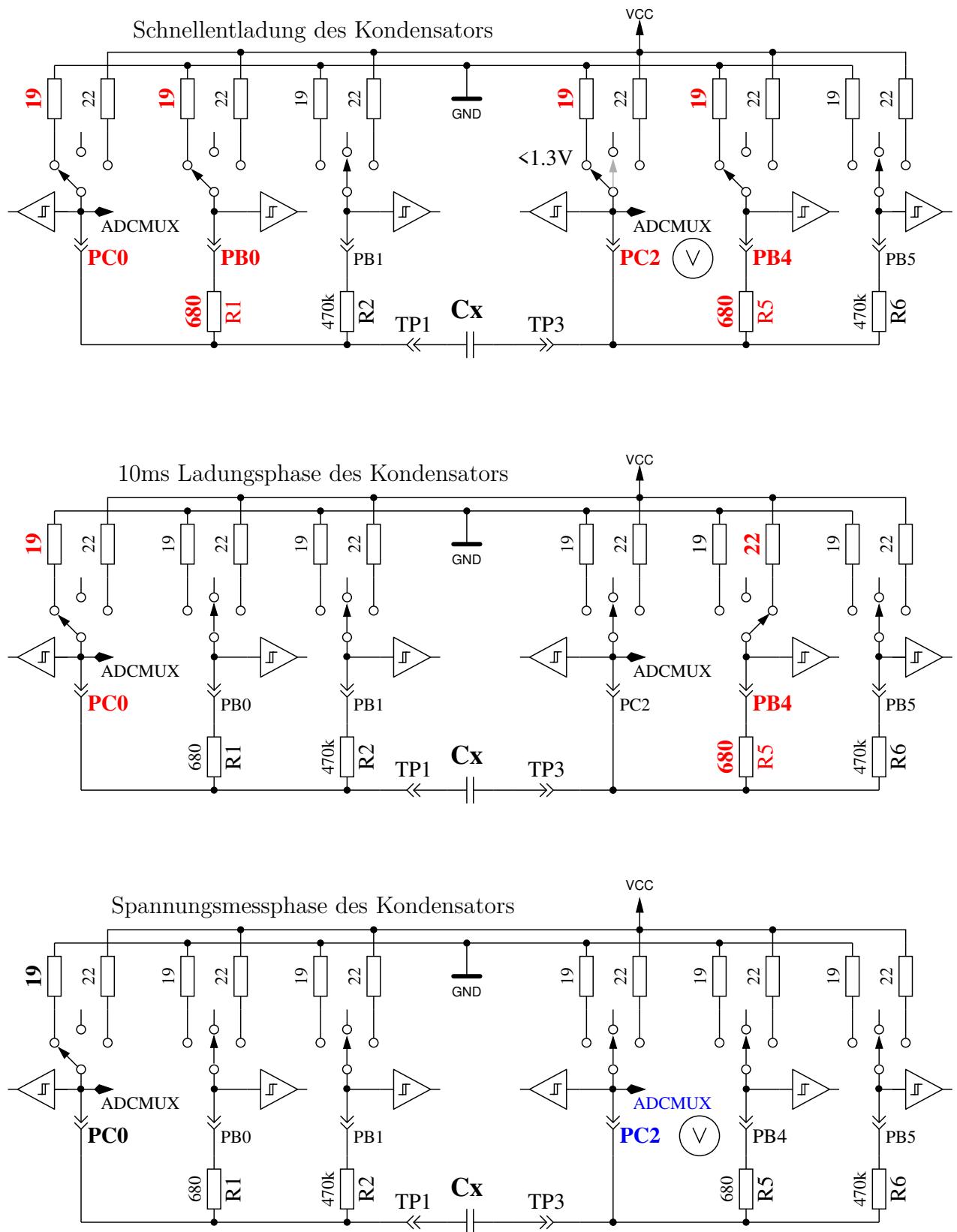


Abbildung 5.24. Kondensator-Entladung und -Ladung mit 10ms Ladepulsen bis zur Spannung > 300mV

Wegen der niedrigen Ladespannung wird die Messung viel schneller als bei der ursprünglichen Softwareversion, weil dieser Vorteil auch bei der Entladung wirkt. So können auch grössere Kondensatoren gemessen werden. Zusätzlich stört in den meisten Fällen eine parallel geschaltete Diode nicht die Messung, weil die Schwellspannung der Diode nicht erreicht wird. Ein Kniff wird seit der Version

1.12k der Software angewandt, um die Restspannung des Kondensators vor der Messung zu erfassen. Die Restspannung kann je nach Vorgeschichte positiv oder auch negativ sein. Negative Spannungen kann der ADC aber nicht messen. Deswegen wird die Spannung des negativen Testpin mit dem 680Ω Widerstand auf etwa $132mV$ angehoben wie in Abbildung 5.25 gezeigt wird. Die Restspannung des Kondensators kann nun mit einer Differenzbildung der Spannungen auf beiden Seiten des Kondensators gebildet werden. Die Spannung am positiven Testpin bleibt so auf jeden Fall positiv, auch wenn der Kondensator eine negative Restspannung von einigen mV hat.

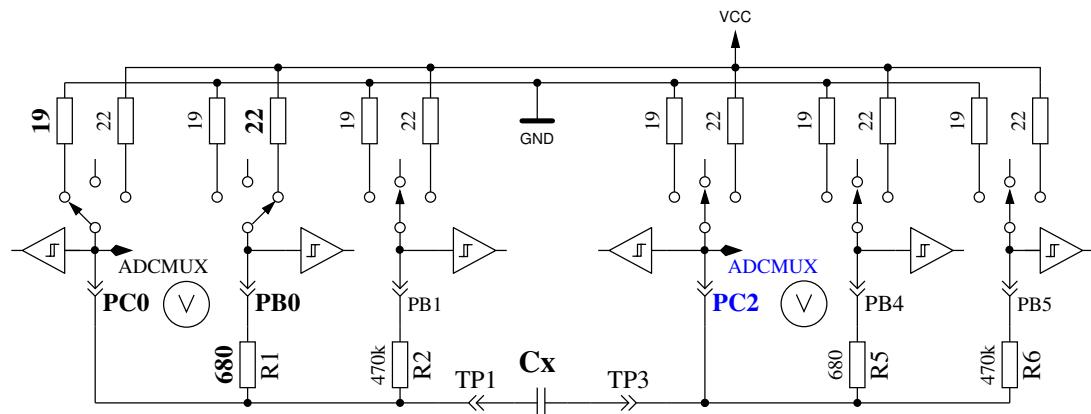
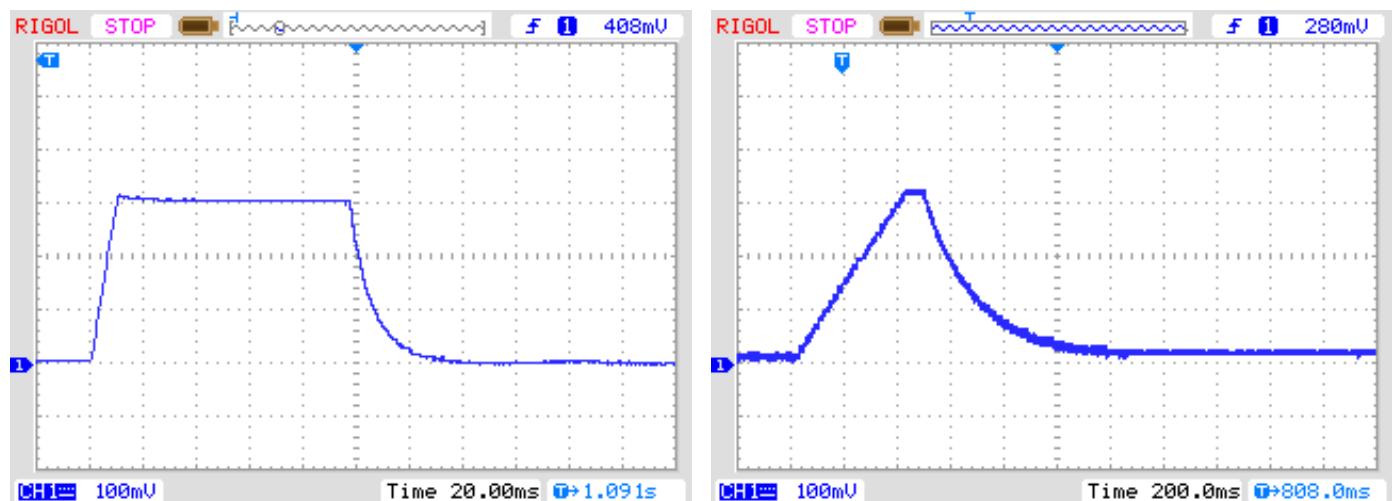


Abbildung 5.25. Messung der Kondensator-Restspannung bei Ladebeginn

Abbildung 5.26a zeigt das Laden und Entladen eines $229\mu F$ großen Kondensators. Das flache Dach der Messkurve bis zum Entladebeginn ist durch die Messzeit und Berechnungszeit des ATmega verursacht. Abbildung 5.26b zeigt die gleiche Messung mit einem $5mF$ -Kondensator, beachte wie die Messzeit inklusive Entladung auf ungefähr 1,5 Sekunden angewachsen ist. Das letzte Beispiel in Abbildung 5.27 zeigt die Messung eines $15mF$ -Kondensators.



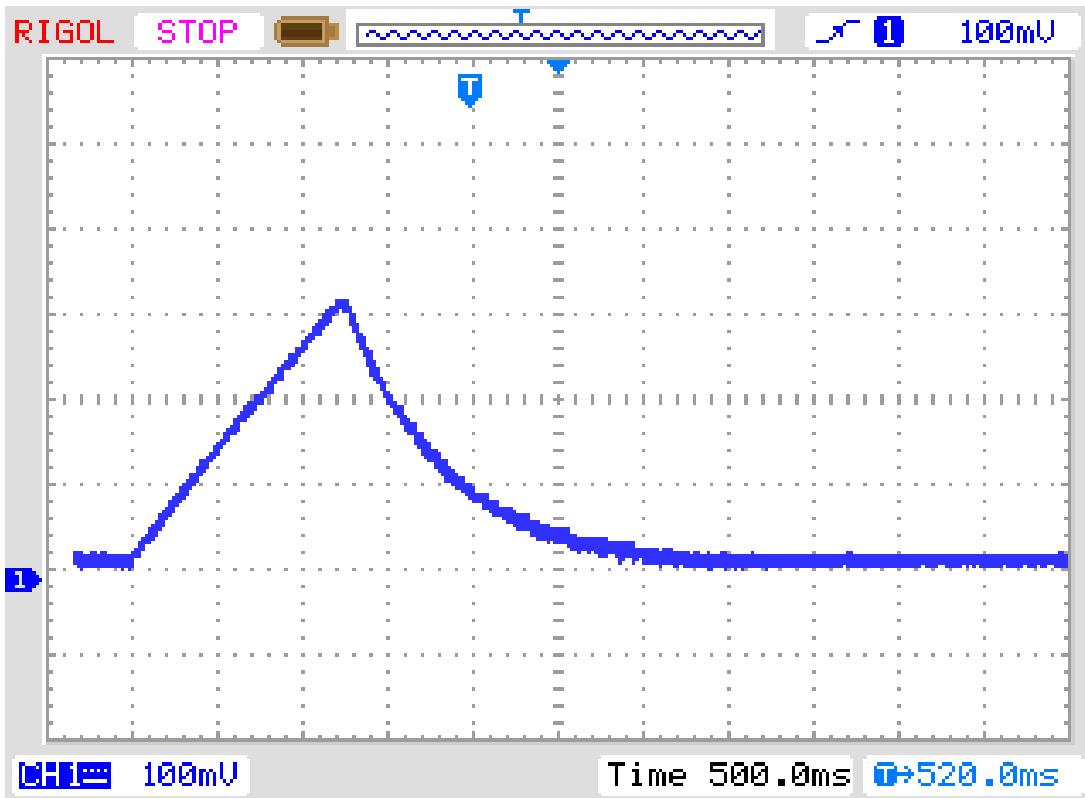


Abbildung 5.27. Laden und Entladen eines $15\mu F$ -Kondensators für die Messung

Nach dieser Kondensatormessung wird die Selbstentladung des Kondensators untersucht, indem der Spannungsverlust in einer der Ladezeit proportionalen Zeit untersucht wird. Der gemessene Kapazitätswert wird entsprechend korrigiert. Ein Test mit einer Parallelschaltung von einem $68\mu F$ -Kondensator mit einem $2,2k\Omega$ -Widerstand zeigt die Wirksamkeit der Methode. Der ermittelte Kapazitätswert ohne den Widerstand beträgt $66,5\mu F$, mit dem parallelen $2,2k\Omega$ Widerstand wird ein Kapazitätswert von $65,3\mu F$ gemessen. Zum Vergleich möchte ich die entsprechenden Ergebnisse mit einem Peaktech 3315 Multimeter angeben. Ohne den Widerstand wird eine Kapazität von $68,2\mu F$ gemessen, mit dem parallelen $2,2k\Omega$ Widerstand wird aber $192\mu F$ mit dem Multimeter gemessen.

5.3.3 Messen von kleinen Kapazitäten

Wenn der erste $10ms$ Ladepuls den Kondensator zu hoch aufgeladen hat, wird eine andere Messtechnik benutzt. Der ATmega-Mikrocontroller hat einen eingebauten 16-Bit-Zähler, der bei voller Taktfrequenz ($1MHz$ oder $8MHz$) arbeiten kann. Diese Zähler hat auch die Fähigkeit aufgrund eines externen Ereignisses den Zählerstand zu sichern. Dieses Ereignis kann auch das Ausgangs-Signal des Komparators sein. Der Komparator kann mit jedem ADC-Eingangspin und der Spannungsreferenz (Band Gap Reference) arbeiten. Das Schaltbild 5.28 zeigt ein vereinfachtes Diagramm der Messsituation. So entlade ich den Kondensator, bereite den Komparator für den richtigen Pin-Eingang vor, starte den Zähler bei 0 und starte sofort das Laden des Kondensators. Dabei ist eine Seite des Kondensators mit GND, die andere Seite über den $470k\Omega$ -Widerstand mit VCC verbunden. Nun prüfe ich in einer Programm-Schleife, ob der Zähler ein Überlauf-Ereignis (overflow) oder ein externes Ereignis (input capture) meldet. Die Überlauf-Ereignisse zähle ich, bis ich ein externes Ereignis feststelle. In diesem Fall halte ich den Zähler an und prüfe, ob ich noch einen zusätzlichen Überlauf zählen muss, weil der Zähler nicht mit dem externen Ereignis (input capture) angehalten werden kann.

Der Zählerstand des externen Ereignisses bildet zusammen mit dem Überlaufzähler die Gesamtzeit, aus der die Kapazität mit einem Faktor bestimmt wird. Die aktuelle Software kann eine Tabelle mit der theoretischen Abhängigkeit der Ladezeit in Bezug auf die gemessene Komparator-Spannung

berücksichtigen. Die Tabelle besitzt Einträge in Schritten von $50mV$, die Ergebnisse werden entsprechend der aktuellen Referenz-Spannung interpoliert. Diese Tabelle wird nur mit der Makefile Option WITH_AUTO_REF aktiviert. Vom Kapazitätswert ziehe ich eine experimentell herausgefundenen vordefinierte Konstante oder eine im Selbsttest herausgefundenen Konstante ab, um den Messwerte-Offset zu beseitigen. Ich weiß nicht, ob die vordefinierte Konstante für andere Leiterplatten angepasst werden muss. Mit einem Selbsttest bei gesetzter AUTO_CAL-Option wird diese Anpassung automatisch erledigt.

Ich habe bemerkt, dass die Referenz-Spannung meistens etwas zu klein gemessen wird, deshalb kann man einen Zusatz mit der Makefile Option REF_C_KORR angeben. Nach der Kalibration mit der AUTO_CAL-Option ist der REF_C_KORR-Wert nur ein Zusatz zu der automatisch gefundenen Differenz zwischen geladener Kondensator-Spannung und der internen Referenz-Spannung. Die gemessene Referenz-Spannung wird dann mit dem Korrekturwert (in mV) korrigiert (addiert). Wenn die Option WITH_AUTO_REF nicht benutzt wird, werden die Referenz-Spannungen entsprechend den Angaben in den Datenblättern [4] [5] der ATmega8, ATmega168 und ATmega328 berücksichtigt. Eine Beispielmessung von diesem Typ ist in Abbildung 5.29 dargestellt. Die Messzeit für den $22\mu F$ -Kondensator beträgt ungefähr $2,6s$ weil der $470k\Omega$ -Widerstand für das Laden benutzt wird. Aber das Entladen geht in diesem Fall viel schneller als das Laden.

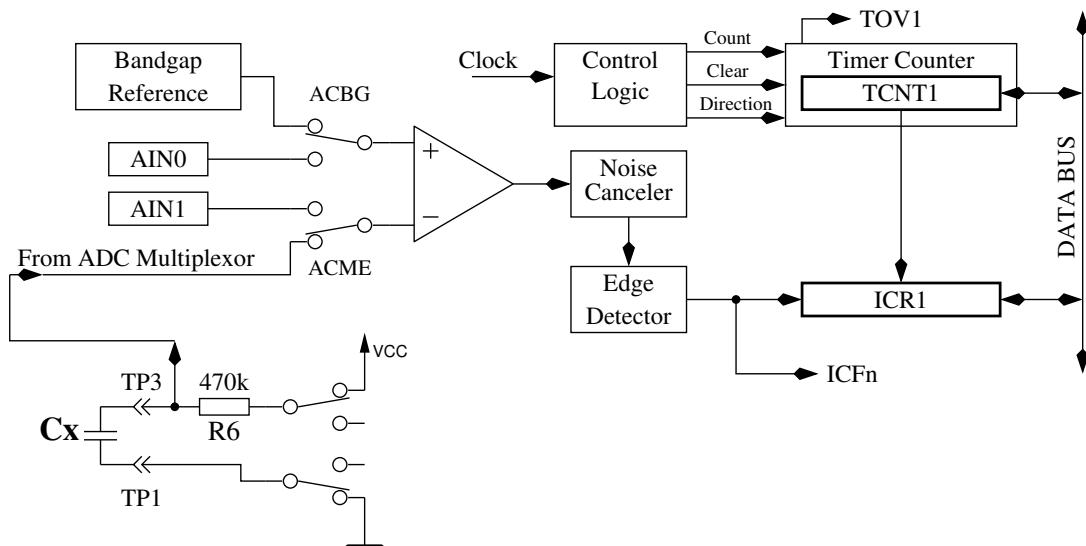


Abbildung 5.28. Messung kleiner Kapazitätswerte mit dem Komparator

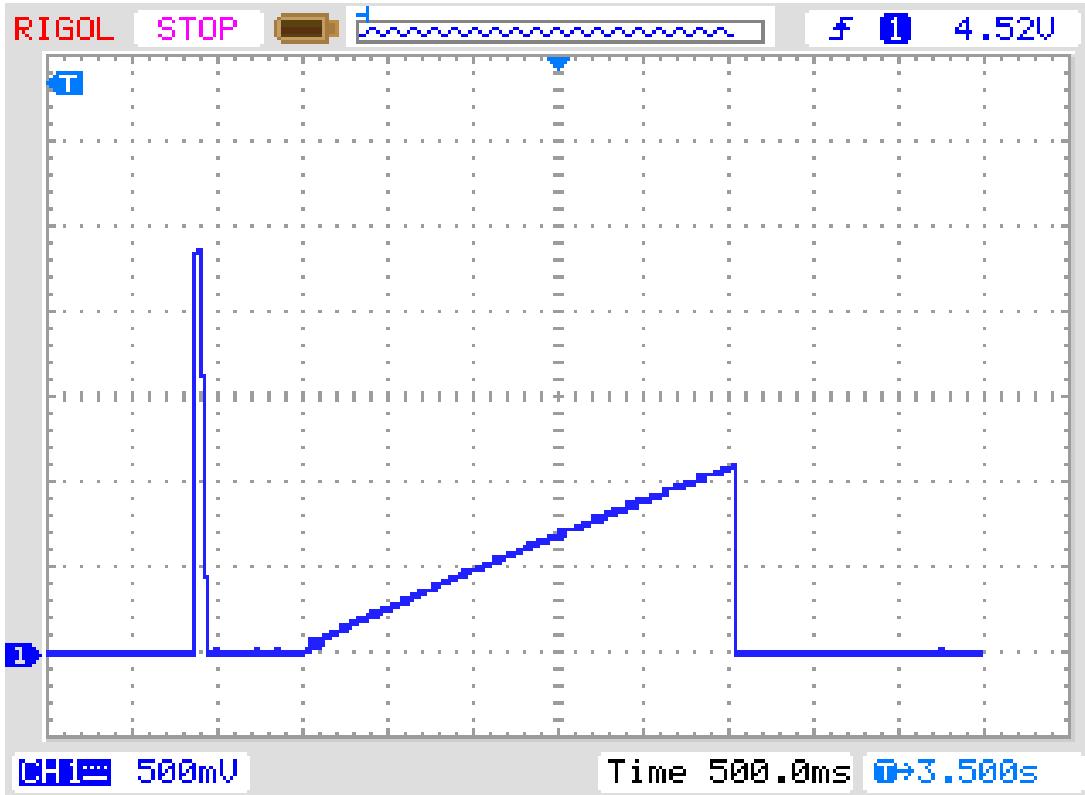


Abbildung 5.29. Laden und Entladen eines $22\mu F$ -Kondensators für die Messung

Im Prinzip kann diese Technik auch mit dem 680Ω -Widerstand benutzt werden, aber weil während dem Komparatorbetrieb der ADC nicht benutzt werden kann, besteht keine Möglichkeit die Ladespannung zu beobachten bis der Komparator angehalten wird. Wenn eine unentdeckte Diode mit dem Kondensator parallelgeschaltet ist, kann der Ladestrom von der Diode aufgenommen werden (Schwellspannung) und die Referenz-Spannung würde nie erreicht. Dieser Konzeptfehler wird mit der Methode vermieden, die in der aktuellen Software für große Kondensatoren in Kapitel 5.3.2 verwendet wird.

5.3.4 Messen sehr kleiner Kapazitäten mit der Samplingmethode

Der Funkamateuer Pieter-Tjerk (PA3FWM) hat für die Messung sehr kleiner Kapazitäten ($< 100\text{pF}$) mit hoher Auflösung die Fähigkeiten des Testers um die Sampling ADC Methode erweitert. Die Wandlungsdauer des ADC reicht eigentlich nicht aus, um schnelle Vorgänge abtasten zu können. Bei der Abtastung der Spannung wird das Eingangssignal aber zu einer exakt festgelegten Zeit abgetastet, der Sample und Hold Zeit (SH). Der ADC braucht für eine komplette Wandlung 13 Takte, wobei der ADC Takt durch Teilung des Prozessortaktes durch 128 oder 64 gewonnen wird. Die Abtastung des Spannungswertes findet immer genau bei ADC-Takt Nummer 1.5 statt. Wenn nun das abzutastende Spannungssignal immer wieder neu erzeugt werden kann, kann man durch Verschiebung der Abtastzeit das komplette Signal mit dem Raster der Verschiebung abtasten. Eine normale ADC-Wandlung dauert bei 8 MHz Prozessortakt $13 \times 64 = 832$ Taktzyklen. Wenn das Signal nun mit einem 831 Takten Zyklus wiederholt wird, würde die ununterbrochene ADC-Wandlung (Freilauf-Modus) das Signal bei jeder fortlaufenden Wandlung einen Prozessortakt später abtasten. Mit dieser Methode muß dafür gesorgt werden, daß die erste ADC-Abtastung des Signals zu der gewünschten Startzeit beginnt. Die Zeiten der fortlaufenden ADC Werte verschieben sich dann jeweils um einen Prozessortakt relativ zum neu erzeugten Signal. Wenn die Wiederholung des Signals gut gelingt, entspricht das aus vielen Signalwiederholungen zusammengesetzte Signal dann dem ADC Signal, das mit Prozessortakt (8 MHz) abgetastet und gewandelt würde. Die Abbildung 5.30 zeigt das Prinzip der Abtastung mit einem zehnmal wiederholten Signal um daraus 10 Abtastwerte (SH0 - SH9) zu gewinnen. Für den

wirklichen Fall ist die relative Zeitverschiebung der aufeinanderfolgenden Abtastungen viel geringer als dargestellt.

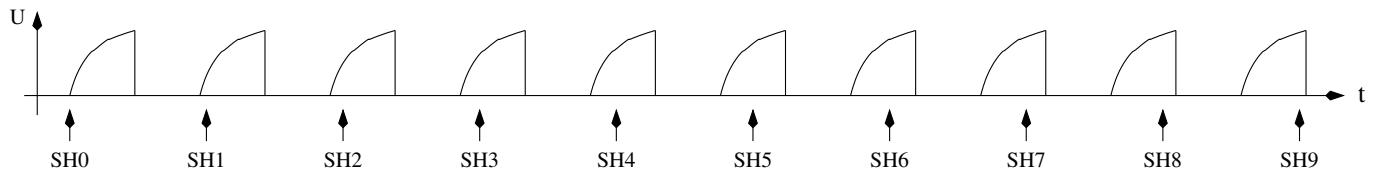


Abbildung 5.30. Abtasten eines Spannungsverlaufs mit der Samplingmethode

Eine Schwierigkeit für die genaue Festlegung der ersten Abtastzeit ergibt sich dadurch, daß der Teiler für die Takterzeugung des ADC nur mit einem externen Signal neu gestartet werden kann. Bei Start des ADC aus dem Programm läuft der Teiler einfach weiter. Die Einhaltung der exakten Start- und Perioden-Zeiten ist nur mit einer Funktion in Assemblersprache möglich, da es auf jeden Prozessortakt ankommt. Bei der Abtastung der Ladekurve von kleinen Kondensatoren mit der Sampling Methode hat sich auch gezeigt, daß die Zeitkonstante während der Abtastperiode nicht konstant bleibt. Das hat Pieter-Tjerk bei einem Vortrag auf der 60. UKW-Tagung in Weinheim dargestellt. Der Kondensator von etwa 10 pF, der die Spannung für die ADC-Wandlung festhält, wird zur SH-Zeit für die Wandlung abgekoppelt und etwa zwei ADC-Takte später wieder angekoppelt. Daneben gibt es einen halben ADC-Takt vorher noch eine kleine Unregelmäßigkeit in den Daten, die wahrscheinlich vom Schalter des Multiplexors verursacht wird. Beide Störungen werden von der Software berücksichtigt. Die Sampling Software kann 255 Werte digitalisieren, wobei für die Abtastung der Ladekurven auch der Mittelwert über 32 Digitalisierungen gebildet werden kann. Damit können Störeinflüsse etwas geringer gehalten werden. Die Software kann sowohl einen Ladevorgang als auch einen Entladevorgang vermessen. Weil bei der Messung der Sperrsichtkapazität einer Diode beide Vorgänge gemessen werden, wird die Kalibration der Nullkapazität für beide Richtungen und alle Pinkombinationen durchgeführt. Mit der Messung der Kapazitäten beim Lade- und Entlade-Vorgang kann die Abhängigkeit der Sperrsichtkapazität von der Spannung gezeigt werden, da der Ladevorgang die Kapazität in der Nähe von 0V und der Entladevorgang die Kapazität in der Nähe von 5V misst. Ein normaler Kondensator sollte bei so kleinen Spannungen keine Kapazitätsdifferenz zeigen. Hier wird deshalb die Kapazität nur beim Ladevorgang gemessen. Pieter-Tjerk hat die Funktion für 16 MHz Betrieb optimiert. Hierbei wird eine Auflösung von 0.01 pF erreicht. Bei 8 MHz Betrieb läuft der ADC-Wandler für die Sampling Methode mit der halben Taktrate, damit die vorhin erwähnten Störeinflüsse an den gleichen Datenpositionen liegen. Der Auflösungsverlust durch den 8 MHz Betrieb spielt wohl meistens keine Rolle und auch die Meßzeit bleibt im erträglichen Rahmen.

5.3.5 Messen des äquivalenten Serienwiderstandes ESR

Der ESR [10] stellt zum Beispiel ein Maß für die Alterung von Elektrolyt-Kondensatoren dar. Die Abbildung 5.31 zeigt ein Ersatzschaltbild eines Kondensators. Der Widerstand R_p stellt den Isolationswiderstand des Kondensators dar, ESL die äquivalente Serieninduktivität und der Widerstand ESR stellt den äquivalenten Serienwiderstand dar.

Es ist in Datenblättern üblich, den ESR gemessen bei einer Frequenz von 100kHz und einer Temperatur von 20°C anzugeben. Die Abbildungen 5.32 und 5.33 zeigen die vom Hersteller Panasonic angegebenen ESR-Werte für die Elektrolytkondensatoren verschiedener Betriebs-Spannungen für die FC- und die „low ESR“ FR-Reihe. Beide Serien sind für eine maximale Temperatur von 105°C ausgelegt. Die Abbildung 5.34 stellt die

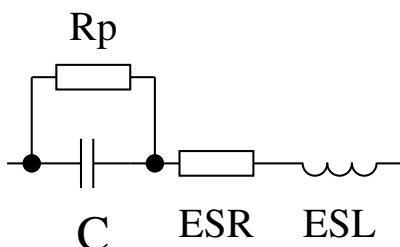


Abbildung 5.31. Kondensator-Ersatzschaltbild

angegebenen Daten beider Serien bei einer zulässigen Betriebsspannung von 25V dar. Wenn in der Baureihe verschiedene Ausführungen mit gleicher Kapazität und Spannung zur Verfügung stehen, werden die Daten mit dem niedrigsten ESR dargestellt. Bei Elektrolytkondensatoren sind sowohl der ESR-Wert als auch die Kapazität relativ stark von der Betriebstemperatur abhängig.

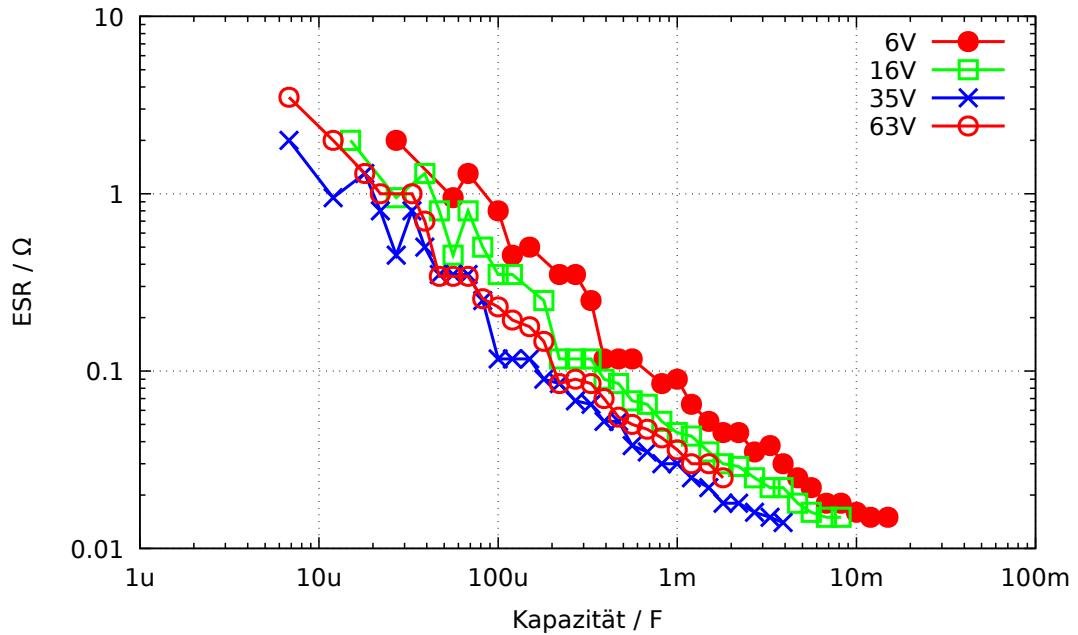


Abbildung 5.32. ESR-Werte aus dem Panasonic Datenblatt der Serie FC

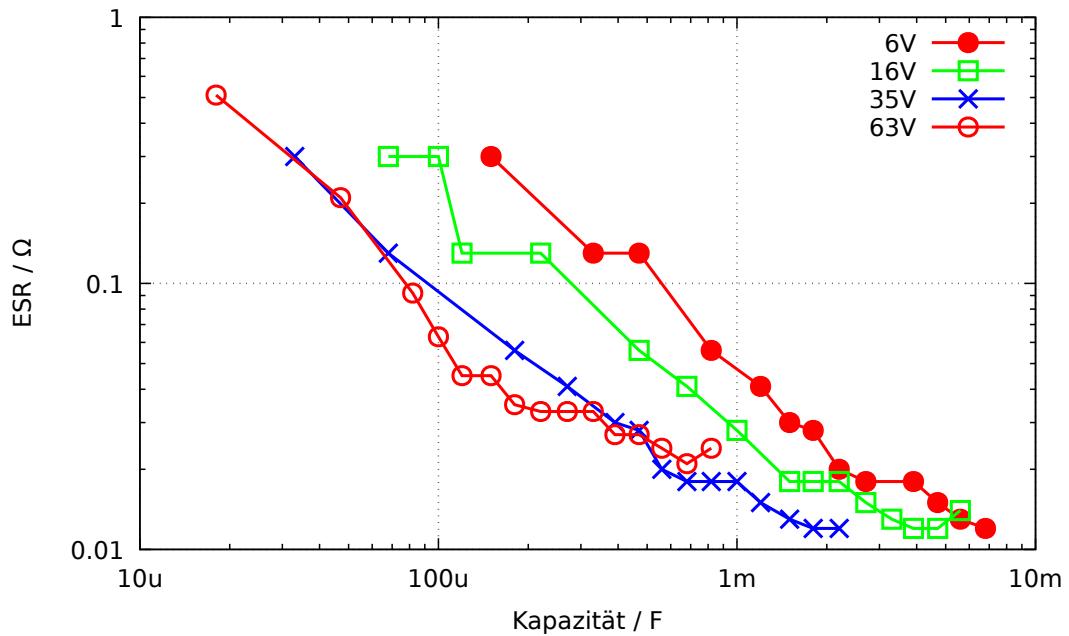


Abbildung 5.33. ESR-Werte aus dem Panasonic Datenblatt der Serie FR

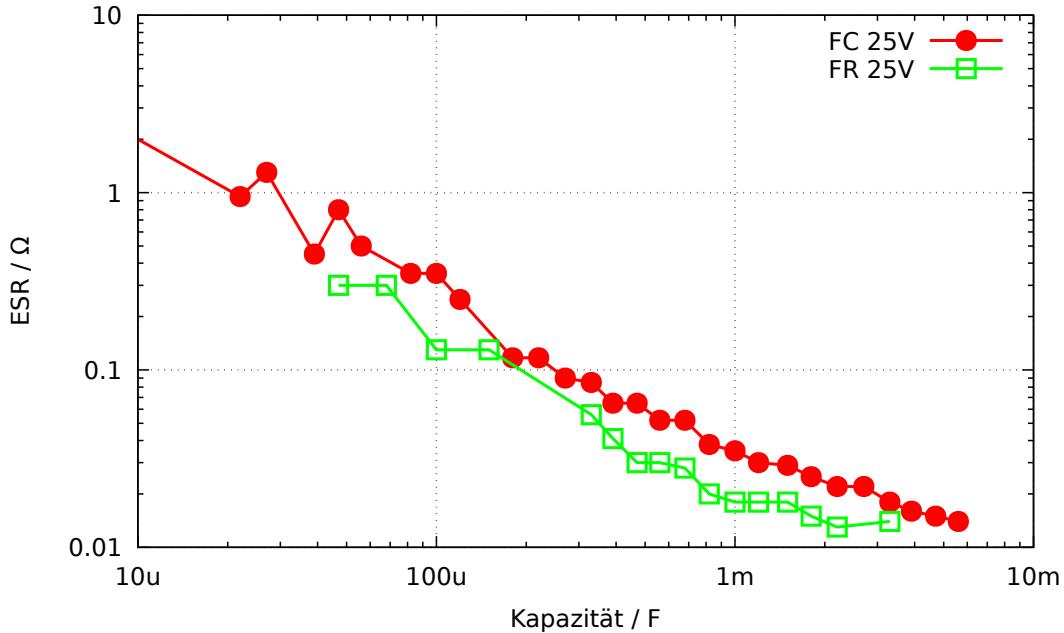


Abbildung 5.34. Vergleich der ESR-Werte der Serie FC mit der Serie FR

Eine Messung bei 100kHz ist mit der Hardware des ATmega nicht ohne weiteres möglich, da weder der ADC eine direkte Abtastung einer so hohen Eingangsfrequenz zuläßt, noch ein 100kHz Signal bei der vorhandenen Schaltung zur Verfügung steht. Als nächstes werden hier zwei Methoden für die Bestimmung des ESR vorgestellt, die mit der vorhandenen Schaltung auskommen. Beide Methoden verwenden für die Messung ein rechteckförmiges Messsignal, sodass die Ergebnisse nie exakt mit den Angaben übereinstimmen können, die bei einem sinusförmigen Signal gemessen werden. Bei der ersten Methode liegen die Ergebnisse nahe bei den ESR-Werten, die bei 1kHz ermittelt werden. Die zweite Methode hat aber den Vorteil, dass der Nullwert mit kurzgeschlossenen Testpins ermittelt werden kann und dass zusätzlich der ermittelte ESR sich dem bei 10kHz Messfrequenz gemessenen Wert nähert. Ich habe derzeit keine Idee für eine Messmethode, die einen direkt zu der 100kHz -Messung vergleichbaren ESR-Wert liefern kann.

Die folgende Tabelle 5.6 soll die Abhängigkeit des ESR von der Frequenz zeigen. Außer dem $47\mu\text{F}$ Kondensator sind alle Kondensatoren aus der Baureihe FC von Panasonic. Die Referenzwerte wurden mit einem Peaktech 2170 LCR-Messgerät ermittelt. Alle TransistorTester-Resultate in dieser Tabelle wurden mit der im Unterkapitel 5.3.7 beschriebenen Methode 2 gemessen. Bei großen Kapazitätswerten macht die ebenfalls vorhandene Induktivität ESL das Messen bei höheren Frequenzen wie 100kHz schwierig.

Kondensator	Datenblatt 100 kHz	PeakTech 100 kHz	Peaktech 10 kHz	PeakTech 1 kHz	Transistor- tester
1uF / 50V	2.4	1.27	1.75	4.31	2.1
2.2uF / 50V	1.8	1.07	1.34	2.76	1.6
4.7uF / 50V	1.3	1.19	1.40	2.37	1.5
4.7uF / 50V	1.3	1.19	1.40	2.37	1.5
10uF / 50V	1.3	1.26	1.45	2.05	1.5
22uF / 10V	2.0	1.52	1.76	2.24	1.9
47uF / 63V	?	0.46	0.50	0.63	0.52

Tabelle 5.6. ESR-Werte verschiedener Elektrolyt-Kondensatoren

5.3.6 Messen des äquivalenten Serienwiderstandes ESR, Methode 1

Bei Kondensatoren mit mehr als $0,45\mu F$ wird versucht, den Serienwiderstand von Kondensatoren zu messen. Bei mehr als $3,6\mu F$ wird dazu die normale Taktrate von $125kHz$ für den Analog-Digital Wandler benutzt. Bei kleineren Kapazitäten wird eine erhöhte Taktrate von $500kHz$ benutzt, um die Messung zu beschleunigen. Die Genauigkeit der ADC-Ergebnisse wird bei der höheren Taktrate zwar schlechter, aber die größeren ESR-Werte bei kleineren Kondensatoren mildern die Auswirkung dieses Genauigkeitsverlusts ab. Andererseits ist sonst nach diesem Verfahren keine ESR-Messung für Kondensatoren unter $1,8\mu F$ möglich.

Genau genommen ist der ESR eines Kondensators abhängig von der Betriebsfrequenz und der Temperatur. Üblicherweise wird der mit einem sinusförmigen Signal bei $100kHz$ gemessene Wert in Datenblättern angegeben. Bei dieser Frequenz kann der ATmega ohne externe Beschaltung keine Messung durchführen. Das nachfolgend beschriebene Verfahren erreicht bei der normalen ADC-Taktrate nur eine Messfrequenz von unter $640Hz$ mit näherungsweise rechteckigem Signal. Bei $500kHz$ ADC-Taktrate wird etwa $2400Hz$ Messfrequenz erreicht. Um den äquivalenten Serienwiderstand zu bestimmen, wird der Kondensator zuerst in einer Richtung geladen und an beiden Anschlüssen die Spannung mit der internen Referenzspannung ($1,1V$) gemessen. Nach der Messung wird der Ladestrom abgeschaltet und die Spannung am Kondensator ohne den Ladestrom gemessen. Wenn die Spannung am Kondensator ohne Ladestrom weniger als $3mV$ beträgt, wird diese Messfolge wiederholt. Die Abbildung 5.35 zeigt die entsprechenden Schaltungen.

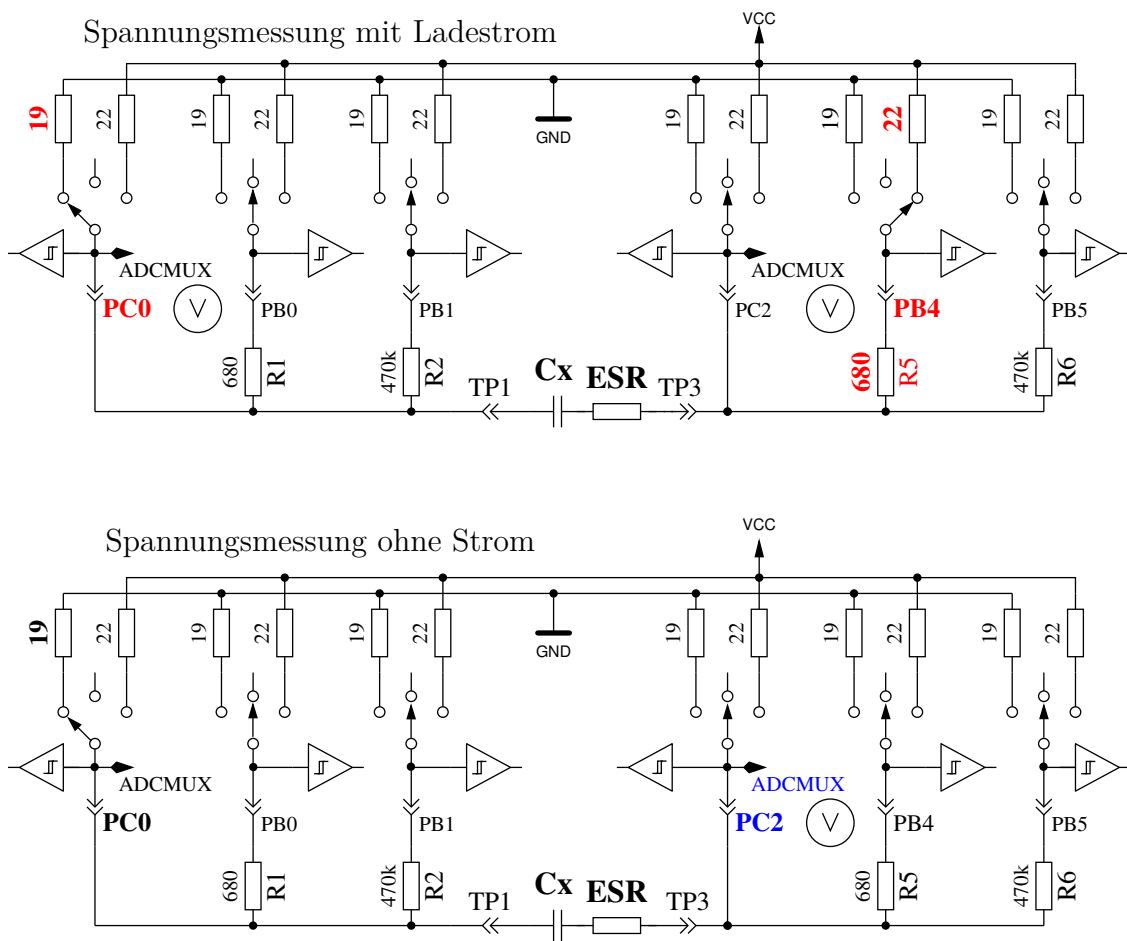


Abbildung 5.35. Schaltbild der ESR-Messung eines Kondensators

Die Differenz der Spannung am Kondensator mit und ohne Strom ist ein Maß für den internen Widerstand des Kondensators. Die zu erwartende Spannungsdifferenz ist allerdings so gering, dass

mit einer Messung kein brauchbares Ergebnis erzielt werden kann. Aus diesem Grund wird danach der Strom umgekehrt und die Messung in der anderen Richtung wiederholt. Die gesamte Messsequenz wird 128 Mal durchgeführt und die Ergebnisse der Spannungsmessungen addiert. Danach hat man drei Spannungssummen, die Spannung U_{lp} am Minuspol des Kondensators mit Strom, die Spannung U_{hp} am Pluspol des Kondensators mit Strom und die Spannung U_c am Pluspol des Kondensators ohne Strom. Die Spannungssumme am Minuspol des Kondensators repräsentiert den Spannungsabfall bei einem mittleren Ladestrom am Port-Ausgangswiderstand R_{port} . Aus der Differenz der Spannungssumme am Pluspol und Minuspol des Kondensators hat man ein Maß für die Spannung am Kondensator mit Ladestrom $U_{diff} = U_{hp} - U_{lp}$. Die Differenz $U_{esr} = U_{diff} - U_c$ soll jetzt den Spannungsabfall bei mittlerem Ladestrom am internen Widerstand des Kondensators repräsentieren. Der Widerstandswert wird aus dem Verhältnis dieser Spannung U_{esr} zu der Spannung U_{lp} skaliert mit dem bekannten Widerstandswert des Port-Ausgangs R_{port} . Dabei wird so skaliert, dass eine Widerstandauflösung von $0,01\Omega$ erreicht wird $Resr = \frac{U_{esr} \cdot 10 \cdot R_{port}}{U_{lp}}$. Die Abbildung 5.36 zeigt den Ausschnitt des Spannungsverlaufs eines $4,2\mu F$ Kondensators während der ESR-Messung. Dem Kondensator wurde ein $6,8\Omega$ Widerstand in Serie geschaltet, um den ESR-Einfluß zu verdeutlichen. Der kleine Spannungseinbruch nach dem Ladevorgang des Kondensators wird von der Software ausgewertet. Der größere Spannungseinbruch bei der Messung gegen GND kommt durch den Einfluß des Port-Ausgangswiderstands von etwa 20Ω zustande. Das Ergebnis der ESR-Messung ist in diesem Fall $7,5\Omega$, ohne den $6,8\Omega$ Widerstand sind es $0,56\Omega$. Die Abbildung 5.37 zeigt die gleiche Messung mit höherer Messfrequenz bei einem $2,2\mu F$ Elko mit einem ESR von 6.5Ω .

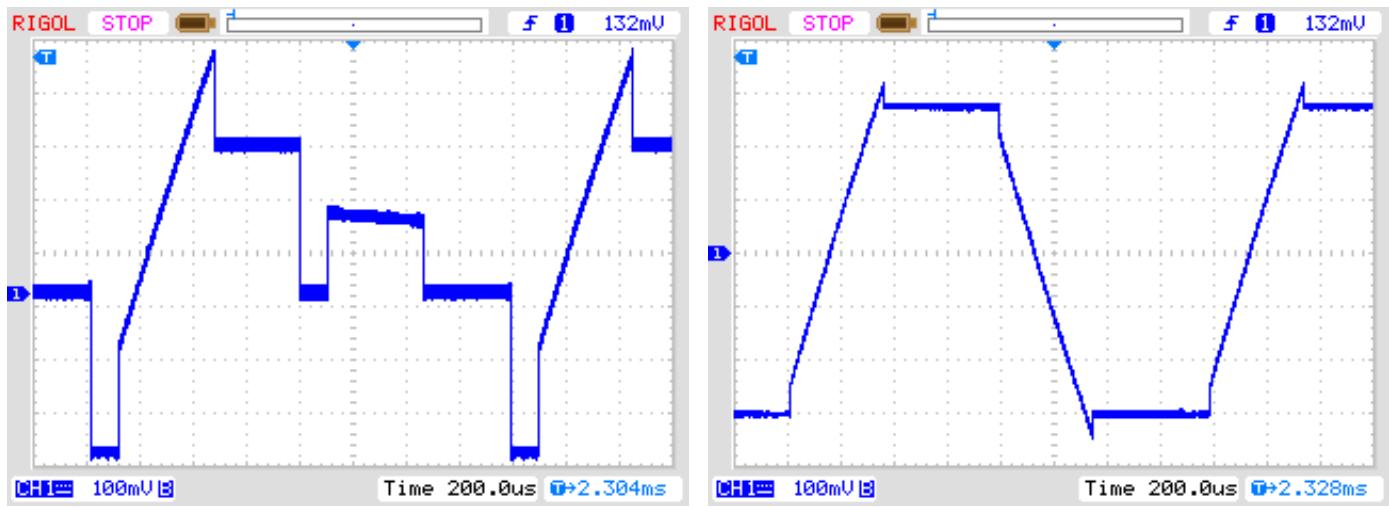
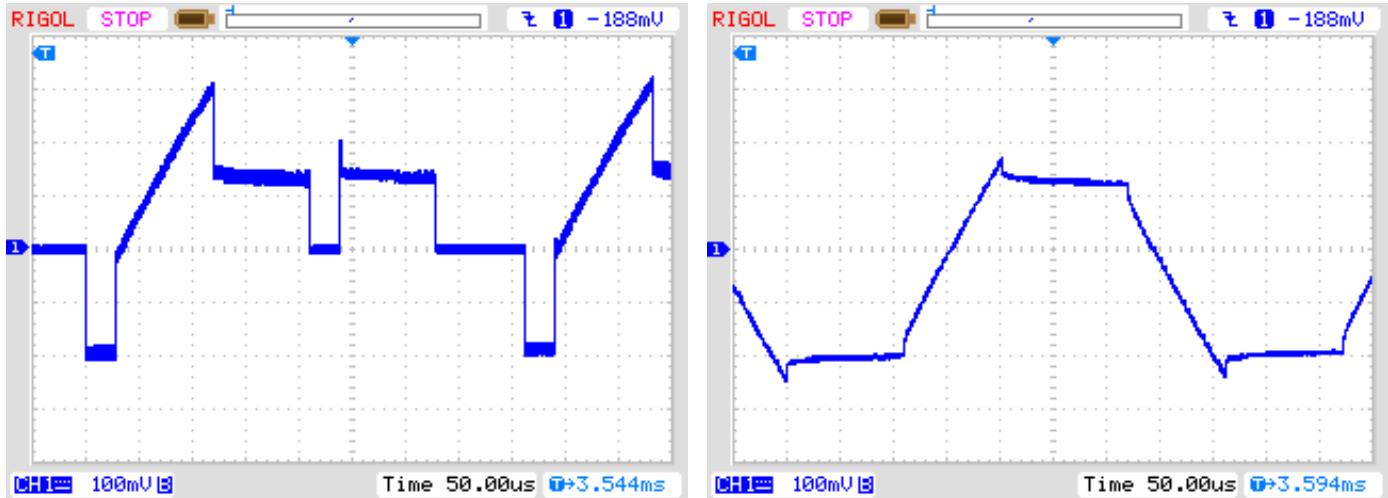


Abbildung 5.36. Spannungsverlauf eines $4,2\mu F$ Kondensators während der ESR-Messung



(a) Ein Pin gegen GND gemessen
 (b) Von Pin zu Pin gemessen
 Abbildung 5.37. Spannungsverlauf eines $2,2\mu F$ Kondensators während der ESR-Messung

Die Genauigkeit der ESR-Messung ist aus verschiedenen Gründen nicht sehr hoch:

1. Die Spannungsmessung an den beiden Kondensator-Anschlüssen kann nicht gleichzeitig sondern nur nacheinander durchgeführt werden. In der Zwischenzeit hat sich der Ladestrom durch den Ladevorgang des Kondensators geändert. Dies wird versucht mit einer kapazitätsabhängigen Korrektur der Minuspol-Spannung auszugleichen.
2. Der ADC nimmt den Spannungswert 1,5 Takte nach dem Start des Wandlungsvorgangs. Der Wandelvorgang beginnt mit der steigenden Flanke des ADC-Taktes, wenn das Startbit gesetzt ist. Wenn der Ladestrom des Kondensators zu früh abgeschaltet wird, nimmt der ADC die falsche Spannung für die strombehaftete Messung auf. Wird der Ladestrom zu spät abgeschaltet, wird der Kondensator weiter geladen als es der strombehafteten Messung entspricht. Dann wird im stromlosen Zustand eine zu hohe Spannung gemessen. Es ist aber schwierig im Programm den genauen Zeitpunkt für die Stromabschaltung zu treffen.
3. Der Port-Ausgangswiderstand wird bei dieser Messmethode als Referenz benutzt, dessen Widerstandswert ist aber auch nicht exakt bekannt.
4. Die Auflösung des ADC reicht nicht aus, um eine Widerstandsauflösung von $0,01\Omega$ zu erreichen. Für alle Messungen wird die interne Spannungsreferenz ($1,1V$) benutzt, um die best mögliche Auflösung zu verwenden. Zusätzlich wird versucht, den Auflösungs-Mangel durch eine große Zahl von Einzelmessungen abzumildern.
5. Mit der Abfrage des Fertig-Signals der ADC-Wandlung gelingt es nicht exakt, die Schaltzeiten der Port-Ausgänge mit dem ADC-Takt zu synchronisieren.

Trotz allen Schwierigkeiten scheinen die Ergebnisse brauchbar zu sein, wie die nachfolgende Abbildung 5.38 zeigt. Die ESR-Werte eines Bauteils gemessen mit dem Transistortester schwanken stärker als die Messungen des LCR-Meters. Die Messergebnisse des LCR-Messgerätes wurden bei einer Frequenz von $1kHz$ gemessen oder für kleine Kapazitäten auf $2,4kHz$ interpoliert. Beim Transistortester muss auf die Qualität der Anschlüsse geachtet werden. Verwendete Anschlusskabel erhöhen den gemessenen Widerstandswert. Auch die Kontakte von Steckverbindern können die gemessenen Widerstandswerte erhöhen. Das LCR-Messgerät macht hier wegen den verwendeten Kelvin-Klemmen weniger Probleme. Bei den Kondensatoren mit einer Kapazität unter $1\mu F$ war ein $500nF$ keramischer Kondensator (Z5U), alle anderen waren Folien-Kondensatoren. Der einzige Elektrolyt-Kondensator der Messreihe unter $9\mu F$ ist ein $2,2\mu F$ Kondensator.

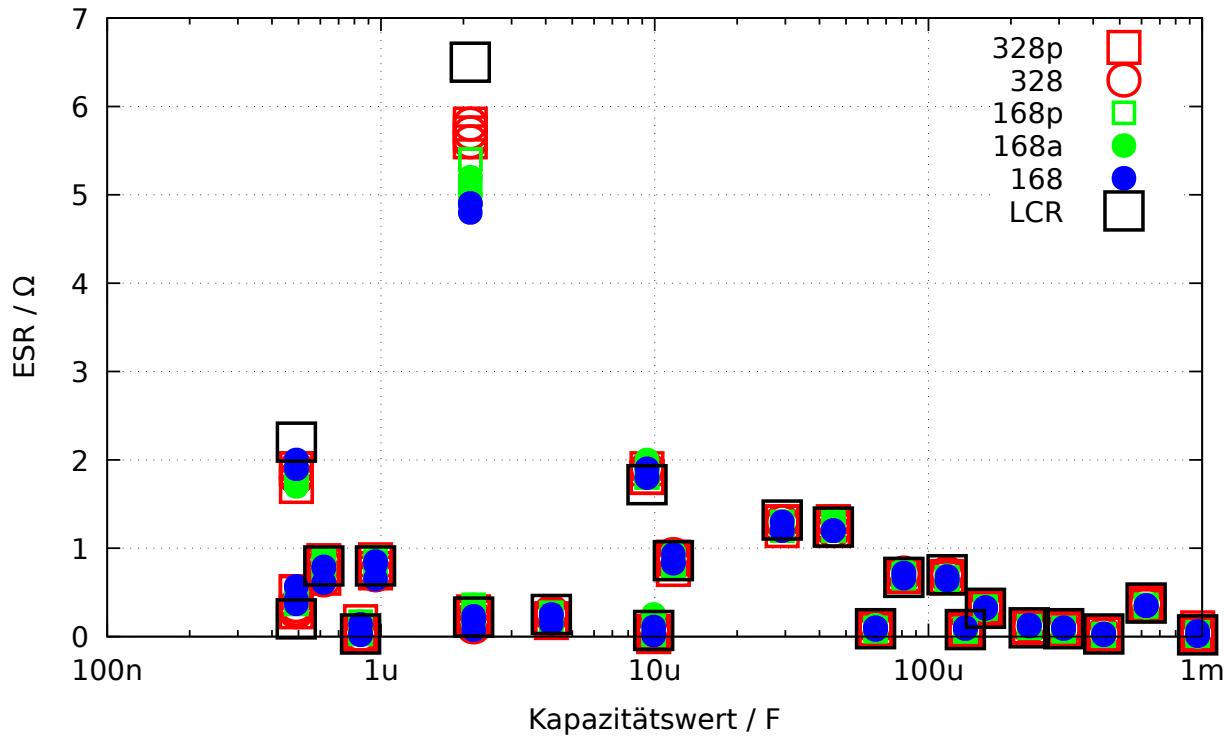


Abbildung 5.38. Messergebnisse der ESR-Messungen mit 15 verschiedenen ATmega

5.3.7 Messen des äquivalenten Serienwiderstandes ESR, Methode 2

Ab der Softwareversion 1.07k wurde die ESR-Messung auf eine modifizierte Messmethode umgestellt. Die einzelnen Messschritte sind in Abbildung 5.40 gezeigt. Das Besondere an der Methode ist, dass die Zeit des Stromflusses durch den Kondensator wesentlich gegenüber der Methode 1 verringert wurde. Der Kondensator wird mit einem Puls der halben Breite negativ vorgeladen und wird dann zyklisch aufgeladen und in der Gegenrichtung wieder entladen. Dabei wird der Ladepuls zeitlich so gelegt, dass bei Sample 4 und 8 zur Pulsmitte die Spannung für den ADC abgegriffen wird (Sample and Hold, 2,5 ADC-Takte nach dem Start). Ein kompletter Messzyklus wird in Abbildung 5.39 gezeigt. Es werden auch bei dieser Messmethode die Summen der Messergebnisse aus 255 Zyklen ausgewertet, um eine ausreichende Auflösung zu erreichen. Eine fortlaufende Aufladung des Kondensators in die eine oder andere Richtung wird durch die kurzen und gleich langen Auflade- und Entlade-Pulse bei gleicher Beschaltung weitgehend vermieden. Bei der Messung der Referenzspannung fließt kein Strom durch den Kondensator. Dadurch ist diese Messung nicht zeitkritisch. Es wird lediglich vorausgesetzt, dass der Kondensator seine Spannung in der stromlosen Zeit beibehält.

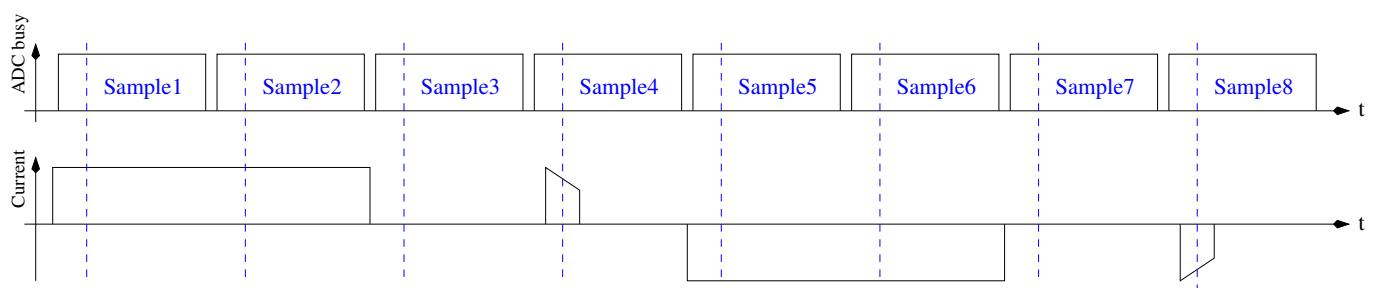


Abbildung 5.39. Zeitlicher Ablauf eines Messzyklus der neuen ESR-Messung

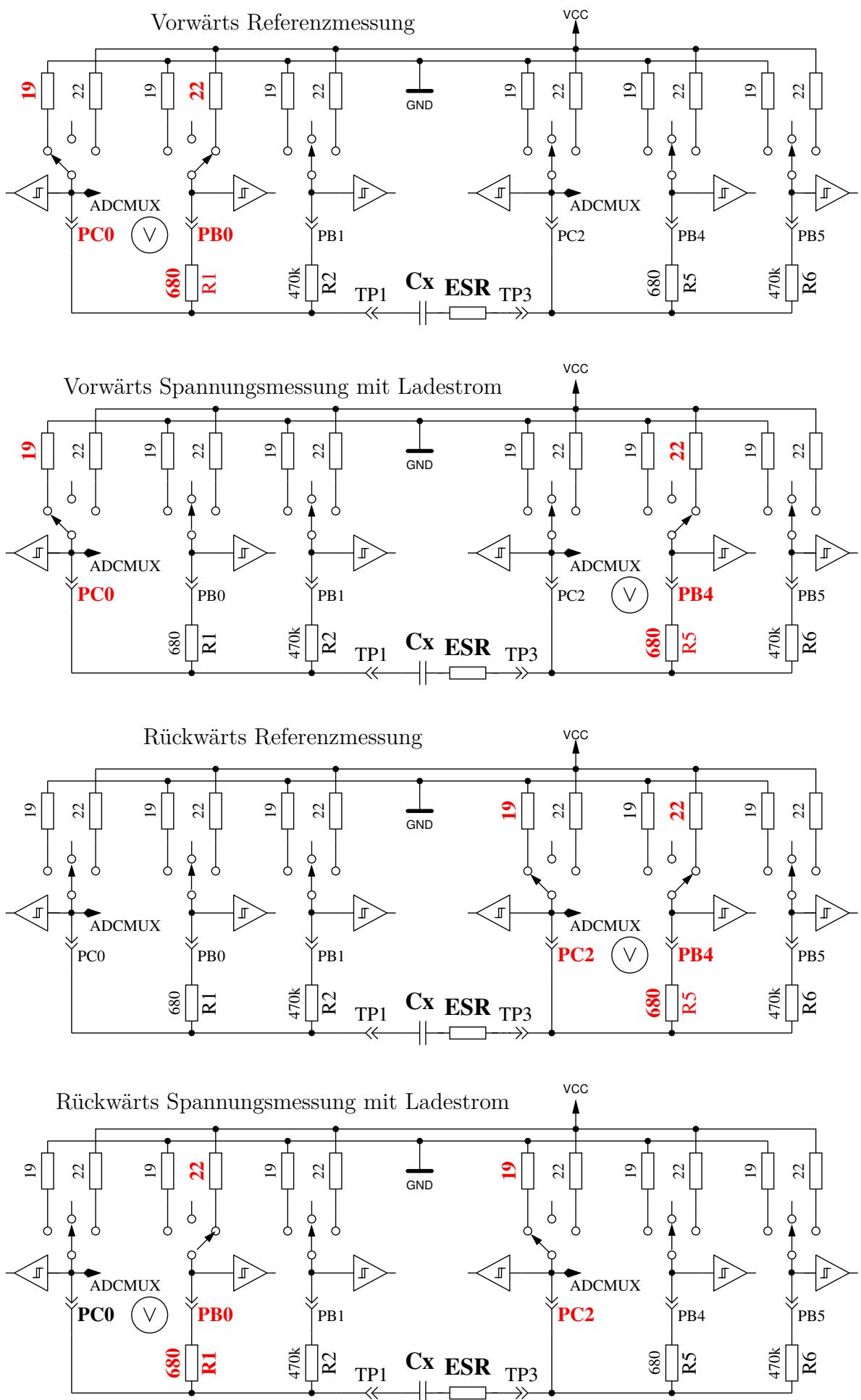


Abbildung 5.40. Vereinfachte ESR-Messung eines Kondensators

Durch den kürzeren Ladepuls kann nicht nur der ESR von geringeren Kapazitätswerten gemessen werden, sondern diese Methode kann auch für die Messung kleiner Widerstandswerte benutzt werden, wenn die Widerstände keine messbare Induktivität besitzen. Dadurch kann die Auflösung bei Widerstandswerten unter 10Ω auf $0,01\Omega$ gesenkt werden. Daneben kann auch der Nullwiderstand für die Widerstandsmessung als auch für die ESR-Messung im Selbsttestzweig der Kalibration für alle drei Testpin-Kombinationen bestimmt werden. Auf solide Steckverbindungen oder Klemmverbindungen sollte für stabile Ergebnisse geachtet werden. Die Messperiode beträgt etwa $900\mu s$, was einer Frequenz von etwa $1,1kHz$ entspricht. Wegen der Kürze der Ladepulse entspricht das Ergebnis aber eher einer Messung mit $10kHz$. Als Beispiel wird die Messung eines $10\mu F$ Folienkondensators einmal direkt und einmal mit einem $2,7\Omega$ Serienwiderstand in Abbildung 5.41 gezeigt. Deutlich kann man den Einfluß des zusätzlichen Widerstandes beim Vergleich der beiden Diagramme erkennen. Hier wird auch verständlich, warum die ADC-Messung in der Mitte des Ladepulses erfolgen muss. Bei großen Kondensatoren bleibt der Ladestrom des Kondensators hinreichend gut konstant, sodass bei der zeitlichen Mitte des Ladepulses auch die mittlere Spannung gemessen wird. Bei kleineren Kondensatoren ergibt sich ein deutlicherer Unterschied, der aber mit dem bekannten Kapazitätswert relativ gut kompensiert werden kann.

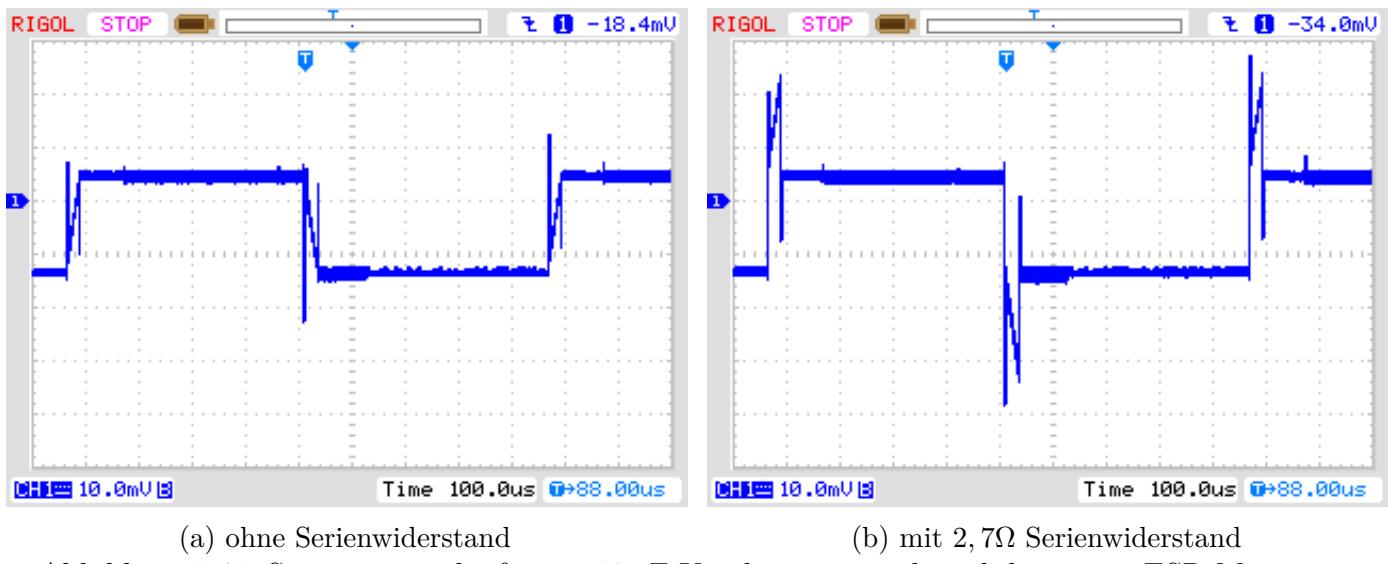
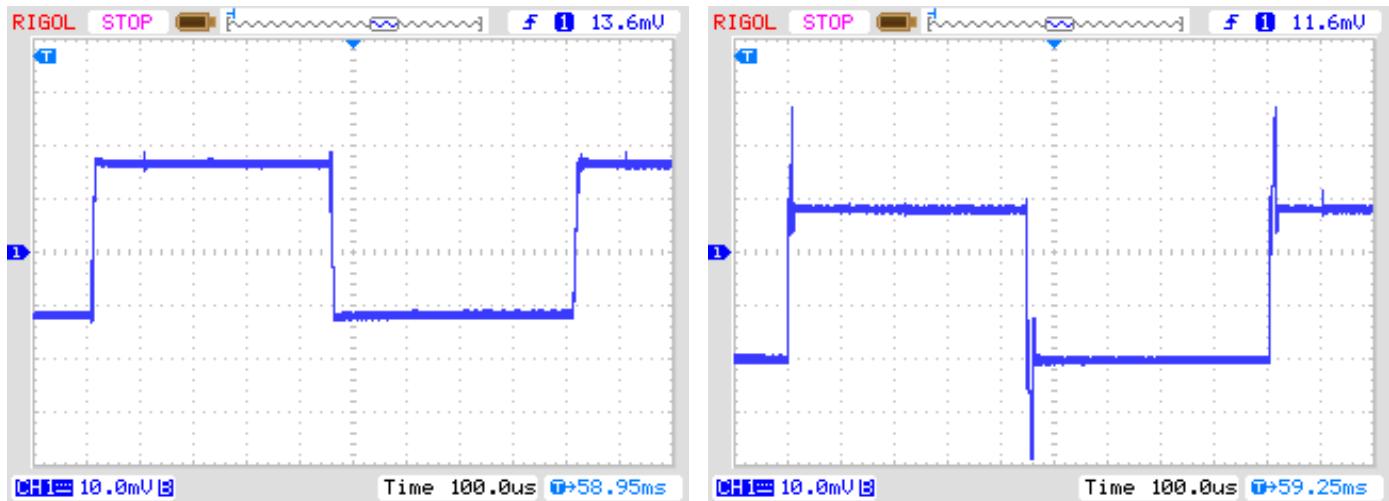


Abbildung 5.41. Spannungsverlauf eines $10\mu F$ Kondensators während der neuen ESR-Messung

Bei Verwendung von $27\mu s$ langen Strompulsen kann der ESR von Kondensatoren ab etwa $180nF$ bestimmt werden. Um noch kleinere Kondensatoren messen zu können, wurde der Strompuls bei Version 1.11k weiter auf $8\mu s$ verkleinert. Die Abbildungen 5.42 zeigen den Spannungsverlauf an einem $2,2\mu F$ Kondensator ohne und mit einem $2,7\Omega$ Serienwiderstand.

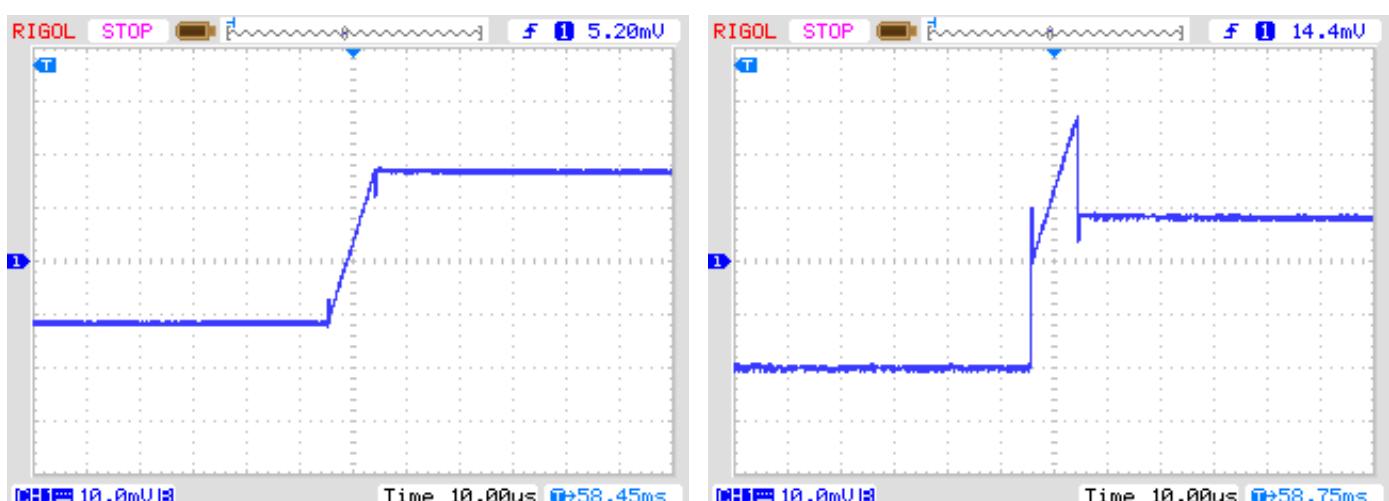


(a) ohne Serienwiderstand

(b) mit $2,7\Omega$ Serienwiderstand

Abbildung 5.42. Spannungsverlauf eines $2,2\mu F$ Kondensators während der ESR-Messung mit $8\mu s$ Ladepulsen

Weil in den Abbildungen 5.42 der Zeitpunkt der Spannungsübernahme des ADC nicht zu erkennen ist, wird in den Abbildungen 5.43 der Ladepuls gedehnt dargestellt. Der Zeitpunkt der Spannungsübernahme des ADC ist hier etwa in der Bildmitte.



(a) ohne Serienwiderstand

(b) mit $2,7\Omega$ Serienwiderstand

Abbildung 5.43. Ausschnitt des Spannungsverlaufs eines $2,2\mu F$ Kondensators während der ESR-Messung mit $8\mu s$ Ladepulsen

Die Messergebnisse der neuen ESR-Messmethode sind in Abbildung 5.44 dargestellt. Die ESR-Werte unterscheiden sich wegen der Frequenzabhängigkeit von den Ergebnissen der Abbildung 5.38. Die Vergleichswerte des LCR-Messgerätes sind bei $10kHz$ ermittelt worden.

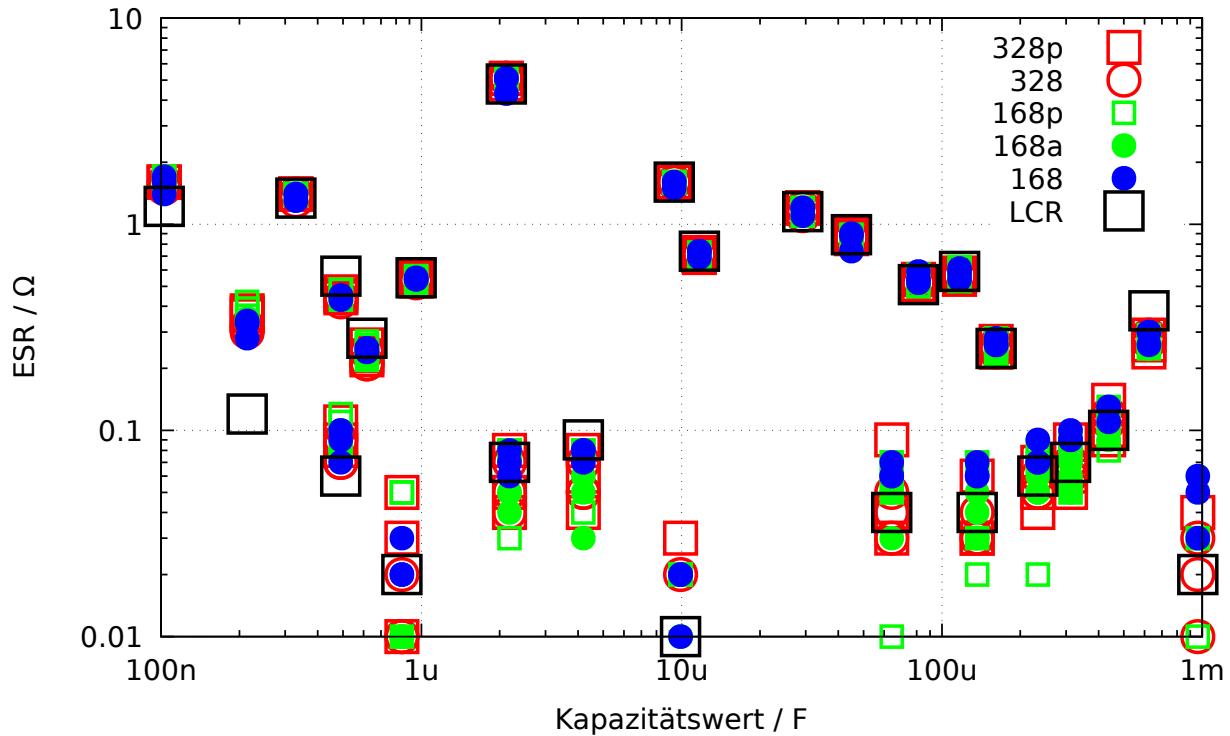


Abbildung 5.44. Messergebnisse der ESR-Messungen mit 15 verschiedenen ATmega, Methode 2

Eine Messreihe mit Elektrolyt-Kondensatoren verschiedener Größen zeigt die Abbildung 5.45. Es werden die Ergebnisse von einem PeakTech 3315 LCR-Messgerät bei verschiedenen Messfrequenzen und die Ergebnisse des TransistorTesters gegenübergestellt. Die Widerstandswerte sind in diesem Diagramm logarithmisch dargestellt. In allen Fällen liegen die Ergebnisse des TransistorTesters nahe an den Ergebnissen des LCR-Messgerätes bei 10kHz Messfrequenz. Nur der 500 μ F/3V Kondensator war ein älteres Exemplar, alle anderen waren neuwertig.

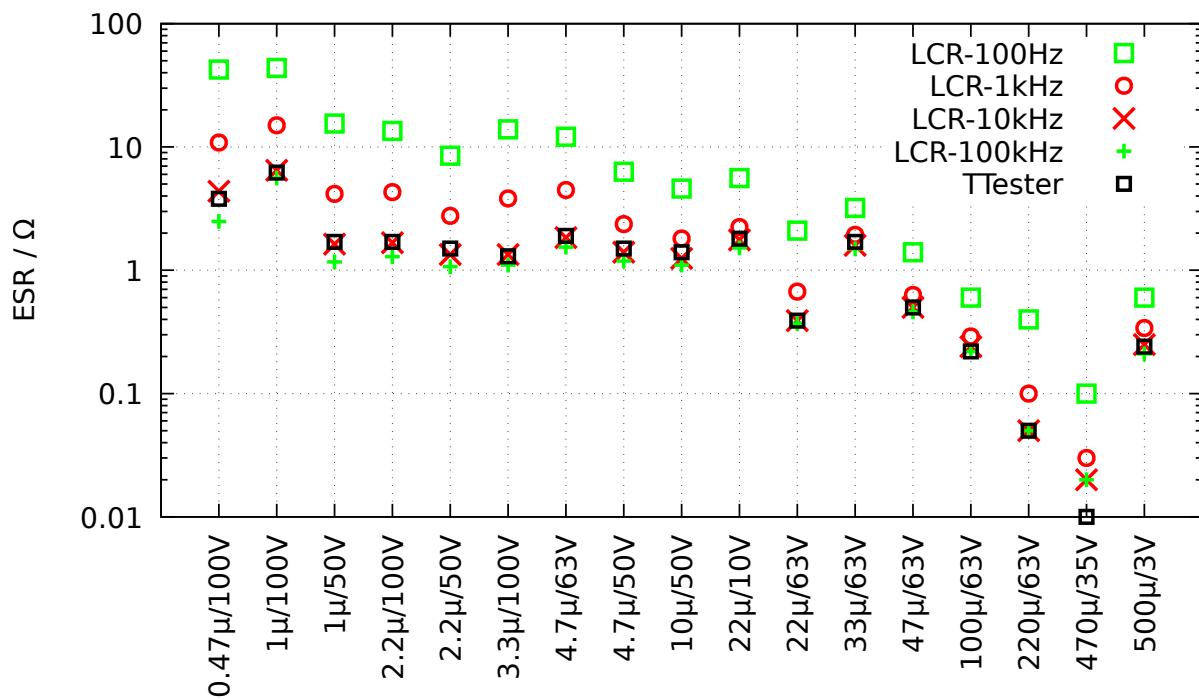


Abbildung 5.45. Ergebnisse der ESR-Messungen von verschiedenen Elektrolyt-Kondensatoren

Da mit der neuen Methode auch Widerstände gemessen werden können, werden in Abbildung 5.46 die Messfehler der Messung von einigen Widerständen unter 10Ω mit jeweils drei Exemplaren verschiedener ATmega gezeigt.

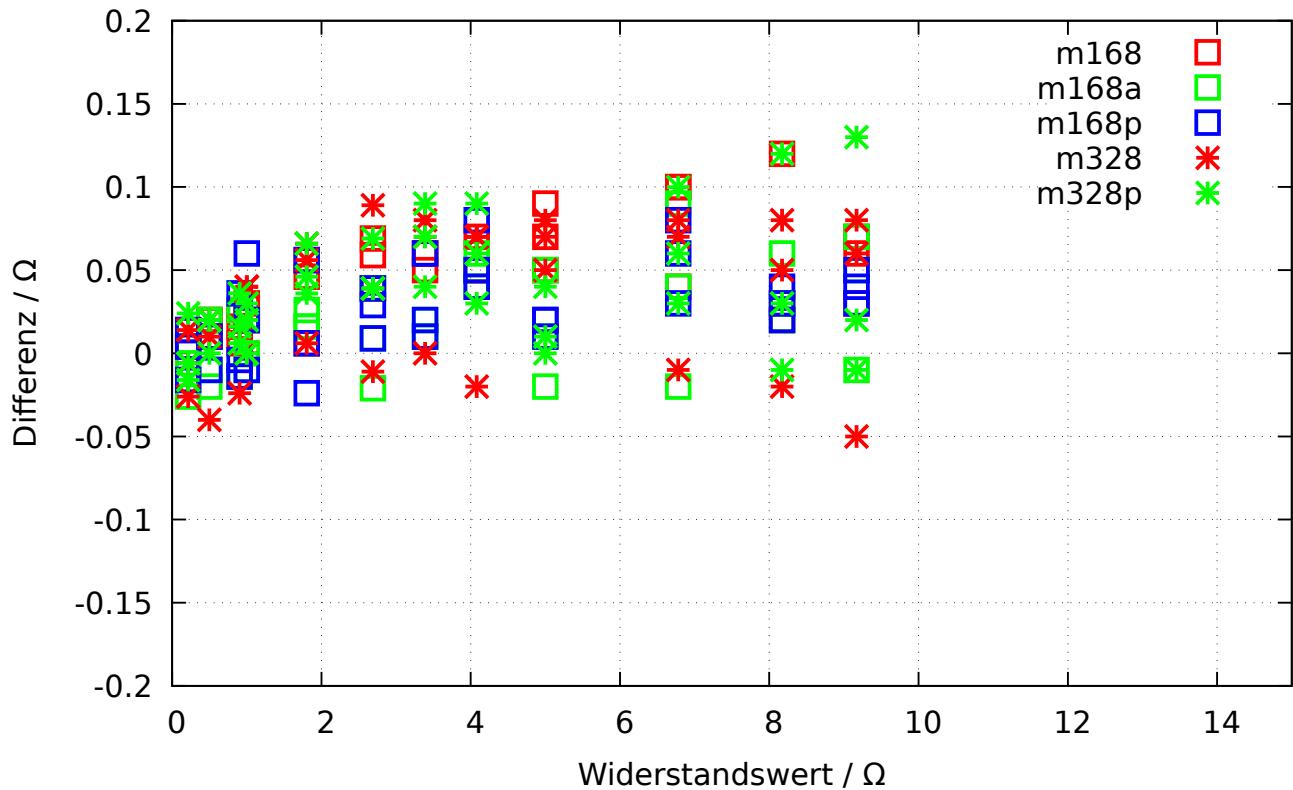
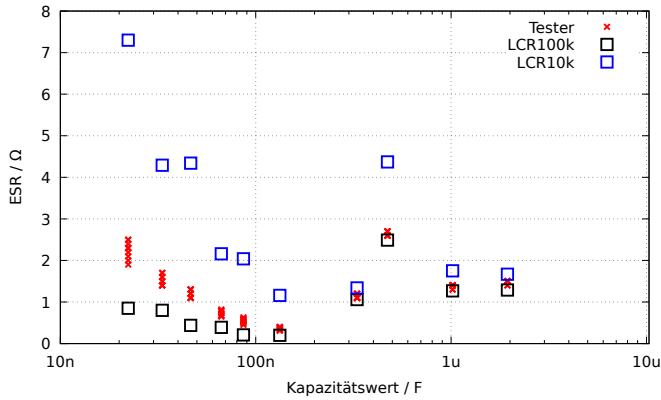
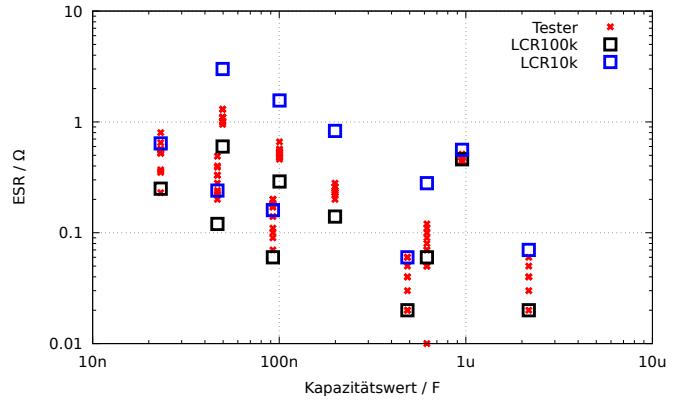


Abbildung 5.46. Messfehler von Widerständen mit der ESR Messung

Ab der Softwareversion 1.12k wurde die Ladepulslänge für die Kondensatoren auf $2\mu s$ verkürzt, um den ESR Wert bei noch geringeren Kapazitätswerten messen zu können. Es kann jetzt der ESR-Wert ab einer Kapazität von $20nF$ gemessen werden. Dabei wächst der Meßfehler mit geringer werdender Kapazität an. Der Grund hierfür ist die kleiner werdende Zeitkonstante der RC-Schaltung, die bei $20nF$ nur etwa $14.4\mu s$ beträgt. Dadurch ändert sich die Spannung auch während des $2\mu s$ Strompulses deutlich. Die Software kann den Abtastzeitpunkt nur auf einen Prozessortakt genau wählen. Aber das Eingangsfilter des ADC hat eine Zeitkonstante von etwa $0.24\mu s$, die wohl leider von Exemplar zu Exemplar etwas variiert. Diese Änderung der Filterzeitkonstante kann im Programm nicht berücksichtigt werden. Hierfür müsste der Abtastzeitpunkt des ADC auf einen Bruchteil der Prozessortaktzeit genau eingestellt werden können. Mit größer werdenden Kapazitätswerten des Meßobjektes wächst die Zeitkonstante an und die Spannungsänderung während des Ladepulses wird immer geringer. Damit hat auch die Variation der Zeitkonstante des ADC-Eingangsfilters immer weniger Einfluß auf die Meßergebnisse. Die Beispiele in Abbildung 5.47 zeigen die Ergebnisse von 10 verschiedenen Testern. Auf der linken Seite sind Kondensatoren mit höheren ESR-Werten vermessen worden. Da sieht das Ergebnis verglichen mit den Meßergebnissen des Peaktech 2170 LCR Meßgerätes bei $10kHz$ und $100kHz$ noch einigermaßen zufriedenstellend aus. Auf der rechten Seite sind die Meßwerte von qualitativ guten Kondensatoren mit niedrigen ESR-Werten dargestellt. Obwohl hier die Grenze des Verfahrens deutlich zu erkennen ist, ist das Ergebnis immer noch besser als gar keine Information. Es läßt sich in jedem Fall die Qualität eines Kondensators auch bei niedriger Kapazität einschätzen.



(a) höhere ESR Werte



(b) niedrigere ESR Werte

Abbildung 5.47. ESR-Messungen von Kondensatoren mit niedriger Kapazität

Für Prozessoren mit mehr als 16K Flash Speicher wird ab Softwareversion 1.12k für die Hälfte der Einzelmessungen zusätzlich zum 680Ω Widerstand der $470k\Omega$ Widerstand parallel geschaltet, um den Meßstrom geringfügig zu variieren. Leider ist der zusätzliche Strom sehr gering, so daß die damit erreichte Spannungserhöhung nur wenig zur Änderung des ADC-Ergebnisses beiträgt. Die Spannungserhöhung beträgt nur etwa 20% eines ADC bits mit der internen 1.1V Referenz. Wünschenswert wäre eine kleine zusätzliche Rauschspannung am ADC Eingang, die zur Änderung der ADC-Einzelwerte führen würde. Hiermit wäre dann eine statistische Erhöhung der ADC Auflösung durch Mittelwertbildung möglich.

5.3.8 Spannungsverlust nach einem Ladepuls, Vloss

Beim Messverfahren von großen Kapazitäten wird der Spannungsverlust nach einem Ladepuls im stromlosen Zustand untersucht. Die erreichte Ladespannung sinkt bei Elektrolyt-Kondensatoren nach kurzer Zeit wieder etwas ab. Dieser Spannungsverlust kann durch einen parallel geschalteten Widerstand verursacht werden. Ich nehme an, dass der Spannungsverlust bei Elektrolyt-Kondensatoren durch eine interne Ladungsverteilung nach dem Ladepuls verursacht wird. Bei einem langsamen Ladevorgang, wie er bei kleineren Kapazitäten mit dem $470k\Omega$ Widerstand gemacht wird, ist diese Umverteilung schon während des Ladevorgangs abgeschlossen. Dann tritt kein messbarer Spannungsverlust nach der Ladung in der betrachteten Zeitspanne auf. Wird der gleiche Elektrolyt-Kondensator aber mit einem kurzem pulsförmigen Strom geladen, ist hier auch ein Spannungsverlust durch die nachträgliche Umverteilung der Ladung zu beobachten. Der gleiche Effekt ist in geringerem Umfang auch bei keramischen Kondensatoren zu beobachten. Nach den bisherigen Beobachtungen sind Kondensatoren mit mehreren % Spannungsverlust verdächtig. Besonders auffällig in Bezug auf den Spannungsverlust sind auch ältere Papierkondensatoren, die auch für andere Messgeräte eine Herausforderung sind. Dies will ich in folgender Tabelle verdeutlichen.

Kondensator Type	Nenn-Kapazität	PeakTech LCR 2170	Voltcraft M2650-B	PeakTech 3315	Transistor-Tester
Papier	4700pF	6.75-10.36nF Q=2.5-32	8.00nF	25.40nF	10.71nF Vloss=11 %
Papier	6800pF	9.40-11.40nF Q=5-25	10.41nF	23.30nF	11.65nF Vloss=5.0 %
unbekannt	4700pF	5.85-6.33nF Q=16-87	6.12nF	6.90nF	6225pF Vloss=1.7 %
Folie	7870pF	7.86-7.87nF Q= > 1540	7.95nF	7.95nF	7872pF Vloss=0 %
Papier	22000pF	37.4-57.5nF Q=2.5-32	52.8nF	112nF	118.5nF Vloss=12 %
Folie	22600pF	22.4-22.5nF Q= > 1540	22.57nF	22.69nF	22.54nF Vloss=0 %
Papier	100nF	144-256nF Q=2.6-28	177nF	318nF	529.7nF Vloss=12 %
Keramisch	100nF	97.7-102nF Q=90-134	103.7nF	103.3nF	103.1nF Vloss=0.1 %
Folie	100nF	98.0-101nF Q=58-700	101.4nF	102.2nF	101.6nF Vloss=0 %

In der Tabelle kann man sehen, dass die Kapazität aller Folienkondensatoren mit befriedigender Genauigkeit von allen Messgeräten bestimmt werden. Die Kapazitäten und die Güten Q des PeakTech LCR-Messgerätes sind Minimum und Maximum der Messergebnisse im Frequenzbereich $100Hz$ bis $100kHz$. Bei allen Beispielen in der Tabelle ist der Spannungsverlust Vloss des TransistorTesters groß, wenn die Kondensatoren eine geringe Güte haben. Nur in diesem Fall sind auch große Abweichungen der gemessenen Kapazität zu beobachten. Damit der Spannungsverlust eines Kondensators vom Transistortester gemessen werden kann, muss die gemessene Kapazität größer als $5000pF$ sein.

5.3.9 Separate Kapazitäts- und ESR-Messung

Die separate Kapazitätsmessung mit anschließender ESR-Messung ist nur für ATmega mit genügend Speicher über den Bediendialog wählbar. Diese Art der Messung ist dafür gedacht, Kondensatoren im eingelöteten Zustand messen zu können. Bitte sorgen Sie dafür, dass alle Kondensatoren auf der Leiterplatte entladen sind, bevor Sie mit den Messungen beginnen. Um die Messung im eingebauten Zustand zu ermöglichen, wird dafür gesorgt, dass die Messspannung möglichst nur knapp über $300mV$ beträgt. Außerdem wird für die Messung nur mit dem 680Ω Widerstand durchgeführt um den Einfluss von Restströmen der angeschlossenen Bauteilen auf der Leiterplatte gering zu halten. Um möglichst kleine Kapazitätswerte messen zu können, wird mit einem nur $200\mu s$ kurzen Ladepuls begonnen. Wenn die Ladespannung erwarten lässt, dass auch bei einem $2ms$ langen Ladepuls die Ladespannung von $300mV$ nicht überschritten wird, wird danach $2ms$ lang weitergeladen. Wenn bei sehr großen Kapazitätswerten auch dann noch kein deutlicher Spannungszuwachs festzustellen ist, wird sogar mit $20ms$ langen Ladepulsen weitergeladen. Bei Annäherung der Spannung an die $300mV$ wird wieder mit den kürzeren Ladepulsen weitergeladen. Die gesamte Ladezeit wird dabei zusammengezählt und nach Überschreitung der $300mV$ die Kapazität aus der geladenen Spannung und der Ladezeit bestimmt. Mit diesem Verfahren sind Kapazitätswerte von etwas unter $2\mu F$ messbar. Die Obergrenze der Kapazität bei dieser Messung beträgt wegen einer Ladezeitbegrenzung auf $2,5s$ etwa $50mF$. Wenn ein Kapazitätswert gemessen wurde, wird danach der ESR-Wert des Kondensators mit dem schon im Kapitel 5.3.7 beschriebenen Verfahren gemessen. Die Ergebnisse werden nur kurz angezeigt und danach sofort mit der nächsten Messung begonnen. Die Messserie wird nach 250 Messungen oder auf

Tastendruck abgebrochen und dann zum Bedienmenü zurückgekehrt.

5.3.10 Ergebnisse der Kondensator-Messung

Die Ergebnisse meiner Messungen werden in Abbildung 5.48 für drei ATmega8 dargestellt. Zusätzlich sind einige Werte der Original-Software mit einem Korrekturfaktor von 0,88 (-12%) dargestellt. Weitere Ergebnisse von ATmega8-Versionen zeigen die Abbildungen 5.49a und 5.49b. Die Ergebnisse der Messung der gleichen Kondensatoren mit einem ATmega168 werden in Abbildung 5.50 gezeigt. Die Referenz für die Fehlerberechnung sind die Messwerte eines PeakTech 2170 LCR-Messgerätes, nicht die aufgedruckten Werte der Bauteile. Die grösseren relativen Messfehler bei großen Kondensatoren liegen zum Teil an der zu hohen Messfrequenz (100Hz) des LCR-Messgerätes für große Elektrolytkondensatoren, zum anderen spielt auch die schlechte Güte der Elektrolytkondensatoren eine Rolle.

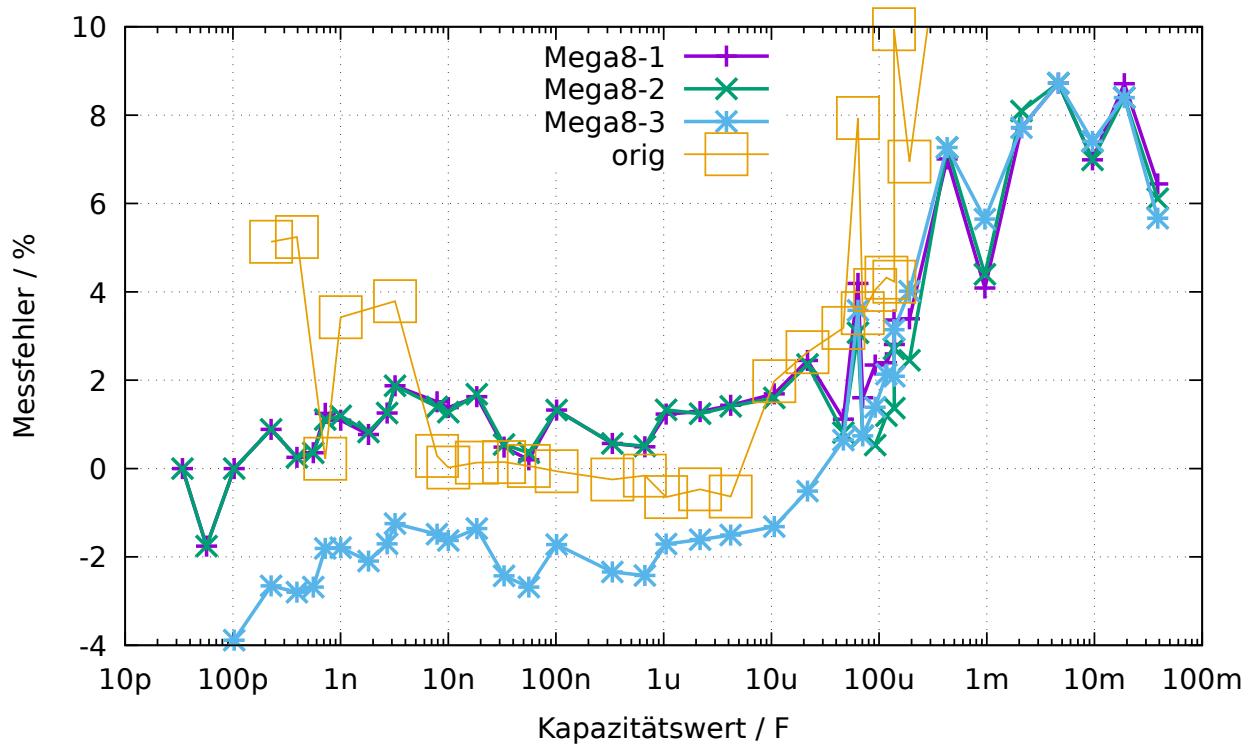


Abbildung 5.48. Prozentualer Fehler der Kondensator-Messungen mit drei ATmega8

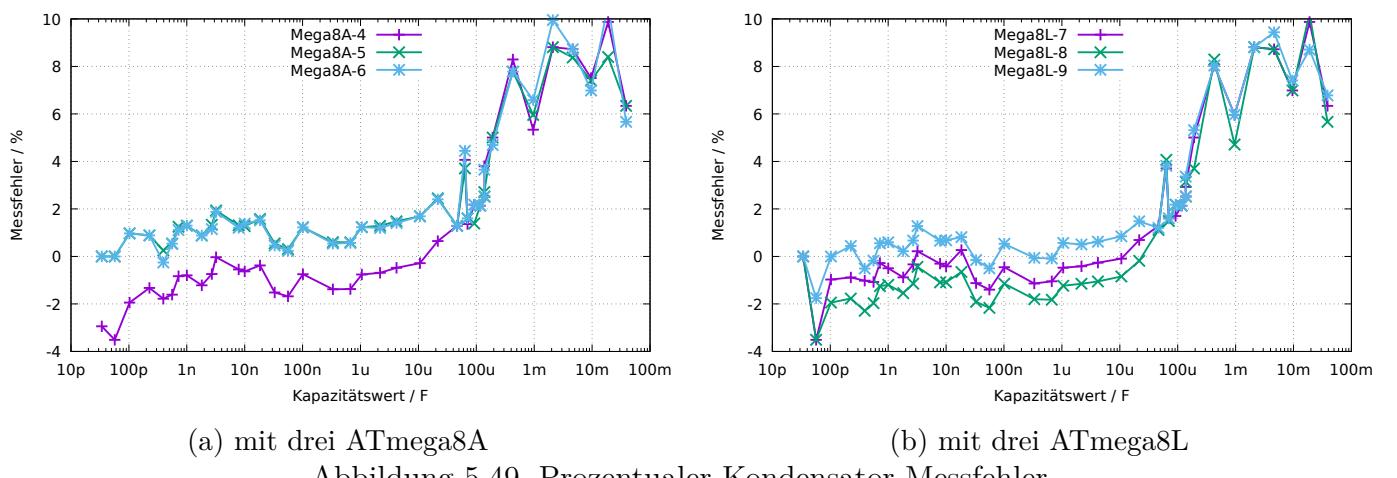


Abbildung 5.49. Prozentualer Kondensator-Messfehler

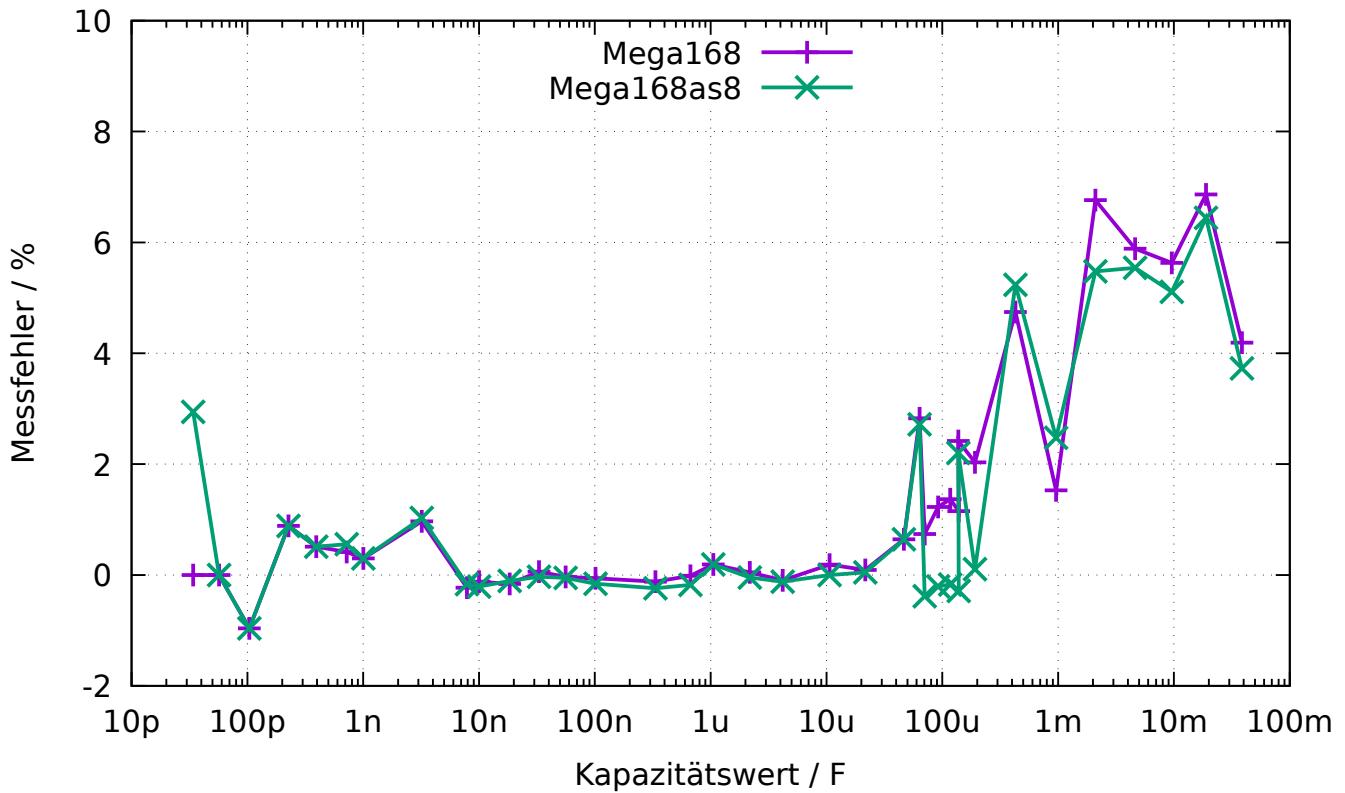


Abbildung 5.50. Prozentualer Fehler der Kondensator-Messungen mit dem ATmega168

Wie schwierig es ist, für Kondensatormessungen den richtigen Bezugswert zu finden, soll die Abbildung 5.51 zeigen. Als Bezugswert wurde hier eine beste Schätzung genommen. Die Kurve „Multimeter“ zeigt die Abweichungen, die mit einem Peaktech 3315 Multimeter gemessen wurden. Die nächste Kurve „LCR“ zeigt die Abweichungen, die mit einem Peaktech 2170 LCR-Meter in dem jeweils günstigsten Frequenzbereich gemessen wurden. Zum Vergleich werden mit der Kurve „ATmega168as“ auch noch die Messabweichungen eines ATmega168 bestücktem Transistor-Testers gezeigt. Ob die gezeigten Fehler aber tatsächliche Messfehler des jeweiligen Gerätes sind, muss bezweifelt werden, da auch die Schätzung des Kapazitätswert nicht der wirklichen Kapazität entspricht.

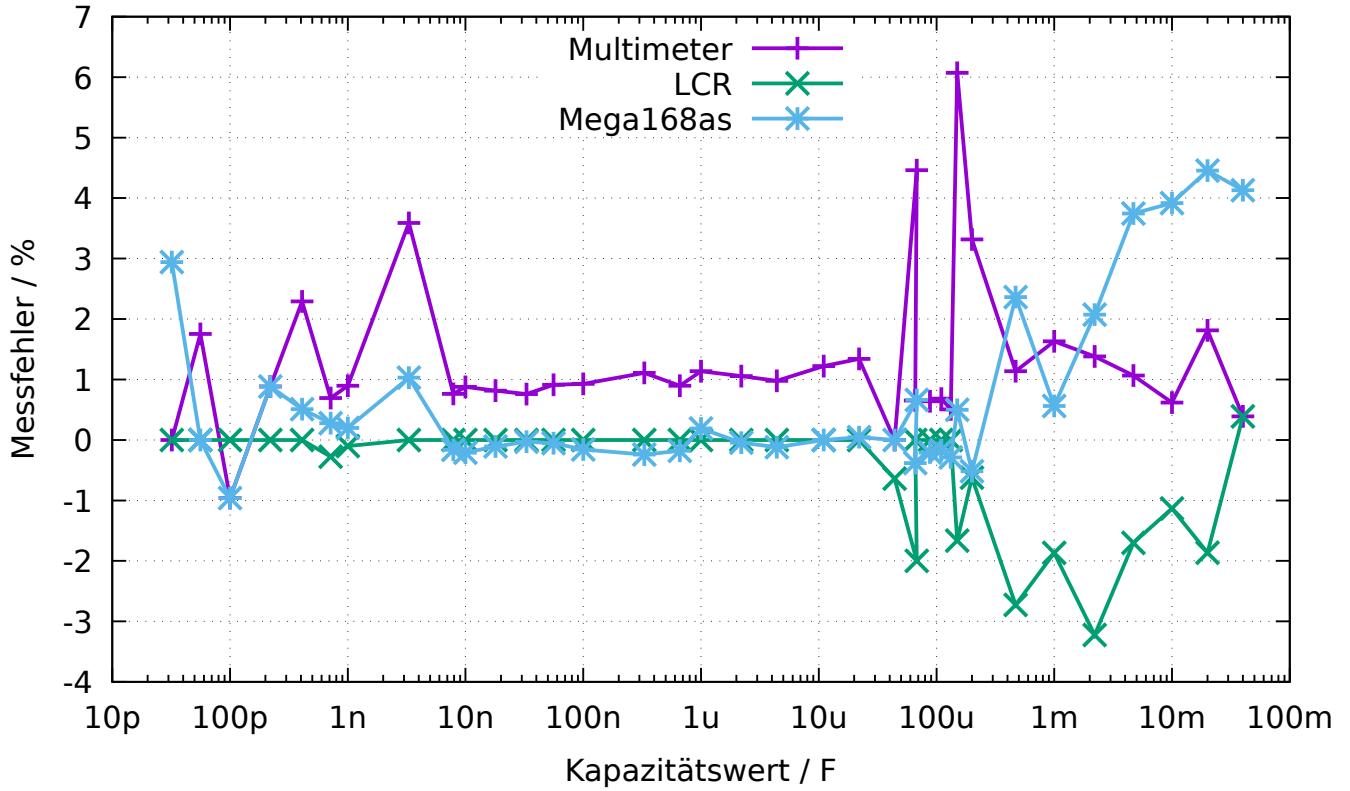
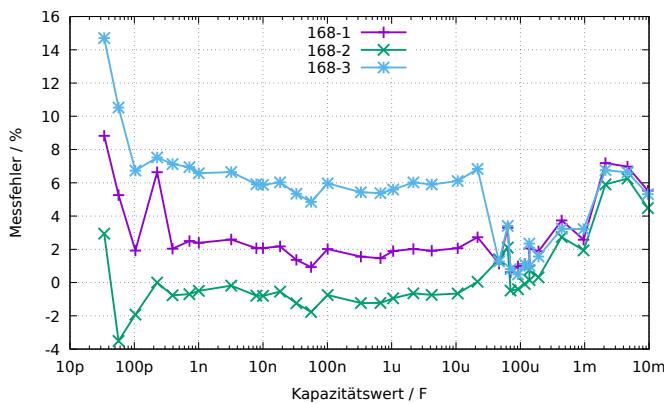
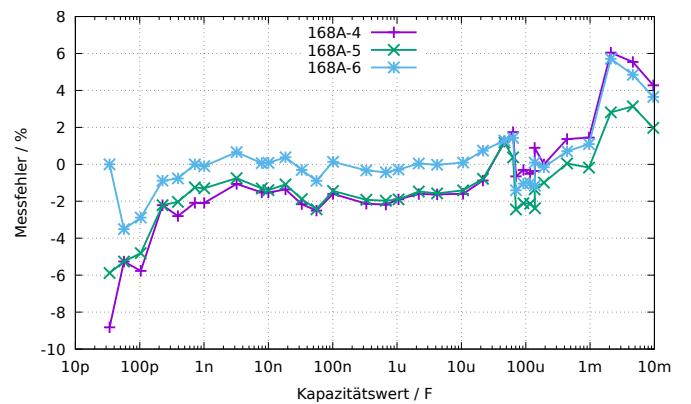


Abbildung 5.51. Vergleich der Kondensator-Messungen mit Multimeter, LCR-Meter und ATmega168

Die Abweichungen der Messergebnisse von drei verschiedenen ATmega168 werden in Abbildung 5.52a dargestellt. Hier wurde die Messung des LCR-Meters als Vergleichsbasis genommen. Entsprechend werden die Messergebnisse von drei verschiedenen ATmega168A in Abbildung 5.52b, von drei verschiedenen ATmega168PA in Abbildung 5.53 und von drei verschiedenen ATmega328 in Abbildung 5.54a sowie von drei ATmega328P in Abbildung 5.54b gezeigt. Hierbei wurde nur der Nullwert der Kapazitätsmessung von $39pF$ berücksichtigt, alle anderen Korrekturmöglichkeiten wurden nicht benutzt. Dieser Nullwert beinhaltet schon die $2 - 3pF$, die durch die etwa $12cm$ langen Anschlussleitungen mit den Klemmen verursacht werden. Auch das Board-Layout hat Einfluss auf diesen Nullwert. Diesen Nullwert habe ich mit der Boardversion „DG2BRS V 5.2.1“ ermittelt.



(a) drei ATmega168



(b) drei ATmega168A

Abbildung 5.52. Kondensator-Messfehler, unkalibriert

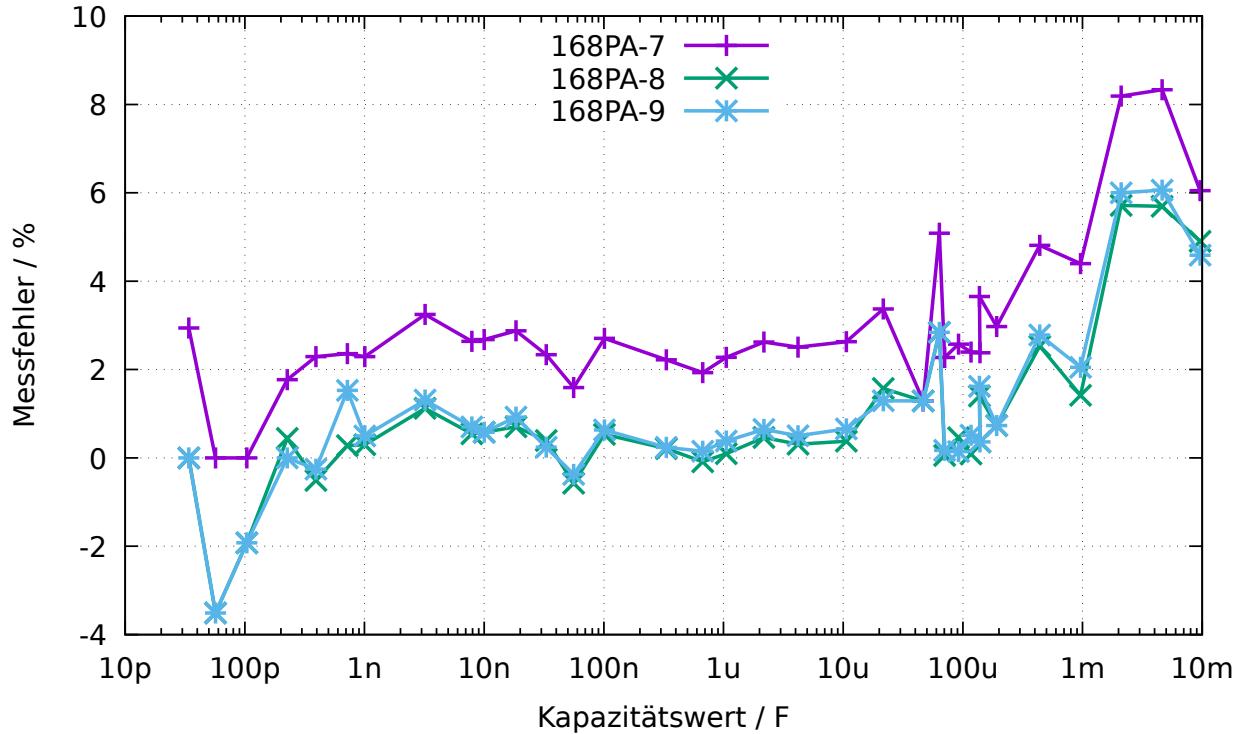
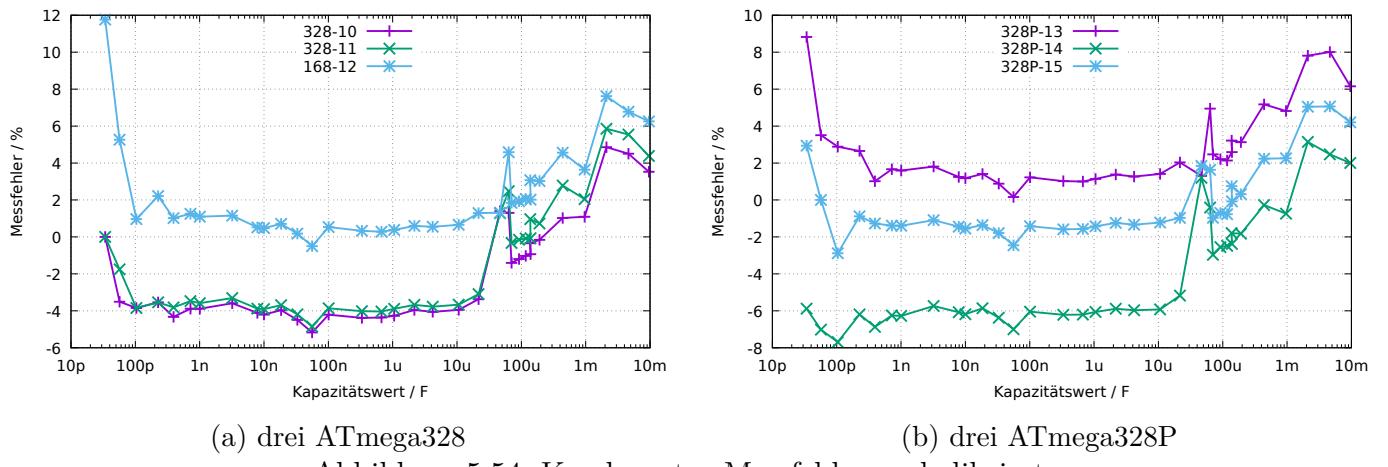


Abbildung 5.53. Kondensator-Messfehler von drei ATmega168PA, unkalibriert



(a) drei ATmega328 (b) drei ATmega328P

Abbildung 5.54. Kondensator-Messfehler, unkalibriert

Zum Erreichen der besten Messgenauigkeit muss die Software an die individuellen Eigenschaften des verwendeten ATmega angepasst werden. Dazu kann die Korrekturspannung REF_C_KORR für den Komparator angegeben werden, der für die Messung der kleinen Kapazitäten eingesetzt wird. Eine Korrekturspannung von 1mV führt zu einer Verminderung der Kapazitätsanzeige von etwa 1,1 Promille. Bei einem automatischen Abgleich ist REF_C_KORR nur ein Offset zu der gemessenen Differenzspannung von geladenem Kondensator und der internen Referenz. Für die großen Kapazitäten kann der Promillewert C_H_KORR angegeben werden, um den die Messungen zu groß sind. Da die großen Kondensatoren meistens Elektrolytkondensatoren mit schlechter Güte sind, ist die Bestimmung des wahren Kapazitätswertes und damit auch des Messfehlers hier besonders schwierig. Besonders bei den ATmega168 habe ich bei kleinen Kapazitätswerten einen Messfehler beobachtet, der von der Anstiegs geschwindigkeit der Spannung beim Ladevorgang abhängig ist. Die Abbildung 5.55 zeigt die Messabweichung bei Kondensatormessung nur mit der Berücksichtigung des Nullwertes (168-3-A), mit dem Korrekturfaktor für kleine Kondensatoren REF_C_KORR=66 sowie dem Korrekturfaktor für große Kondensatoren C_H_KORR=5 (168-3-B), sowie zusätzlich als Kurve 168-3-C

mit der Berücksichtigung einer Spannungsanstiegskomponente ($\text{COMP_SLEW1}=4000$ und $\text{COMP_SLEW2}=220$) und des Selbstentladeverhaltens der großen Kondensatoren. Der Spannungsanstiegs-Korrekturfaktor berechnet sich nach $\frac{\text{COMP_SLEW1}}{\text{cval}+\text{COMP_SLEW2}} - \frac{\text{COMP_SLEW1}}{\text{COMP_SLEW2}}$, wobei cval der gemessene Kapazitätswert in pF ist.

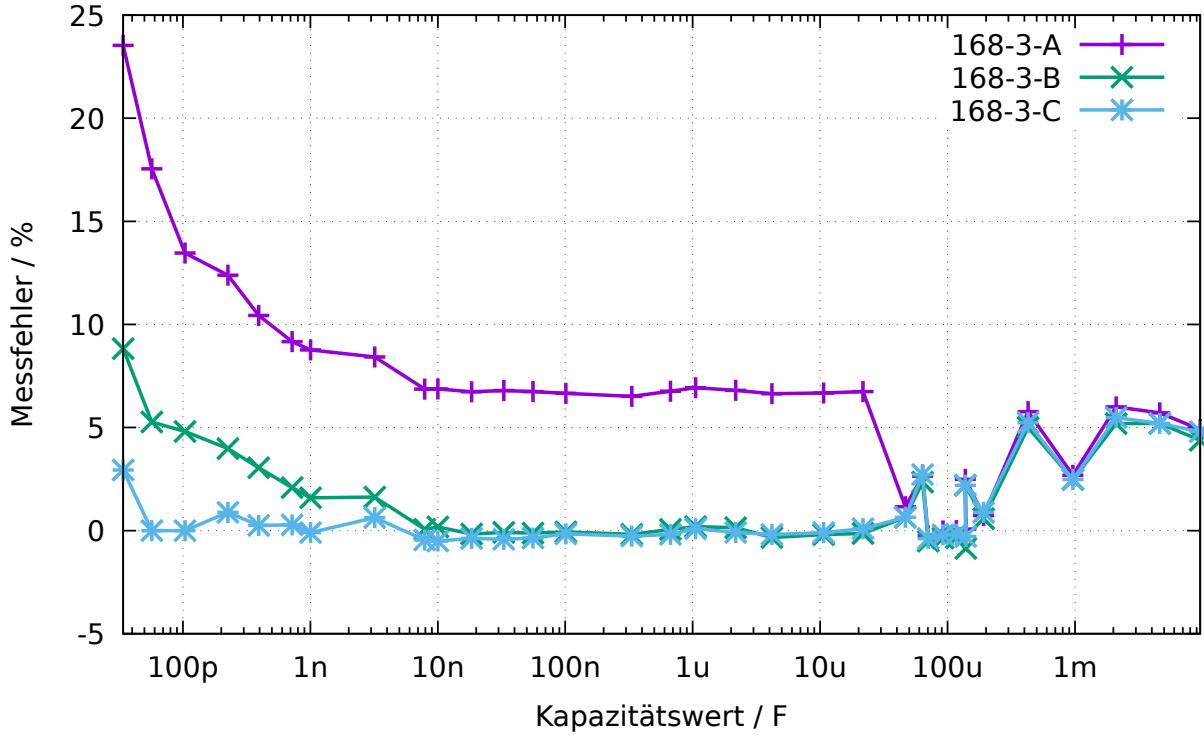


Abbildung 5.55. Optimierung der Kondensator-Messung eines ATmega168

5.3.11 Automatischer Abgleich der Kondensator-Messung

Der automatische Abgleich besteht aus zwei Teilen. Der erste Teil besteht darin, einen Nullabgleich der Kondensatormessung durchzuführen. Dazu wird in der Selbsttest-Routine der Mittelwert der gemessenen Kapazität bei nicht angeschlossenem Kondensator bestimmt. Es werden die Mittelwerte aus jeweils 8 Messungen für alle 6 Messkombinationen bestimmt. Nach erfolgreichem Abgleich werden die Korrekturwerte im EEPROM festgehalten und für künftige Messungen verwendet. Schwieriger ist die Beseitigung der Exemplarstreuungen bei der Messung von Kondensatoren bis etwa $40\mu\text{F}$, wie sie in den Abbildungen 5.52a, 5.52b und 5.53 gezeigt wurden. Als wesentliche Ursache wurde das unterschiedliche Verhalten (Offset-Spannung) des analogen Komparators herausgefunden.

In Abbildung 5.56 werden die Daten von den neun untersuchten Prozessoren gezeigt. Die Punkte „diff2ref“ zeigen die Spannungsdifferenz, die sich nach dem Laden eines Kondensators mit 660nF zu der jeweiligen internen Referenzspannung ergibt. Idealerweise wäre diese Spannung immer Null, wenn der analoge Komparator rechtzeitig das Signal zum Beenden des Ladevorganges gegeben hätte. Die kurze Verwaltungszeit des ATmega sollte bei der relativ großen Kapazität zu keiner messbaren Spannungserhöhung geführt haben. Die Punkte „CapErr“ zeigen die aus den Abbildungen 5.52a, 5.52b und 5.53 geschätzten Messfehler der einzelnen ATmega-Exemplare in Promille. Auffällig ist, wie die „CapErr“ Punkte den „diff2ref“ Punkten folgen. Deshalb zeigen die Punkte „diff“ die Differenz zwischen den jeweiligen „CapErr“ und „diff2ref“ Punkten. Für die Korrektur der Kondensatormessung kann also die Differenz der Kondensator-Ladespannung zur internen Referenzspannung („diff2ref“) benutzt werden. Dabei kann zusätzlich der mittlere Wert der Differenzen zu den „CapErr“ Schätzungen („diff“) berücksichtigt werden.

Im zweiten Teil der Abgleichsprozedur muss also ein Kondensator hoher Güte zwischen Pin 1 und Pin 3 mit einer Kapazität zwischen 100nF und $20\mu\text{F}$ angeschlossen werden. Es sollte ein Folienkondensator,

nach Möglichkeit kein keramischer Kondensator und schon gar kein Elektrolyt-Kondensator sein. Der Kapazitätswert braucht aber für diesen Abgleich nicht genau bekannt zu sein.

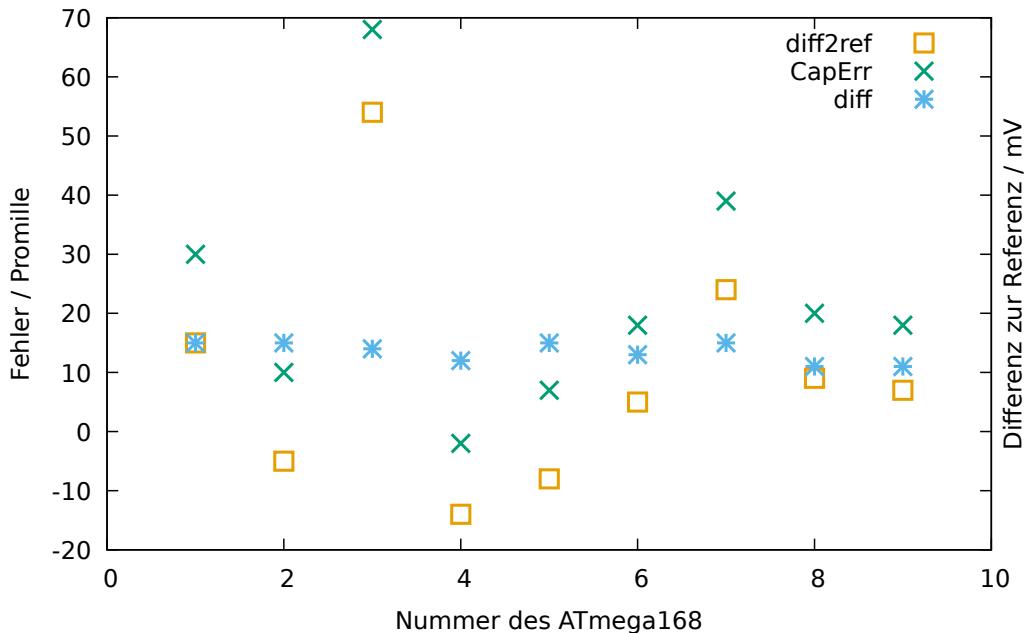


Abbildung 5.56. Daten von neun ATmega168-Prozessoren

Die Diagramme 5.57a, 5.57b, 5.58, 5.59a und 5.59b zeigen die Messergebnisse der gleichen Prozessoren in einer Standard-Softwarekonfiguration nach der Autokalibration. Die Prozessoren wurden alle mit der selben Software programmiert, lediglich die Makefile Option „PARTNO =“ musste wegen dem avrdude-Programm an den unterschiedlichen Prozessortyp („m168“, „m168p“, „m328“ oder „m328p“) angepasst werden. Nach der Programmierung wurde bei jedem ATmega ein Selbsttest gestartet und bei Test 10 ein Kondensator mit $330nF$ an Pin 1 und Pin 3 angeschlossen.

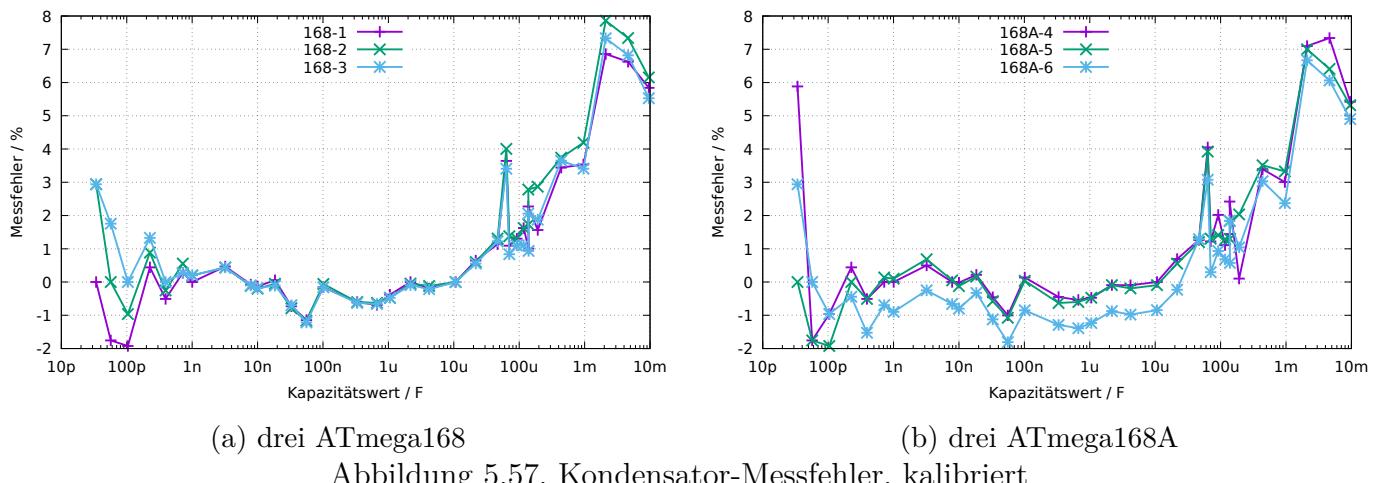


Abbildung 5.57. Kondensator-Messfehler, kalibriert

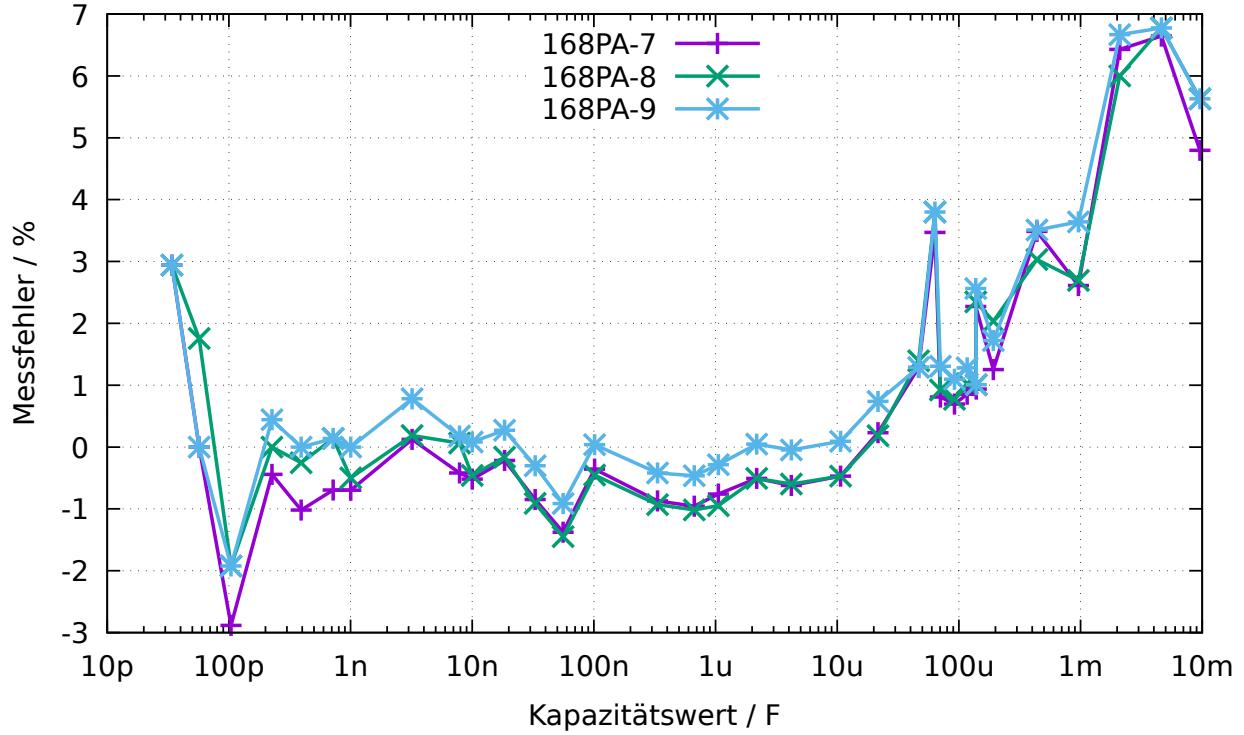
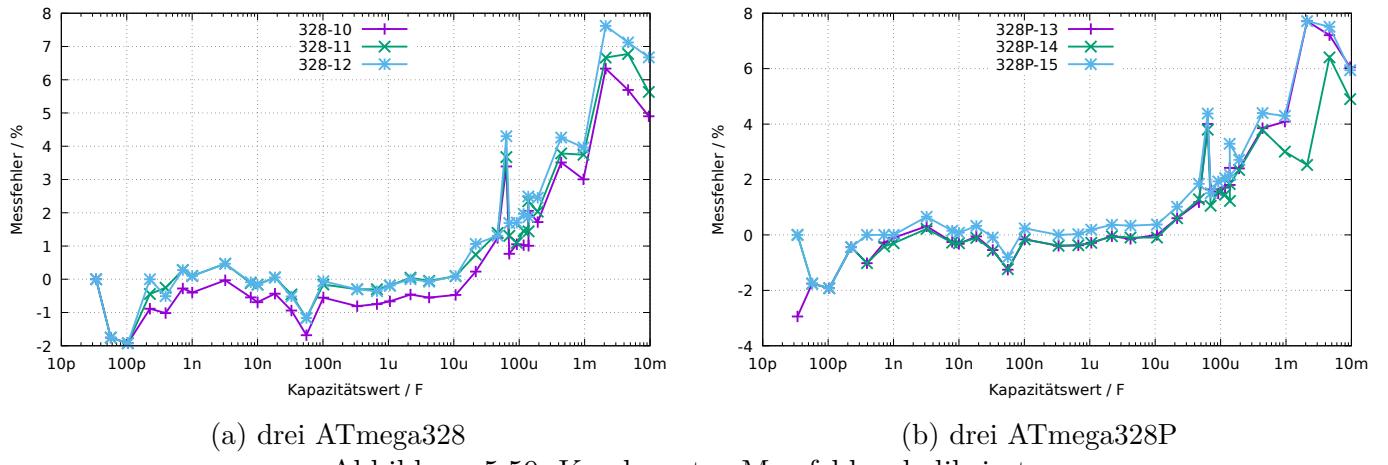


Abbildung 5.58. Kondensator-Messfehler von drei ATmega168PA, kalibriert



(a) drei ATmega328 (b) drei ATmega328P

Abbildung 5.59. Kondensator-Messfehler, kalibriert

Zuletzt will ich die Wirkung der AUTO_CAL-Option im Selbsttest noch einmal verdeutlichen. Die folgende Abbildung 5.60 zeigt die Ergebnisse von drei ATmega-Prozessoren mit der grössten Messabweichung noch einmal vor und nach der Kalibration. Die Punkte mit der Endung „unc“ zeigen die Messabweichungen ohne Kalibration. Die Linien mit der Endung „cal“ zeigen die Messabweichungen der gleichen Prozessoren mit der gleichen Software nach der Kalibration im Selbsttest-Zweig. Die Ursache der Messabweichungen für große Kondensatoren ($>40\mu F$) ist noch nicht bekannt. Alle verwendeten Kondensatoren für diese Messreihe waren Folienkondensatoren oder keramische Kondensatoren ($56pF$, $100pF$ und $3,3nF$), keine Elkos.

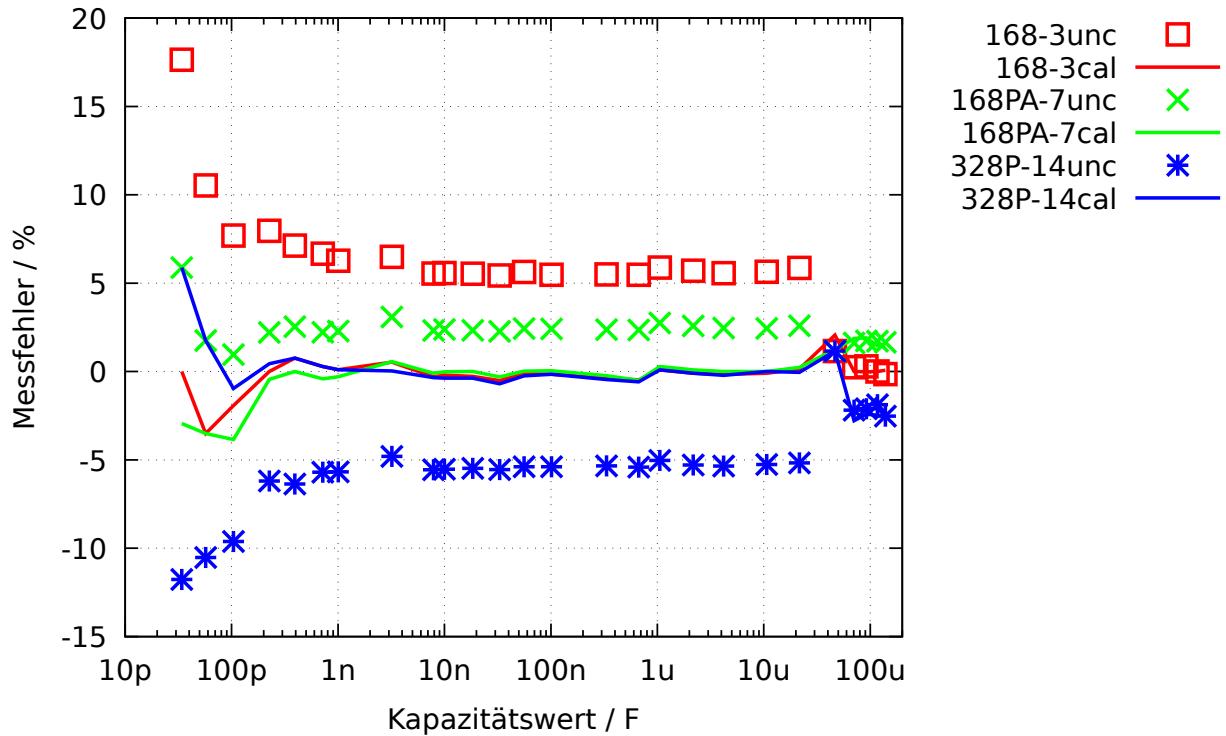


Abbildung 5.60. Kondensator-Messfehler von drei ATmega, vor und nach der Kalibration

Die Schaltung mit dem ATmega644 oder ATmega1284 sieht einen Kondensator auf der Platine für den Abgleich vor. Das Diagramm 5.61 zeigt die Ergebnisse der Kondensatormessung mit einem ATmega1284, sowohl mit dem $100nF$ keramischen Kondensator auf der Platine (1284-int) abgeglichen als auch mit einem externen $220nF$ Folien-Kondensator abgeglichen, im Vergleich zu den Ergebnissen eines ATmega328 auf einer anderen Platine.

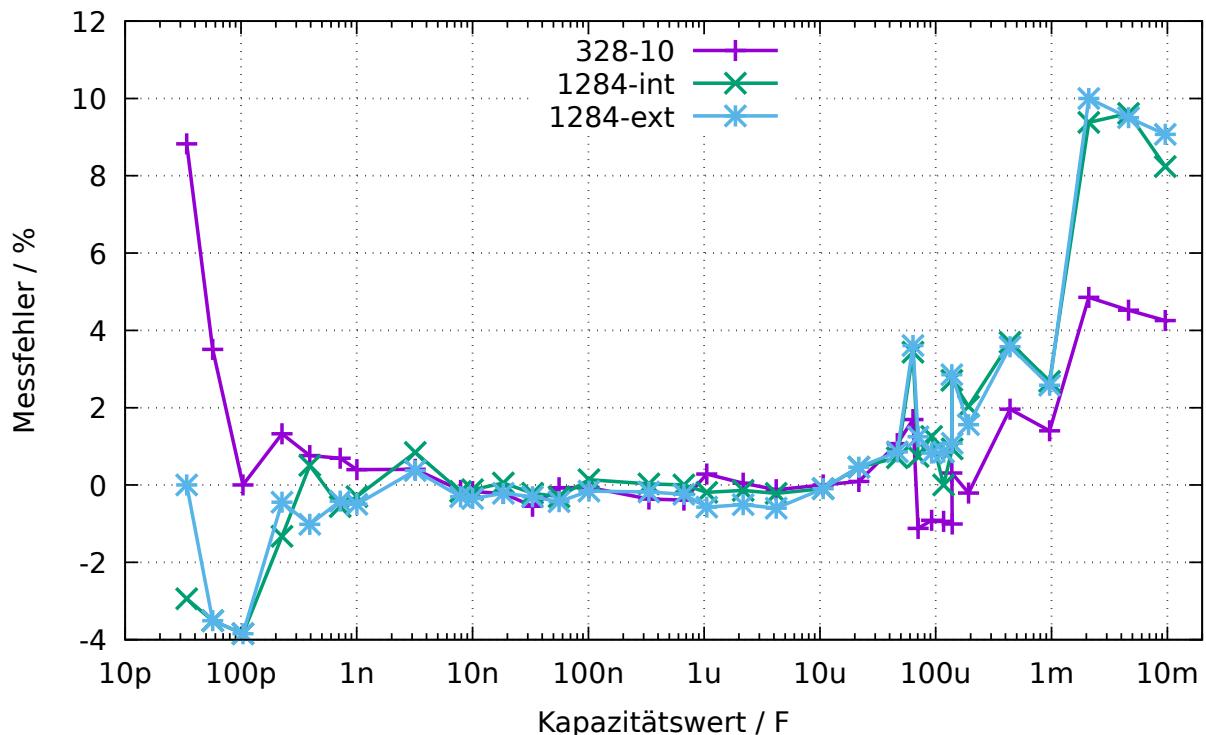


Abbildung 5.61. Kondensator-Messfehler von einem ATmega1284 im Vergleich zu einem ATmega328

5.4 Messen von Induktivitäten

Die Messung von Induktivitätswerten wird nach allen anderen Messungen als separater Teil mit allen gefundenen Widerständen mit weniger als 2100Ω durchgeführt. Das Messverfahren beruht auf dem Prinzip, dass beim Schliessen des Stromkreises der Strom nach der Formel $I_l = I_{max} \cdot (1 - \exp \frac{-t}{\tau})$ ansteigt. Die Zeitkonstante $\tau = \frac{L}{R}$ ist proportional zu der Induktivität L , aber umgekehrt proportional zum Widerstand R . Der Strom kann hier nur indirekt über den Spannungsabfall an einem Widerstand gemessen werden.

Leider wird durch den relativ hohen Widerstand 680Ω die Zeitkonstante zusätzlich verringert, was wiederum die Messung von kleinen Induktivitäten mit dem Takt von $8MHz$ zusätzlich erschwert. Um die Zeitkonstante zu bestimmen, wird die Spannung am 680Ω -Widerstand als Stromsensor mit dem analogen Komparator überwacht. Wenn der Spannungsabfall am 680Ω -Widerstand grösser als die Vergleichs-Spannung der internen Spannungsreferenz wird, meldet der Komparator dies an den beim Stromeinschalten gestarteten 16-Bit-Zähler weiter, der daraufhin den Zählerstand dieses Ereignisses festhält. Eventuelle Überläufe des Zählers werden vom Programm mitgezählt. Wenn die Spannung grösser ist, wird der Zähler sofort angehalten und aus dem festgehaltenen Zählerstand und dem Überlaufzähler die Gesamtzeit bestimmt. Der Anschluss der Spule wird wieder von VCC auf GND geschaltet, und über eine Spannungsüberwachung beider Anschlüsse gewartet, bis kein Strom mehr festgestellt wird. Das Schaltbild 5.62 zeigt ein vereinfachtes Diagramm der Messsituation.

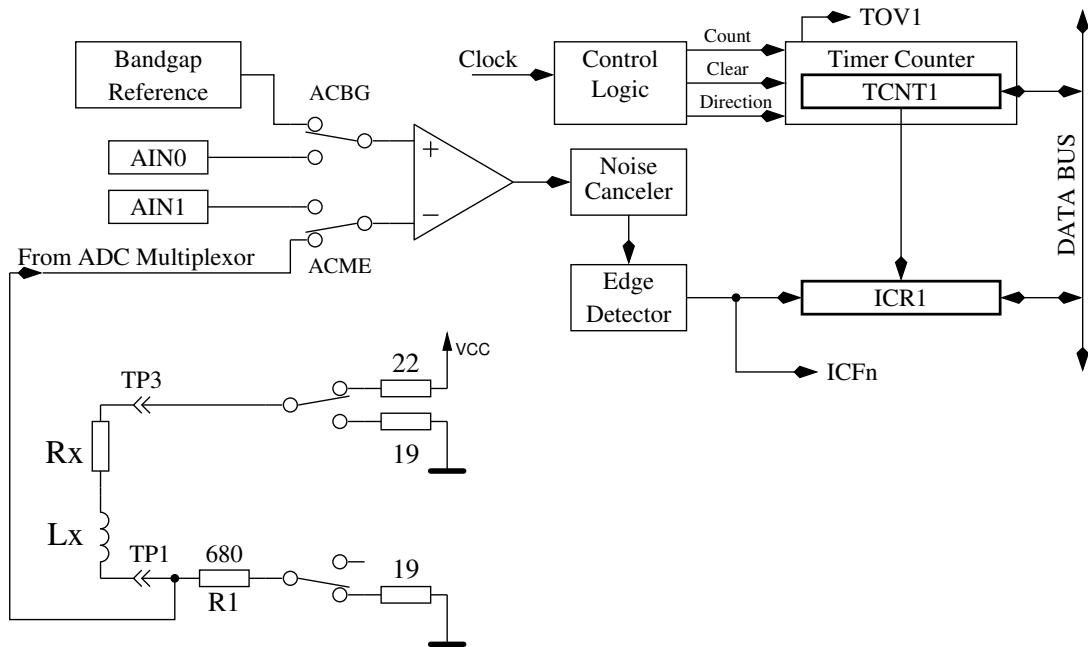


Abbildung 5.62. Messung von Induktivitäten mit dem Komparator

Aus der Versorgungsspannung VCC und der Summe aller Widerstände im Stromkreis kann der Maximalstrom I_{max} und daraus der Anteil der Vergleichsspannung im Verhältnis zur Maximalspannung am 680Ω -Widerstand $U_{max} = I_{max} \cdot (680 + 19)$ bestimmt werden. Mit der Formel $L = -\frac{t \cdot R_{ges}}{\log(1 - \frac{U_{ref}}{U_{max}})}$ kann die Induktivität bestimmt werden. Der natürliche Logarithmus wird im Programm mit einer Tabelle ermittelt. Die Auflösung der Induktivität wird für diese Art der Messung auf $0,1mH$ gesetzt.

Um auch kleinere Induktivitäten messen zu können, wird der 680Ω -Widerstand im Stromkreis weggelassen, wenn der Widerstandswert der Spule kleiner 24Ω gemessen wurde. Als Messwiderstand für die Strom-Messung dient in diesem Fall der Ausgangswiderstand der Ausgabeports (19Ω). In diesem Fall wird der Spitzenstrom grösser als es die Spezifikation des ATmega erlaubt. Da das nur für eine sehr kurze Zeit passiert, erwarte ich keine Schäden. Um eine längere Zeittdauer mit

überhöhtem Strom auszuschliessen, wird die zusätzliche Messung mit verzögertem Zählerstart immer mit 680Ω -Widerstand durchgeführt. Für diesen Typ der Messung wird die Auflösung der Induktivität auf $0,01mH$ gesetzt. Um die Messergebnisse an den tatsächlichen Induktivitätswert anzugeleichen, wird vom Zählerstand ein Nulloffset von 6 abgezogen, wenn ohne 680Ω gemessen wurde. Sonst wird ein Nulloffset von 7 oder 8 berücksichtigt.

Bei großen Induktivitäten können parasitäre Kapazitäten den Strom so schnell ansteigen lassen, dass die Spannungsüberwachung mit dem Komparator sofort anspricht. Um dennoch die Induktivität bestimmen zu können, wird die gleiche Messung noch einmal gemacht, aber der Zähler etwas später gestartet, damit der Spannungsanstieg durch den Stromzuwachs der Induktivität und nicht die Stromspitze durch die Steukapazität gemessen wird. Die Messungen werden in beiden Stromrichtungen durchgeführt. Von den beiden Messungen in gleicher Stromrichtung wird das höhere Messergebnis verwendet. Von den Messungen in verschiedenen Stromrichtungen wird der kleinere Wert als Resultat der Induktivitätsmessung genommen.

5.4.1 Ergebnisse der Induktivitäts-Messungen

Die Abbildung 5.63 zeigt die Messergebnisse verschiedener Induktivitäten. Die Induktivitäten über $1H$ sind Relais und Primärwicklungen von Netztrafos, die wegen der Remanenz des Eisenkerns schwierig zu messen sind.

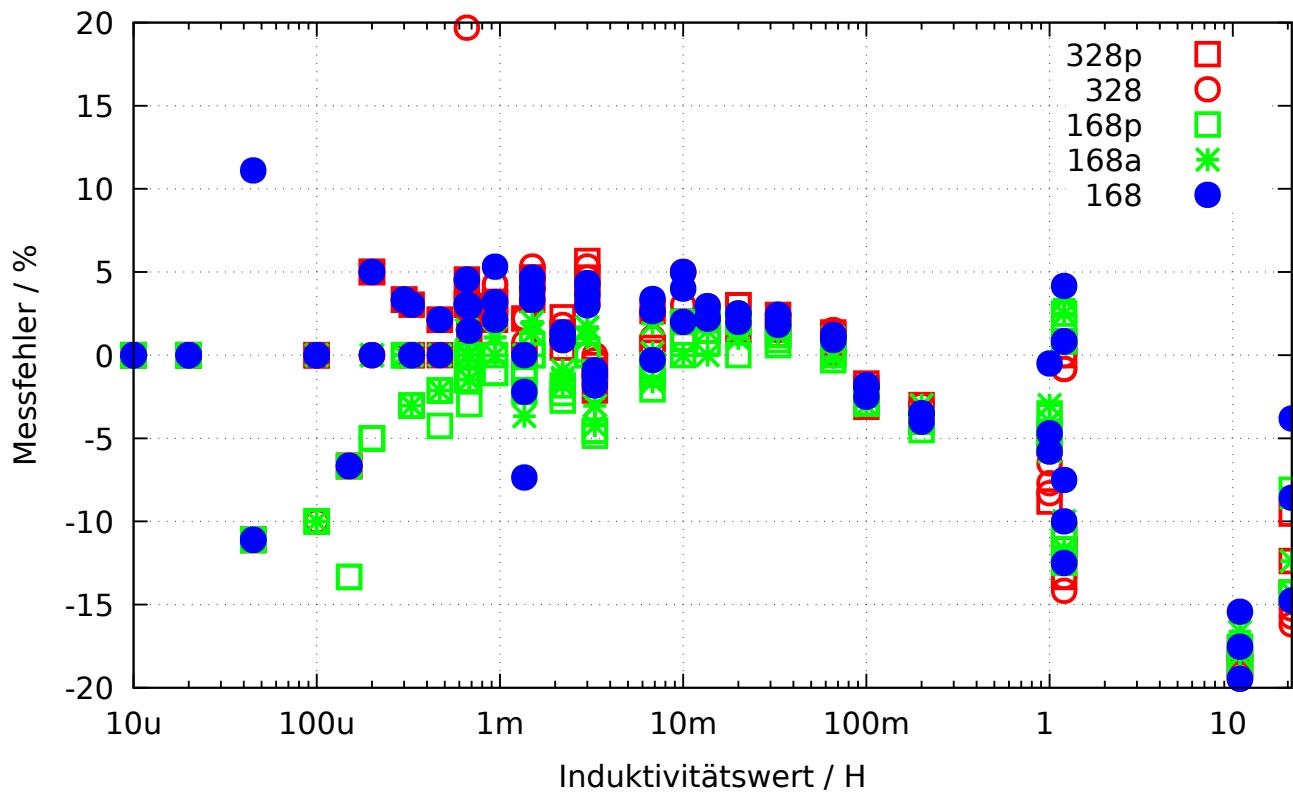


Abbildung 5.63. Induktivitäts-Messfehler von 15 verschiedenen ATmega

5.4.2 Messung kleiner Induktivitäten mit der Sampling Methode

Die kleinste erfassbare Induktivität bei der normalen Meßmethode beträgt $0.01 mH$. Für Hochfrequenzanwendungen ist aber die Messung kleinerer Induktivitätswerte sinnvoll. Die normale Messung verwendet für die Induktivitätsbestimmung den Stromanstieg in der Spule. Dieser Weg ist für die Sampling Methode nicht verwendbar, da für die Messung keine zusätzlichen Widerstände verwendet werden. Der Strom erreicht damit schnell unzulässig hohe Werte. Ein Schaden des ATmega wird bei

der normalen Messung nur dadurch verhindert, daß der Strom frühzeitig wieder abgeschaltet wird. Für die Sampling-Methode wäre das Abschalten schwierig durchzuführen und zusätzlich müßte der kritische Vorgang ja oft hintereinander wiederholt werden. Aus diesem Grunde hat der Funkamateur Pieter-Tjerk (PA3FWM) eine andere Methode für die Messung benutzt. Mit einem parallel geschaltetem Kondensator wird mit der Induktivität ein Schwingkreis gebildet. Mit einem kurzen Strompuls wird der Schwingkreis angeregt und mit der Sampling Methode versucht, die Eigenfrequenz des Schwingkreises zu bestimmen. Weil für die ADC-Messung ein Ende der Spule auf Massepotential gehalten werden muß, ergeben sich zwei Schwierigkeiten. Die Spannung der Schwingung geht natürlich auch in den negativen Bereich. Hier wird die Schwingung durch die interne Schutzdiode des ATmega auf etwa 0.6V begrenzt. Damit wird auch die positive Spitzenspannung der Schwingung auf diesen Wert begrenzt. Daneben kann der ADC nur positive Spannungen messen. Deswegen fehlt jeder negative Teil der Schwingungen. Anstelle der negativen Spannungen liest der ADC immer eine Null. Trotzdem ist es Pieter-Tjerk gelungen, die Eigenfrequenz hinreichend genau aus den gemessenen ADC-Daten zu bestimmen. Aus der Eigenfrequenz des Schwingkreises läßt sich die Induktivität berechnen, wenn der Kapazitätswert bekannt ist. Aus diesem Grunde ist die Kalibration um die Messung einer Parallelkapazität für die Induktivitätsmessung erweitert worden. Der Kondensator wird mit der Meldung inquotes1 —||— 3 10-30nF(L) angefordert. Beim unkalibrierten Tester ist ein Wert von $18nF$ vorbesetzt. Die Werte für den Parallelkondensator werden relativ hoch gewählt, um die Eigenfrequenz des Schwingkreises auch bei kleineren Induktivitäten klein zu halten. Der Kondensator für die Parallelschaltung sollte eine hohe Güte haben (Folien-Typ), da auch die Güte des Schwingkreises aus der Abnahme der Spannungsamplitude bestimmt wird. Bei hoher Güte des Kondensators ist im Regelfall die Spule bestimmend für die Güte des Schwingkreises.

Für die Bedienung ist bis auf die Parallelschaltung des Kondensators keine weitere Aktion erforderlich. Der Schwingkreis wird dann normalerweise automatisch erkannt. Falls der Schwingkreis entdeckt wurde, wird hinter der Induktivität der Text „if“ und die angenommene Kapazität des Parallelkondensators in Zeile 2 angegeben. Für diesen Fall wird der Widerstandswert der Spule am Ende der Zeile 1 mit ausgegeben. Dem Widerstandswert sollte man auf jeden Fall ohne parallel geschalteten Kondensator überprüfen, da die Widerstandsmessung am Schwingkreis oft nicht funktioniert! In einer weiteren Zeile wird die gemessene Schwingfrequenz und die Güte Q angegeben. Wenn kein Schwingkreis festgestellt wurde, wird der Widerstand und die Induktivität in Zeile 2 ausgegeben. In einer weiteren Zeile wird die Resonanzfrequenz und die Güte angegeben, wenn trotz fehlendem Parallelkondensator eine Eigenresonanz der Spule festgestellt wurde.

Für eine Luftspule mit 6 Windungen und einem parallelgeschaltetem $18nF$ Kondensator bestimmt die Samplingmethode folgendes Ergebnis:

260nH if 18.1nF
2306kHz Q=38.7

Der ATmega328 wurde dafür mit 8 MHz betrieben. In etwa das gleiche Ergebnis erzielt auch ein 25cm langer Kupferdraht, der zu einem großen Kreis gebogen wurde. Die gemessene Induktivität ist bei dieser Messung etwas zu hoch, da der Kondensator ein gewickelter Folienkondensator mit Eigeninduktivität war. In der folgenden Tabelle 5.7 sind die Meßergebnisse mit verschiedenen Spulen dargestellt, die mit einem Tester bei 16 MHz Taktfrequenz ermittelt wurden.

Cp= Lp=	6.68nF	11.4nF	18.2nF	20.3nF	33.3nF
3 turns, 13mm (91.4nH)	100nH 6.039MHz Q=29.9	116nH 4.358MHz Q=15.6	108nH 3.568Mhz Q=49.8	115nH 3.282MHz Q=12.1	111nH 2.619MHz Q=31.4
4 turns, 13mm (144.9nH)	141nH 5.172MHz Q=44.8	161nH 3.724MHz Q=16.0	151nH 3.03Mhz Q=46.2	152nH 2.86MHz Q=14.6	153nH 2.226MHz Q=30.5
6 turns, 13mm (212.5nH)	217nH 4.18MHz Q=30.5	232nH 3.094MHz Q=18.4	223nH 2.492Mhz Q=43.0	224nH 2.343MHz Q=15.4	227nH 1.832MHz Q=31.7
12 turns, 13mm (569.5nH)	547nH 2.632MHz Q=36.9	571nH 1.973MHz Q=26.4	559nH 1.573Mhz Q=50.6	560nH 1.491MHz Q=20.8	566nH 1.16MHz Q=39.2
27 turns, 11mm (1.9 μ H)	1.93 μ H 1.403MHz Q=36.5	1.92 μ H 1.067MHz Q=33.4	2.02 μ H 828.5kHz Q=43.6	2.00 μ H 789.5kHz Q=26.6	2.01 μ H 615.4kHz Q=34.5
6.3 μ H 7.12 μ H	6.69 μ H 752.9kHz Q=28.5	6.84 μ H 570.2kHz Q=30.5	6.84 μ H 449.9kHz Q=32.3	6.82 μ H 428.1kHz Q=25.5	6.90 μ H 332.3kHz Q=28.3

Tabelle 5.7. Meßergebnisse von einigen kleinen Induktivitäten

Die Kondensatoren in dieser Tabelle sind Exemplare mit niedriger Induktivität wie die WIMA MKS Serie. Die Spule mit 4 Windungen ergibt mit einem gewickelten 18.2nF Kondensator eine Induktivität von 196nH anstelle der 151nH aus der Tabelle. Mit Ausnahme der letzten Induktivität handelt es sich um selbstgewickelte Spulen, deren berechnete Induktivität in Klammern angegeben ist. Die letzte 6.3 μ H Spule ist ein industriell gefertigtes Exemplar, das mit 6.3 μ H beschriftet ist. Die Messung mit einem RCL-Meßgerät ergibt bei 100kHz aber eine Induktivität von 7.12 μ H! In der Tabelle fallen auch die unterschiedlichen Güten bei gleicher Spule und fast gleicher Parallelkapazität auf. Bei der Spule mit 12 Windungen beträgt die Güte mit dem 18.2nF Kondensator 50.2, bei der Parallelkapazität 20.3nF aber nur 20.8. Das legt den Verdacht eines Programmfehlers nahe. Zur Kontrolle werden in Bild 5.64 die Daten des ADC-Wandlers für die Spule mit 12 Windungen und den 18.2nF und 20.3nF Kondensatoren gegenübergestellt. Auch bei den Rohdaten ist der Unterschied für beide Varianten des Schwingkreises deutlich zu erkennen. Wahrscheinlich ist der verwendete Kondensatortyp die Ursache für den Unterschied bei der Güte.

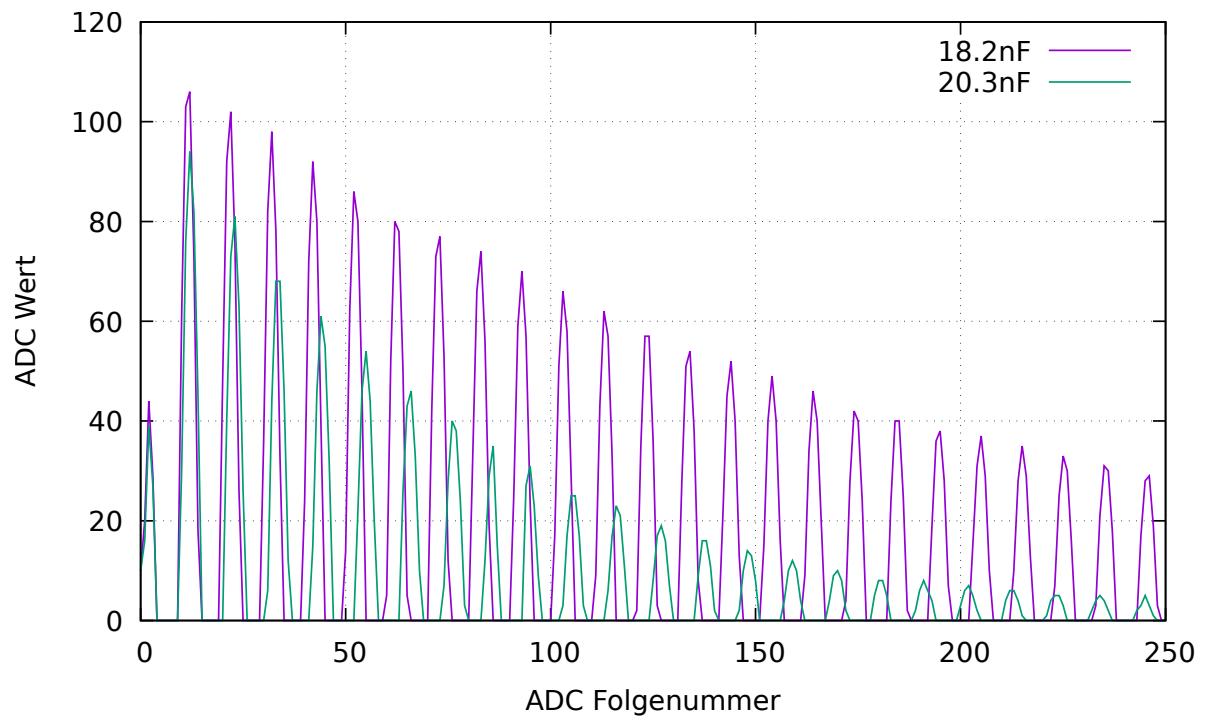


Abbildung 5.64. ADC Daten von zwei Schwingkreisen mit einer 12-Windungen Spule

5.5 Selbsttest-Funktion

Beginnend mit Version 0.9k habe ich eine Selbsttest-Funktion eingebaut. Die Benutzung ist einfach. Wenn Sie Prüfspitzen mit Klemmen verwenden, klemmen Sie alle Klemmen auf ein unisoliertes Stück Draht und drücken Sie den Start-Taster. Das Programm bemerkt die kurzgeschlossenen Klemmen und startet den Selbsttest, wenn innerhalb von zwei Sekunden durch einen Tastendruck der Starttaste bestätigt wird. Diese Bestätigung ist eingebaut worden, damit der Tester nicht von selbst beim Messen von einem defekten Transistor in den Selbsttest geht. Nach Beenden des Selbsttests wird der Tester mit normaler Messung fortfahren. Wenn kein Bauteil angeschlossen ist, wird der TransistorTester mit der Meldung „kein, unbek. oder defektes Bauteil“ enden. Man kann die Selbsttest-Funktion nur für einen ATmega168 oder ATmega328 wählen. Bevor die Testschritte durchgeführt werden, wird zuerst der Null-Widerstand für alle drei Testpin-Kombinationen (T1:T3, T2:T3 und T1:T2) bestimmt. Diese Nullwiderstände werden für die zukünftigen ESR- und Widerstands-Messungen unter 10Ω berücksichtigt. Es werden nur Nullwiderstandswerte unter 0.90Ω akzeptiert, da diese Korrekturwerte bei der Messung von Widerständen über 10Ω nicht berücksichtigt werden. Bei der Verwendung von Kabeln muß also auf entsprechend geringe Widerstandswerte geachtet werden. Falls die später gemessenen Widerstandswerte den jeweiligen Nullwiderstand um mehr als $0,2\Omega$ unterschreiten, wird der Tester auf „unkalibriert“ zurückgesetzt. Dies wird durch den aktvierten Positionszeiger (Cursor) beim Testen angezeigt. Die einzelnen Schritte der Selbsttestfunktion werden für Test 1 bis Test 7 in Zeile 1 des LC-Displays mit dem Buchstaben T gefolgt von der Schritt-Nummer dargestellt. Die Schritte 1 bis 7 werden viermal wiederholt, bevor das Programm zum nächsten Schritt geht. Aber wenn man den Start Taster gedrückt hält, wenn ein Durchlauf beendet wird, wird dieser Test nicht mehr wiederholt. Wenn man den Taster während dem gesamten Selbsttest gedrückt hält, wird jeder Schritt nur einmal ausgeführt.

Ohne die AUTO_CAL-Option werden in jedem Schritt nur Messergebnisse dargestellt, es wird keine Fehleranalyse durchgeführt. Man muss die Ergebnisse selbst interpretieren. An dieser Stelle möchte ich noch einen wichtigen Hinweis geben. Führen Sie niemals Messungen mit eingestecktem ISP-Stecker durch! Die ISP-Schnittstelle stört die Messungen.

Hier ist die Liste der derzeit eingebauten Tests:

1. **Messung der 1,3V (oder 1,1V) Referenzspannung (band gap Reference).** In Zeile 1 wird der Text „Ref=“ und die gemessene Spannung in mV angezeigt. Für den ATmega8 sollte die gemessene Spannung nahe bei 1,3V liegen, für die anderen Prozessoren liegt die Referenzspannung normal um die 1,1V. Die zweite Zeile zeigt den resultierenden Faktor für die Kapazitäts-Messung mit dem $470k\Omega$ Widerstand.
2. **Vergleich der 680Ω Widerstände.** In Zeile 1 wird der kryptische Text „+RL- 12 13 23“ angezeigt. Das bedeutet: RL ist eine Kurzform für Resistor Low, was die 680Ω Widerstände meint. Die „12“ steht für: Widerstand am Pin 1 ist verbunden mit VCC (+) und Widerstand am Pin 2 ist verbunden mit GND (-). Das Ergebnis dieser Messung steht in Zeile 2 an erster Stelle als Differenz zum theoretischen Wert. In Zeile 1 folgt nun „13“, was bedeutet, dass der Widerstand von Pin 1 wieder nach VCC verbunden ist, aber dass jetzt der 680Ω Widerstand von Pin 3 mit GND verbunden ist. Das Ergebnis steht in Zeile 2 an der mittleren Stelle als Differenz zum theoretischen Wert.

Die letzte Messung dieses Tests „23“ bedeutet, dass nun der Widerstand von Pin 2 mit VCC verbunden ist und der Widerstand von Pin 3 mit GND verbunden ist. Das Ergebnis steht in der zweiten LCD-Zeile an letzter Stelle als Differenz zum theoretischen Wert. Ich möchte daran

erinnern, dass die Auflösung des ADC etwa $4,88mV$ beträgt! Die Messsituation wird auch in Abbildung 5.65 dargestellt. Der theoretische Wert ergibt sich im Hinblick auf die internen Port-Widerstände wie folgt: $\frac{5001 \cdot (19+680)}{(19+680+680+22)} = 2493$.

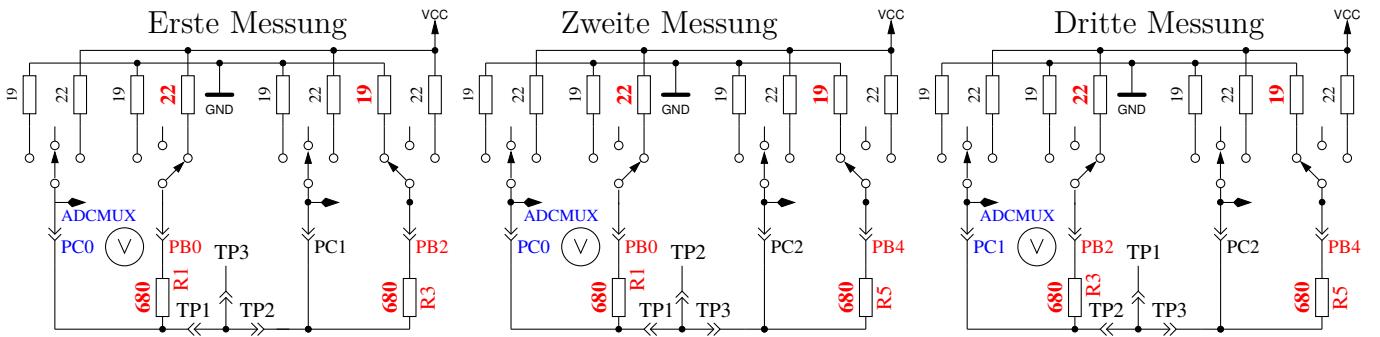


Abbildung 5.65. Vergleich der 680Ω -Widerstände

3. **Vergleich der $470k\Omega$ Widerstände.** Jetzt zeigt das Display in Zeile 1 „+RH- 12 13 23“. Das gleiche Verfahren wie in Schritt 2 wird wiederholt mit den $470k\Omega$ -Widerständen (Symbol RH). Die Ergebnisse werden als Differenz zu $\frac{VCC \cdot (19+470000)}{(19+470000+470000+22)}$ für alle Kombinationen dargestellt.
4. In diesem Schritt wird nichts gemessen, es erfolgt lediglich die **Anweisung „isolate Probe!“**, was bedeutet, dass es nun Zeit ist die Klemmen zu trennen (vom blanken Draht lösen). Dieser Schritt wird nur beendet, wenn Sie die Verbindung zwischen den Testports getrennt haben.
5. Dieser Schritt testet die **Fähigkeit von mit GND (-) verbundenen $470k\Omega$ -Widerständen (H) die Testpins auf GND zu ziehen**. Zeile 1 zeigt den Text „RH-“. Zeile 2 sollte für alle drei Pinne Null mV anzeigen.
6. Dieser Schritt testet die **Fähigkeit von mit VCC (+) verbundenen $470k\Omega$ -Widerständen (H) die Testpins auf VCC zu ziehen**. Zeile 1 zeigt den Text „RH+“. Der bestmögliche mV Wert für die drei Messungen wäre 0 in Zeile 2, da die Diffenz zu VCC dargestellt wird. Große Abweichungen vom Idealwert für Schritt 5 und 6 sind Fehler wie Isolations-Problem, Flussmittelreste oder beschädigter Port.
7. **Dieser Schritt testet die Spannungen der $470k\Omega/680\Omega$ Widerstandsteiler.** Zeile 1 zeigt den Text „RH/RL“. Für alle drei Testpins wird die Abweichung von der erwarteten Spannung eines $470k\Omega / 680\Omega$ 5V-Teilers in Zeile 2 des Displays ausgegeben. Abweichungen von mehr als einige mV weisen auf einen Bestückungsfehler der Widerstände hin.
8. **Messung des Innenwiderstandes eines auf GND geschalteten Ausgangs.** Dieser und die folgenden Schritte werden nur bei gewählter AUTO_CAL-Option durchgeführt. Die internen Port-C-Widerstände von auf GND (-) geschalteten Ausgängen werden mit dem Strom der nach VCC (+) geschalteten 680Ω Widerstände gemessen, siehe Abbildung 5.66. Es werden nur die drei Pins des ADC-Ports vermessen, die Widerstands-Ports PB0, PB2 und PB4 können nicht ohne Veränderung der Hardware gemessen werden. Es wird angenommen, dass die Portwiderstände von unterschiedlichen Ports nahezu identisch sind. Der Widerstandswert wird im nächsten Schritt mit ausgegeben.

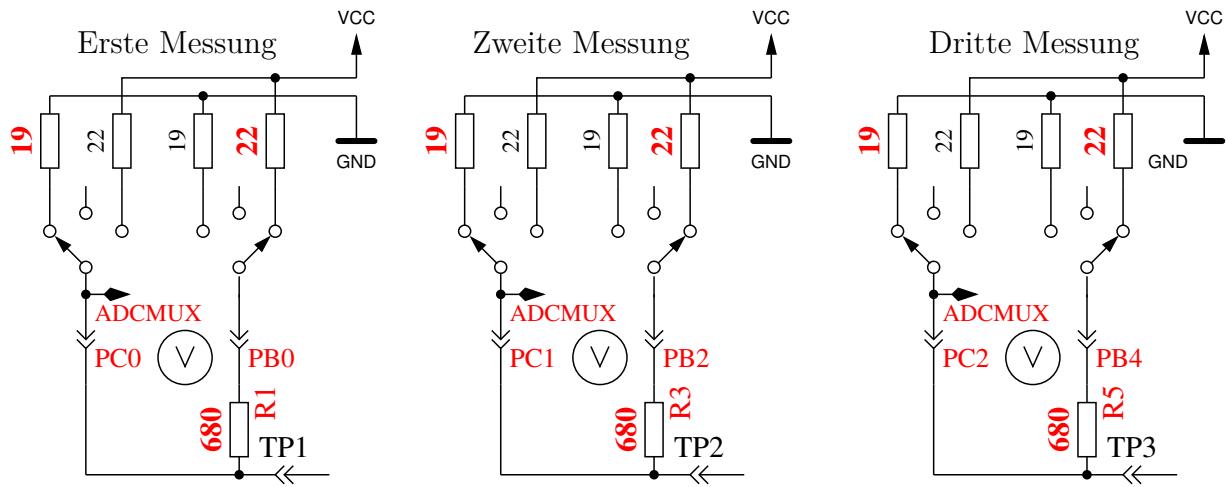


Abbildung 5.66. Messung des Innenwiderstandes von auf GND geschalteten Port-C-Ausgängen.

9. Messung des Innenwiderstandes von auf VCC geschalteten Port-Ausgängen. Der benötigte Strom wird von auf GND geschalteten 680Ω -Widerständen geliefert. Es ist die gleiche Messung wie die in Test 8 zur anderen Seite, wie in Abbildung 5.67 gezeigt wird. Der Innenwiderstand wird wie folgt berechnet: $(5001 - (\text{Ergebnis von Test 8}) - (\text{Ergebnis von Test 9})) / 680$. Die Widerstandswerte ergeben sich, wenn die gemessene Spannung durch diesen Strom dividiert wird. Das Ergebnis für diesen Test wird dann in Zeile 1 mit dem Text „RI_Hi=“ in Ω angezeigt, der Innenwiderstand zur GND-Seite wird in Zeile 2 mit dem Text „RI_Lo=“ angezeigt. Diese Werte werden ab Software-Version 1.06k bei jeder Messung neu bestimmt und hier nur dargestellt.

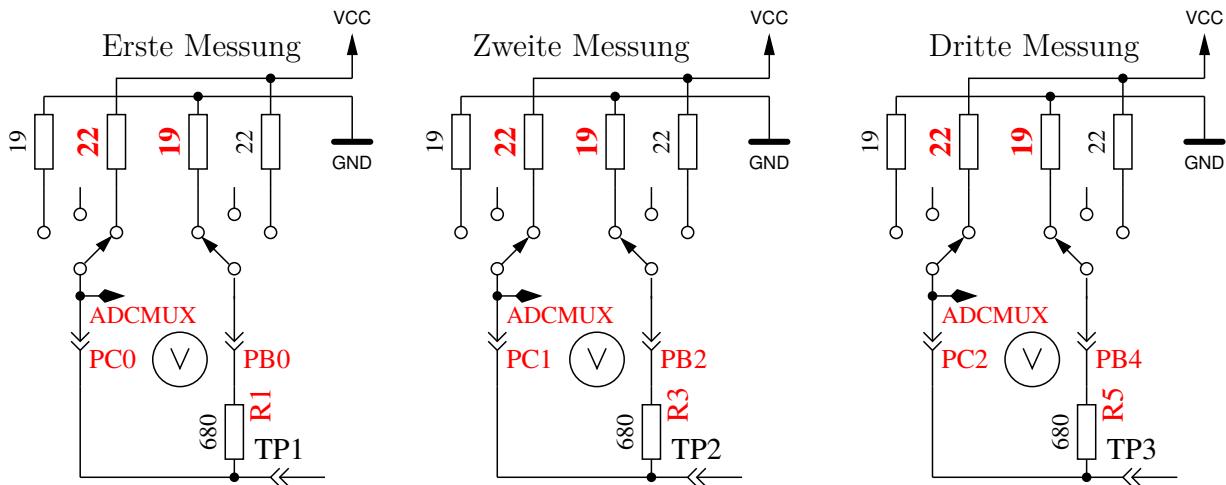


Abbildung 5.67. Messung des Innenwiderstandes des auf VCC geschalteten Port-C-Ausgängen.

10. Messung des Nulloffsets der Kondensatormessung. Für die Pinkombinationen 1:3, 2:3 und 1:2 wird der Nullwert der Kondensatormessung in pF in Zeile 1 hinter dem Text „C0“ ausgegeben. In der Software wird ein Vorgabewert von circa $39pF$ für die normale Messwertausgabe berücksichtigt. Für die Ausgabe dieses Tests ist keine Korrektur berücksichtigt, es wird auch kein Nulloffset abgezogen. Es werden auch die Nulloffsets für die umgekehrte Pinkombination ermittelt. Die gefundenen Nulloffsets werden im EEPROM festgehalten, wenn alle Nulloffsets kleiner als $190pF$ sind. In Zeile 2 wird dann ein „OK“ ausgegeben. Die gefundene Nulloffsets werden für die weiteren Kapazitäts-Messungen pinabhängig berücksichtigt. Dabei wird überwacht, ob die gemessene Kapazität die festgehaltene Nullkapazität um mehr als $20pF$ unterschreitet. Wenn dies der Fall sein sollte, wird der Tester auf „unkalibriert“ zurückgesetzt. Dies wird durch die Aktivierung des LCD-Positionszeigers (Cursor) beim nächsten Testen

angezeigt. Bitte beachten Sie, dass bei Veränderungen des Messaufbaus ein Neuabgleich sinnvoll ist. So kann sich der Nulloffset durch Verwenden von Kabeln mit Klemmen um circa $3pF$ gegenüber einem leeren Sockel erhöhen. Wenn der Tester mit der SamplingADC Funktion konfiguriert wurde, wird zusätzlich die Nullkapazität für die SamplingADC Messmethode in doppelt so vielen Konfigurationen gemessen. Das kommt daher, weil die Nullkapazität in allen Pin-Kombinationen sowohl für den Ladevorgang als auch für den Entladevorgang bestimmt wird.

11. **Warten auf den Anschluss eines Kondensators an Pin 1 und Pin 3.** Die Meldung „1 —||— 3 >100nF“ wird in Zeile 1 des Displays angezeigt. Zur Vorbereitung der Messung des Spannungs-Offsets des analogen Komparators muss ein ausreichend großer Kondensator zwischen Pin 1 und Pin 3 angeschlossen werden. Es sollte ein Kondensator hoher Güte mit einer Kapazität zwischen $100nF$ und $20\mu F$ sein. Auf keinen Fall sollte man Elektrolyt-Kondensatoren verwenden.
12. **Messung des Komparatoroffsets für den Abgleich der Kondensatormessung.** Um den Offset des analogen Komparators zu bestimmen, muss ein Kondensator an Pin 1 und Pin 3 angeschlossen sein. Der Kondensator wird für die Pufferung der Ladespannung bei der Kondensatormessung benötigt, um den Unterschied der Ladespannung zur internen Referenzspannung bestimmen zu können. Bei erfolgreicher Messung wird der Korrekturwert kurz in Zeile 1 mit dem Text „REF_C=“ angezeigt und in den EEPROM geschrieben.

Wenn Sie die AUTOSCALE_ADC-Option gewählt haben, wird die Verstärkung der ADC-Lesefunktion mit der internen Referenzspannung abgeglichen. Das geschieht durch den Vergleich der Kondensatorspannung von unter 1V, einmal mit der VCC-Referenz gemessen und einmal mit der internen Referenz. Die gefundene Differenz wird in Zeile 2 mit dem Text „REF_R=“ angezeigt und ebenfalls im EEPROM festgehalten. Der REF_R_KORR-Wert ist dann nur ein zusätzlicher Offset für diese automatisch gefundene Differenz.

13. **Warten auf einen Kondensator für die Messung kleiner Induktivitäten** Wenn der Tester mit der SamplingADC Funktion konfiguriert wurde, ist für die Messung kleiner Induktivitäten ein Kondensator bekannter Größe für die Berechnung der Induktivität aus der Schwingfrequenz erforderlich. Brauchbare Kapazitätswerte sind hierfür etwa $10nF$ bis $27nF$. Ein geeigneter Kondensator sollte an Pin 1 und Pin 3 angeschlossen werden, wenn die Meldung „1 —||— 3 10-30nF(L)“ in Zeile 1 des Displays erscheint. Genau dieser Kondensator sollte für die Induktivitätsmessung der Spule parallelgeschaltet werden, wenn diese Art der Induktivitätsbestimmung gewünscht wird.

Am Ende der Selbsttest-Funktion wird der Text „Test End“ in Zeile 1 und die Versions-Nummer der Software in Zeile 2 angezeigt. Wenn gewählt, wird ein **50 Hz Rechtecksignal** auf Testpin 2 generiert und das gegenphasige Signal auf Testpin 3. Testpin 1 wird auf GND geschaltet. Der Strom für Testpin 2 und 3 wird mit 680Ω Widerständen begrenzt. Angezeigt wird das durch die Ausgabe von „50Hz“ am Ende der Zeile 1. Diese Ausgabe wird dreißig Mal mit jeweils 2 Sekunden Dauer durchgeführt. Man kann die Zeit der Verzögerungs-Aufrufe prüfen, wenn man ein Oszilloskop oder einen Frequenz-Zähler besitzt. Abbildung 5.68 zeigt ein Oszillosrogramm der ausgegebenen Pegel mit Quarz-Betrieb.

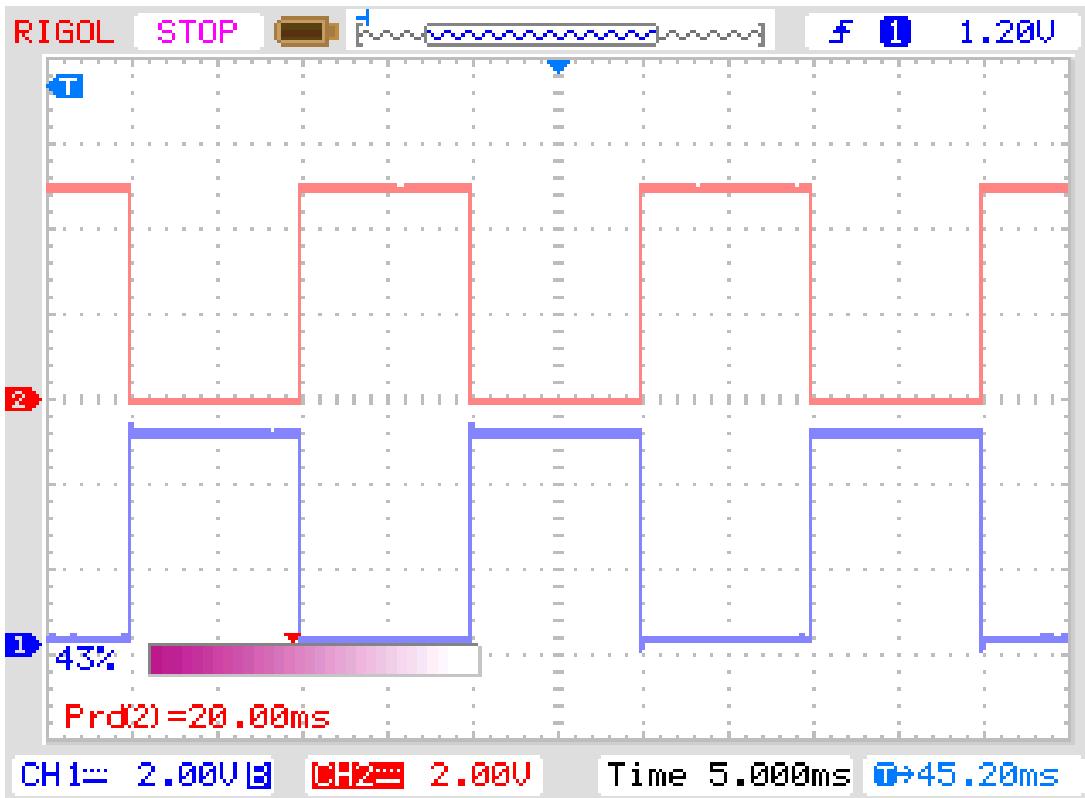


Abbildung 5.68. Oszillogramm der 50Hz Ausgänge von Port 2 und 3

Wenn man für die Takterzeugung keinen Quarz benutzt, können die Ergebnisse der Kondensatormessung ungenau sein. Eine genaue Taktfrequenz und genaue Wartezeiten sind wichtig für die Bestimmung von Kapazitätswerten. Die Ausgabe des 50Hz-Signals kann durch Gedrückthalten des Start-Tasters vorzeitig abgebrochen werden. Dann fährt das Programm mit der normalen Messfunktion fort.

5.5.1 Einige Selbsttest-Ergebnisse

Die Ergebnisse des Selbsttests der 9 verschiedenen ATmega168-Prozessoren und der 6 ATmega328-Prozessoren sind in den folgenden Abbildungen dargestellt.

Test No.	Messtyp	Idealwert	Abbildung
Test 1	band gap Ref	1100	5.69
Test 2	RL Mitte	0	5.70a
Test 3	RH Mitte	0	5.70b
Test 5	RH Low	0	5.71a
Test 6	RH High	0	5.71b
Test 8	R out Lo	131	5.72a
Test 9	R out Hi	151	5.72b
Test 10	Cap0	30	5.73
Test 11	Referenz-Korrektur	0	5.74

Tabelle 5.8. Verzeichnis der Selbsttest-Diagramme

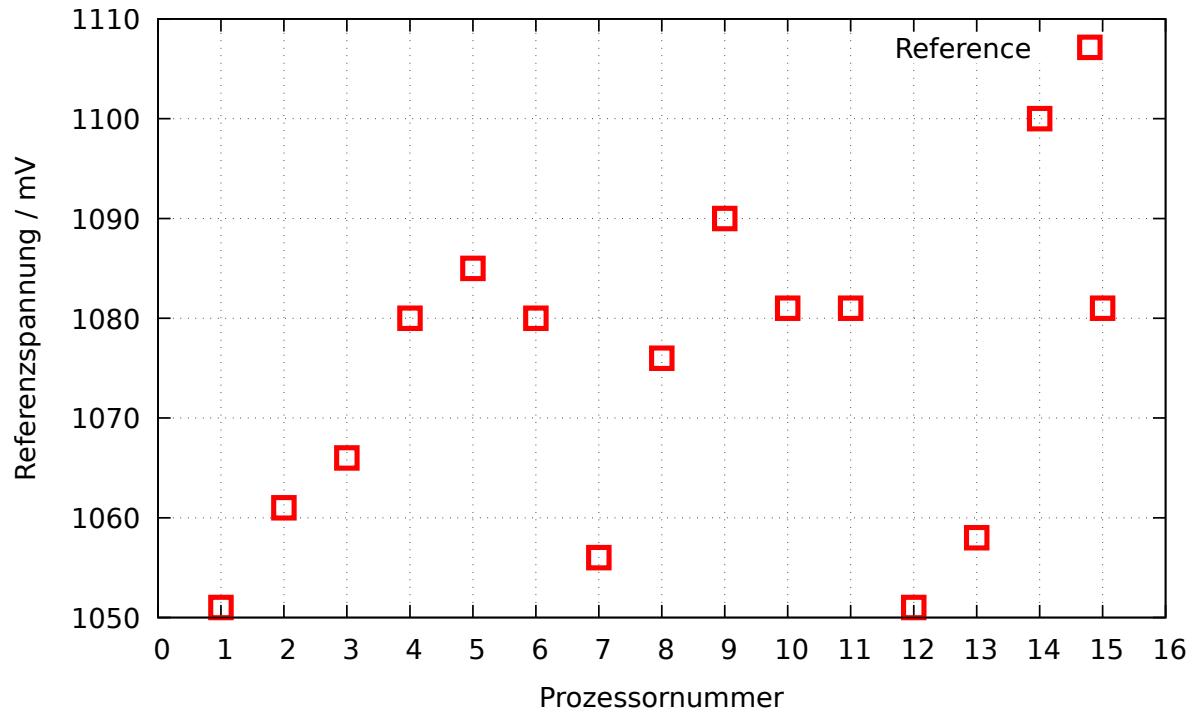
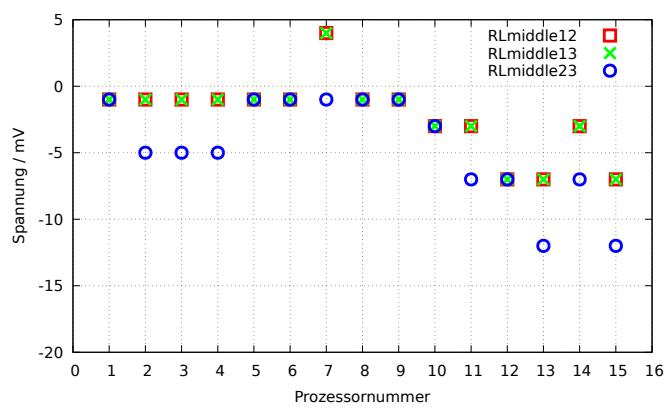
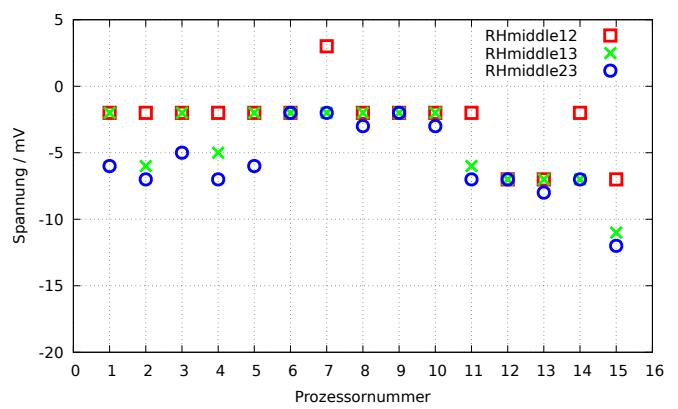


Abbildung 5.69. Selbsttest: Referenz-Spannungen

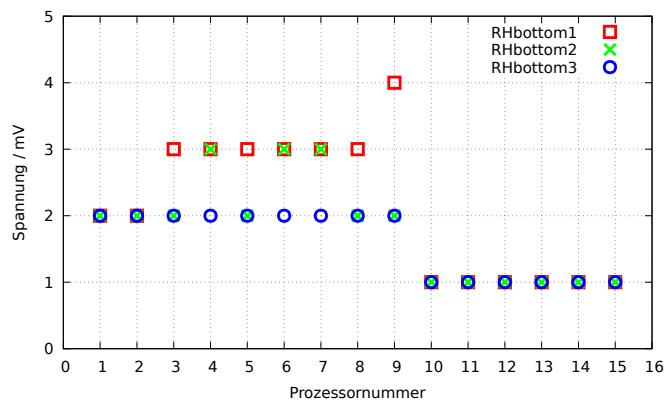


(a) mit 680Ω

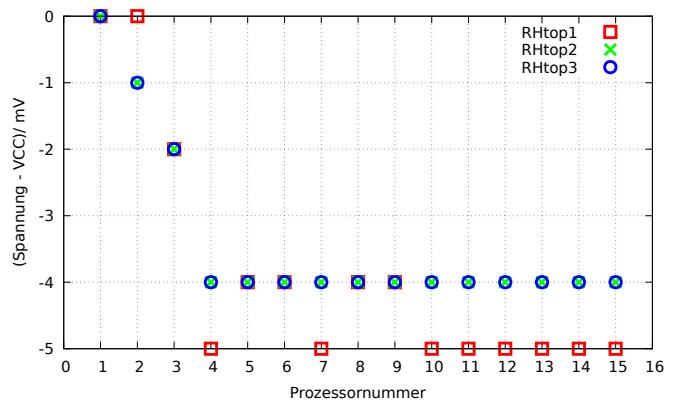


(b) mit $470k\Omega$

Abbildung 5.70. Selbsttest: Abweichung der Mittenspannung

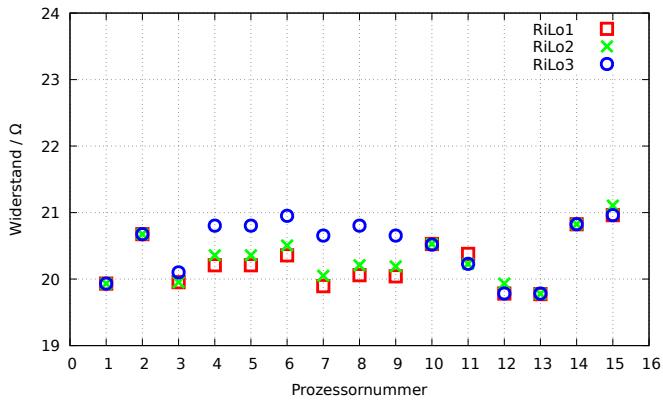


(a) mit $470k\Omega$ auf $0V$

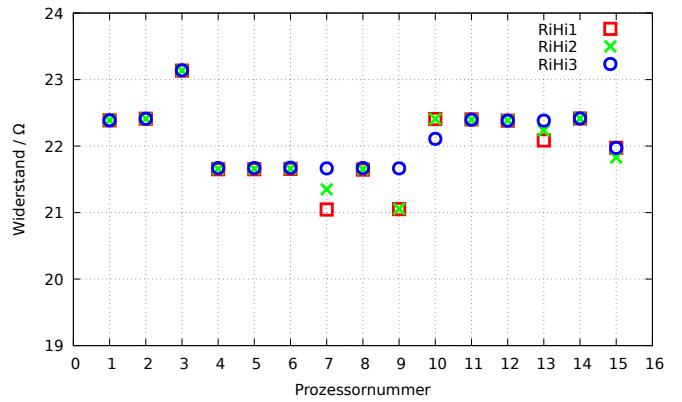


(b) mit $470k\Omega$ auf $5V$

Abbildung 5.71. Selbsttest: Eingangsspannung



(a) mit 680Ω auf $5V$



(b) mit 680Ω auf $0V$

Abbildung 5.72. Selbsttest: Ausgangswiderstand

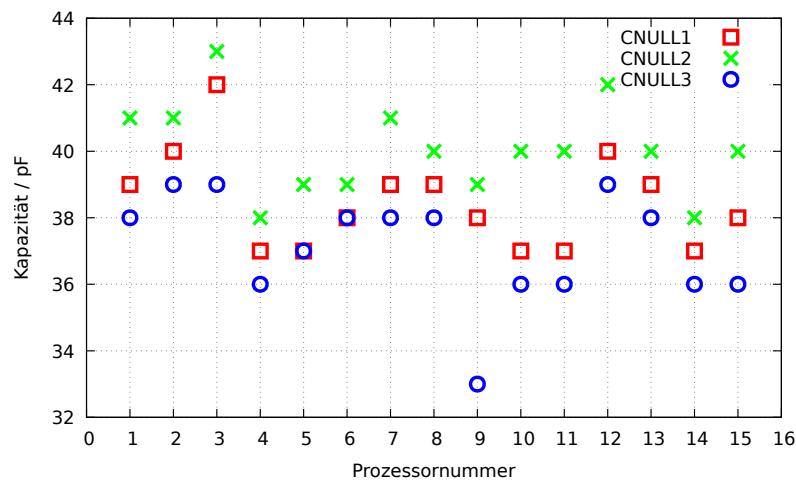


Abbildung 5.73. Selbsttest: Nullwert der Kapazitätsmessung

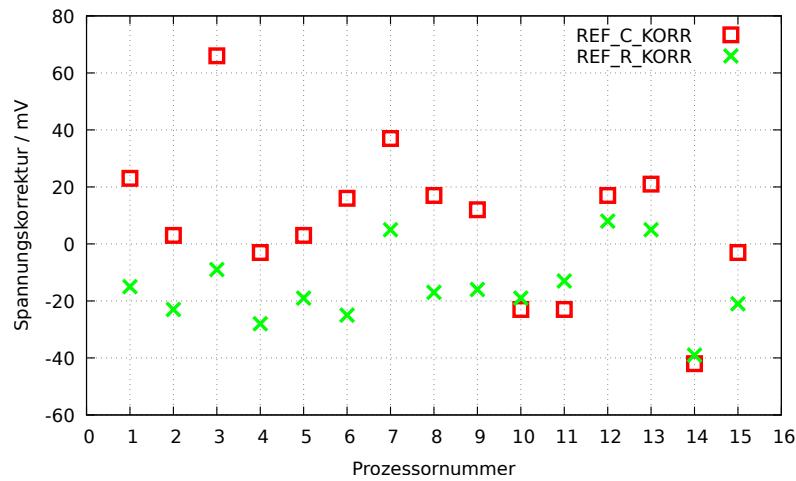


Abbildung 5.74. Selbsttest: Korrekturwerte nach automatischer Kalibration

Zuletzt möchte ich die Differenzen der am AREF-Pin mit einem Multimeter gemessenen Spannungen und der intern mit dem ADC gemessenen Referenzspannungen der 15 verschiedenen ATmegas und die mit dem automatischen Abgleich gefundenen Korrekturspannungen (REF_R_KORR) in Abbildung 5.75 gegenüberstellen. Sie können sehen, dass die Werte der automatischen Kalibration nahezu den extern gemessenen Referenz-Spannungswerten folgen.

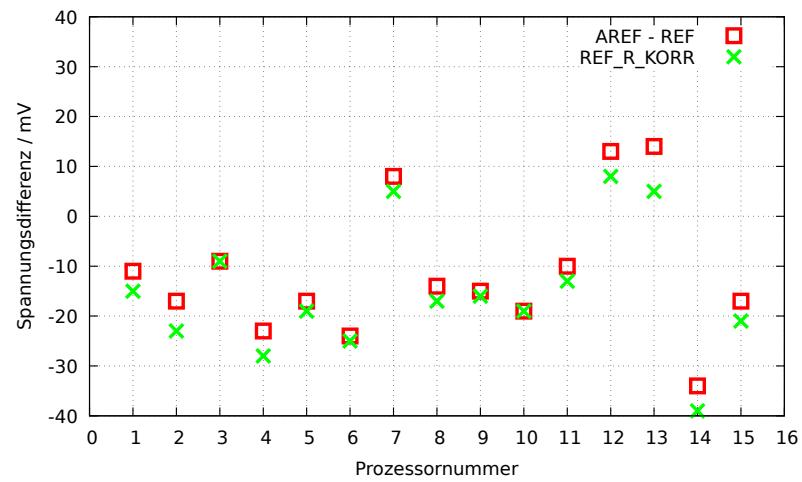


Abbildung 5.75. Selbsttest: Spannungsdifferenz der internen Referenz

5.6 Frequenzmessung

Beginnend mit Version 1.10k kann die Frequenzmessung mit einem Bedienmenü angewählt werden. Die normale Frequenzmessung wird durch Zählen der fallenden Flanken des Eingangssignals T0 (PD4) mit dem Zähler 0 (COUNTER0) für eine Sekunde erledigt. Für die Einhaltung einer exakten Sekunde wird der Zähler 1 mit einem 256:1 Verteiler der CPU-Frequenz benutzt. Der 16-Bit Zähler des ATmega kann mit dem Verteiler auch bei 16 MHz CPU-Frequenz in einem Durchlauf eine Sekunde auszählen. Für das Starten und das Stoppen des Zählers 0 werden die Vergleichregister B und A des Zählers 1 benutzt. Damit keine zusätzliche Zeitunsicherheit bei den Abfragen entsteht, werden die Interrupt-Service-Routinen für beide Vergleichs-Ereignisse benutzt. Die Zeitverzögerung durch die beiden Interrupt-Service-Routinen ist ungefähr gleich. Für die Einhaltung der exakten Sekunde ist eine konstante Zeitverzögerung unerheblich. Durch Analyse des Assemblercodes kann die Zeitungleichheit ausgeglichen werden.

Bei Frequenzen unter $33kHz$ wird die Messung durch die Messung einer Periodendauer ergänzt. Diese Messung wird im Anschluss an eine normale Frequenzmessung durchgeführt. Dabei werden die Zeit einer bestimmten Anzahl von Pegelwechsel des PCINT20-Eingangs (PD4) mit dem Zähler 0 gemessen. Bei der Periodenmessung sollte sowohl die negative Pulsweite als auch die positive Pulsweite in jeder Periode mindestens $10\mu s$ betragen. Der Zähler 0 läuft dabei mit voller CPU-Taktrate und das ergibt für eine Periode eine Auflösung von $125ns$. Durch eine größere Anzahl von gemessenen Perioden kann die Auflösung verbessert werden. Bei 125 Perioden (250 Pegelwechsel) ergeben sich so schon eine mittlere Auflösung für eine Periode von $1ns$. Damit keine Ungenauigkeit beim Starten und Stoppen von Zähler 0 entsteht, wird der Zähler 0 mit dem ersten Pegelwechsel von PCINT20 gestartet und nach der vorgegebenen Zahl über die gleiche Interrupt Service Routine wieder gestoppt. Die Zahl der Perioden wird so gewählt, dass die Messzeit etwa 10 Millionen Takte beträgt. Der Fehleranteil eines Taktes macht dann 0,1 ppm aus. Bei $8MHz$ beträgt die Messzeit also etwa 1,25 Sekunden. Aus der so ermittelten mittleren Periode wird dann eine Frequenz mit höherer Auflösung berechnet.

Zur Kontrolle wurden zwei Tester gegeneinander gemessen. Erst wurden mit Tester 2 die Testfrequenzen erzeugt und mit Tester 1 die Frequenzen gemessen. Danach wurde die Messung mit vertauschten Testern wiederholt. Abbildung 5.76 zeigt die Ergebnisse. Die nahezu konstanten relativen Abweichungen sind durch die geringen Frequenzunterschiede der beiden Quarze zu erklären.

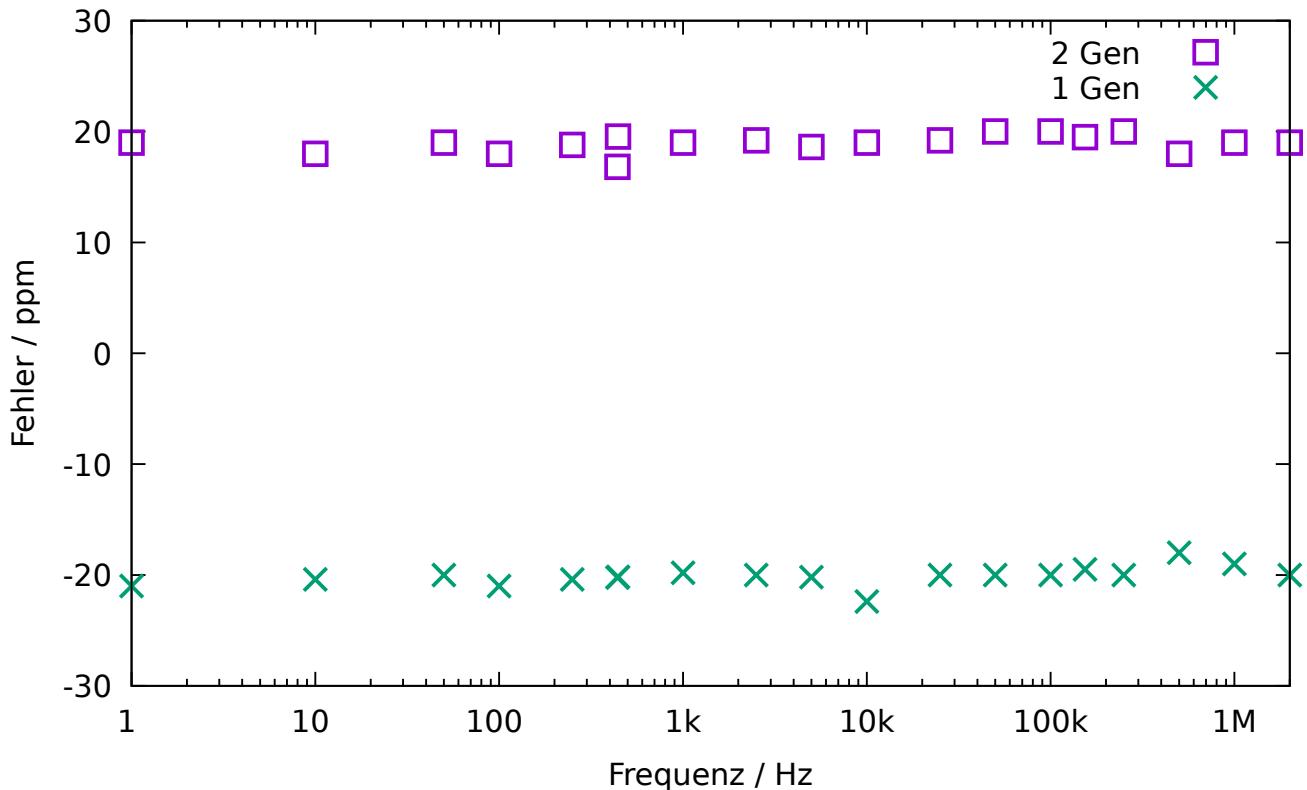
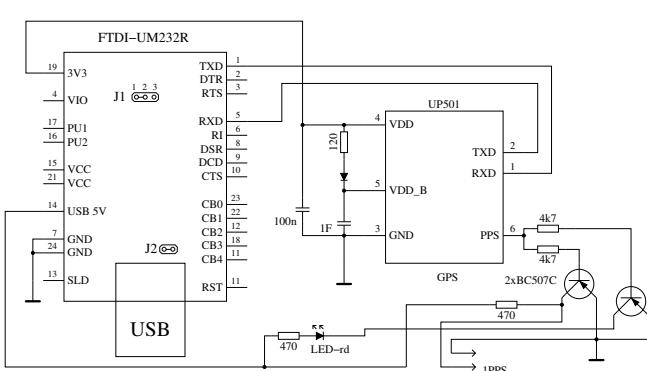


Abbildung 5.76. Relativer Fehler für Frequenzmessung

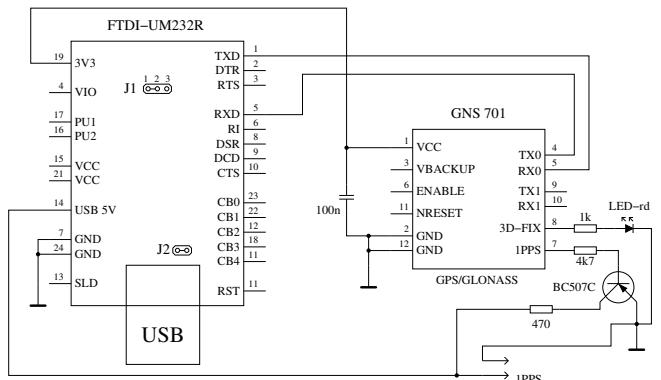
5.6.1 Kalibration der Frequenz mit GPS- oder GLONASS-Empfänger

Eine Abstimmung der Quarzfrequenz des Transistorstesters ist mit einem einstellbaren Kondensator (5 – 25pF Trimmer) am Quarz möglich. Als Referenz zum Abstimmen habe ich das Sekundensignal (1PPS) mit dem GPS-Empfänger **UP501** von **Fastrax Ltd.** und mit dem GPS/GLONASS-Empfänger **GNS701** von **Global Navigation Systems GmbH** erfolgreich getestet. Die gemessene Periode ließ sich mit dem Trimmer auf exakt 1000,000ms abgleichen. Lediglich die letzte angezeigte Ziffer kann um 1 schwanken. Natürlich ist die Frequenz des Quarzes temperaturabhängig. Deswegen kann man keine sehr gute Langzeitstabilität erwarten.

Die Abbildung 5.77 zeigt die verwendeten Schaltungen mit einem UM232 USB-seriell-Wandler als Verbindung der Empfänger-Module mit einem Computer. Der UM232-Wandler versorgt die Schaltung sowohl mit 5V als auch mit 3,3V Betriebsspannung aus der USB-Versorgung. Für den Betrieb der Empfänger ist eine Verbindung zum Rechner nicht erforderlich. Lediglich die 5V-Versorgung der USB-Steckbuchse muss gewährleitet sein.



(a) GPS



(b) GPS/GLONASS

Abbildung 5.77. Sekundenpuls Erzeugung mit GPS-Empfängern

5.6.2 Kalibration der Quarzfrequenz mit einem Uhrenmodul

Zum Abstimmen der Quarzfrequenz des Transistortesters ist ein Austausch eines der beiden Quarz-Kondensatoren gegen einen Trimmer mit einstellbarer Kapazität erforderlich. Der Vorteil bei der Verwendung von Uhrenmodulen gegenüber von GPS- oder GLONASS-Modulen für den Frequenzabgleich ist, daß man keine freie Sicht zum Himmel braucht. Man kann den Abgleich nahezu an jedem Ort durchführen. Untersucht habe ich Uhrenmodule mit einem DS3231 und der Platinen-Aufschrift „ZS-042“. Die untersuchten Module stammen mit großer Wahrscheinlichkeit aus chinesischer Produktion und verwenden nicht den DS3231SN Chip, sondern einen DS3231M Chip mit einem MEMS Resonator (MEMS = Micro Electro Mechanical System). Einen ähnlichen MEMS Resonator benutzt übrigens auch der DCP1301 Chip. Das Bild 5.78 zeigt eins der verwendeten Module.

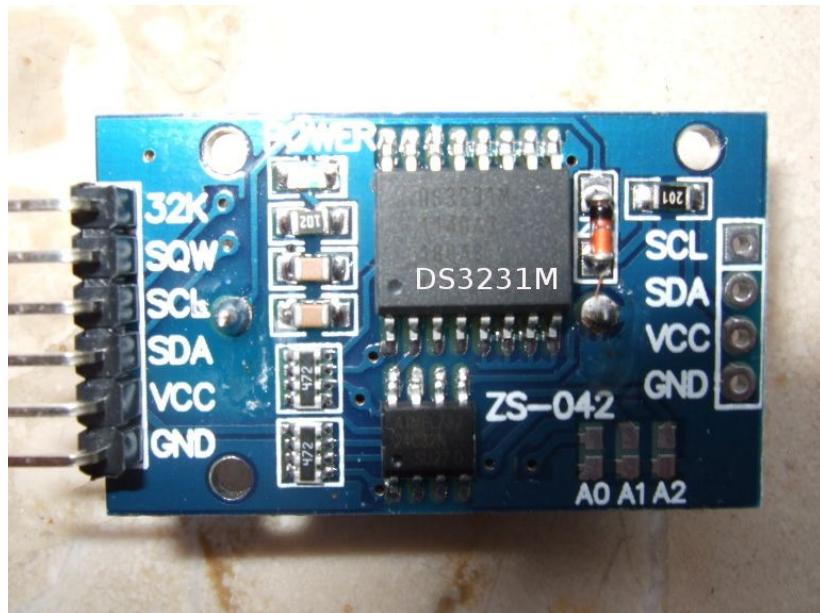


Abbildung 5.78. Ein Exemplar der untersuchten DS3231 Module

Der DS3231SN Chip benutzt aber einen Uhrenquarz mit 32768Hz als Zeitbasis. Bei beiden DS3231 Chip Varianten wird das Temperatur-Verhalten der Schwingfrequenz über eine Chip interne Temperaturmessung kompensiert, so daß eine gute Frequenzgenauigkeit in einem weiten Temperaturbereich erreicht wird. Leider ist aber das auf dem Modul herausgeführte 32kHz Signal bei der DS3231M Chip Variante für eine Kalibration ungeeignet. Bei den vorhandenen vier Modulen waren Frequenzen von 32641Hz , 32710Hz , 32730Hz und 32748Hz meßbar. Alle Frequenzen waren also weit entfernt von den eigentlich erwarteten präzisen 32768Hz . Wenn man die Module an einen Arduino UNO anschließt, kann man auch einen 1PPS (1Hz) Ausgang auf den Ausgang SQW schalten. Dieser Ausgang ist aber so stabil in der Periode, daß er für eine Kalibration benutzt werden kann. Das Datenblatt des DS3231M verspricht für den 1PPS Ausgang eine Genauigkeit von $\pm 5\text{ppm}$ über den vollen Temperaturbereich -45°C bis $+85^\circ\text{C}$, während die Genauigkeit des 32kHz Ausgangs nur auf $\pm 2,5\%$ (25000ppm) garantiert wird.

Im Datenblatt des DS3231SN wird über einen Temperaturbereich -40°C bis $+85^\circ\text{C}$ eine Genauigkeit von $\pm 3.5\text{ppm}$ versprochen und für den Temperaturbereich 0°C bis $+40^\circ\text{C}$ eine Genauigkeit vom $\pm 2\text{ppm}$. Der DS3231SN Chip verwendet als Zeitbasis einen Uhrenquarz mit der Frequenz 32768Hz , der mit schaltbaren Kapazitäten über einen weiten Temperaturbereich nachgezogen wird. Mit dem bekannten Temperaturgang des Quarzes und einem auf den Chip integrierten Temperatursensor wird die Frequenz weitgehend konstant gehalten. Um auch diesen Chip untersuchen zu können, habe ich die DS3231M Chips bei allen vier Modulen gegen DS3231SN Chips ausgetauscht. Bei einem frisch kalibriertem Transistortester mit 16MHz Quarz ergibt die Frequenzmessung für alle vier Module die gleiche Frequenz von 32.76800kHz mit einer selten angezeigten Abweichung von

0.03Hz . Das entspricht einer Abweichung von etwa 1ppm . Bruchteile von 1Hz werden übrigens nur angezeigt, wenn die Frequenz aus einer Periodenmessung errechnet wird. Damit die 32768Hz mit dieser Methode gemessen werden können, habe ich die Grenze für die Periodenmessung von 25kHz auf 33kHz geändert. Diese Genauigkeit der Frequenzanzeige kann natürlich nur erreicht werden, wenn die Frequenzquelle entsprechend rauscharm ist. Ein Signal, welches von einem RC-Generator abgeleitet ist, schwankt erheblich bei den Anzeigewerten.

Kapitel 6

Signal-Erzeugung

Die verschiedenen Signalerzeugungs-Betriebsarten können nur für einen ATmega328-Prozessor gewählt werden. Sie müssen außerdem die Dialogfunktion mit der Makefile-Option WITH_MENU eingeschaltet haben. Das Bedienmenü kann mit einem langen Tastendruck aufgerufen werden. Die wählbaren Funktionen werden in der zweiten Zeile des LCD angezeigt. Die angezeigte Funktion kann man mit einem langen Tastendruck anwählen. Die nächste Funktion wird automatisch in 5 Sekunden oder nach einem kurzen Tastendruck angezeigt.

6.1 Frequenz-Erzeugung

Die Frequenzerzeugung wird gestartet, wenn die Menüfunktion „f-Generator“ mit einem langen Tastendruck angewählt wird. Die Ausgabe der Frequenz erfolgt über den 680Ω Widerstand auf den Messport TP2. Der Messport TP1 wird auf GND geschaltet. Die Frequenzen werden mit dem 16-Bit Zähler aus der CPU-Taktfrequenz erzeugt ($8MHz$ oder $16MHz$). Die Frequenz kann dekadenweise beginnend mit der 1Hz Stelle mit den Ziffern 0-9 eingestellt werden. Die höchste wählbare Stelle ist die 100kHz Stelle. Hier können Zahlen bis zu 20 eingestellt werden. Somit sind Ausgabefrequenzen bis 2Mhz einstellbar. Ohne Drehimpulsgeber kann der Stellenwert durch einen kurzen Tastendruck ($< 0.8s$) erhöht werden. Mit Drehimpulsgeber können die Stellenwerte beliebig erhöht oder erniedrigt werden. Mit einem längeren Tastendruck wird die Stelle gewechselt. Dabei wird in Spalte 1 angezeigt, in welche Richtung die nächste Stelle durch den längeren Tastendruck gewechselt wird. Bei einem $>$ Zeichen in Spalte 1 der Frequenzzeile wird die nächsthöhere Stelle gewählt. Bei einem $<$ Zeichen in Spalte 1 wird die nächstniedrige Stelle gewählt (bis 1Hz). Wenn die höchste Stelle (100kHz) eingestellt ist, wird anstelle des $>$ Zeichen ein R angezeigt. Dann bewirkt ein längerer Tastendruck ein Rücksetzen der Frequenz auf 1Hz. Weil mit dem Zähler nicht jede eingestellte Frequenz korrekt erzeugt werden kann, wird die Frequenzabweichung des erzeugten Signals in Zeile 3 oder hinter dem Frequenzwert gezeigt. Ein langer Tastendruck ($> 2s$) kehrt zu der Dialog-Funktion zurück, wo die gleiche oder eine andere Funktion gewählt werden kann.

6.2 Pulsweiten-Erzeugung

Die Pulsweiten-Erzeugung wird gestartet, wenn Sie die Funktion „10-Bit PWM“ mit einem langen Tastendruck anwählen. Die Ausgabe der Frequenz erfolgt über den 680Ω Widerstand auf den Messport TP2. Der Messport TP1 wird auf GND geschaltet. Die Ausgabefrequenz ist fest und ergibt sich aus der CPU-Taktrate dividiert durch 1024. Das ergibt eine Frequenz von $7812,5Hz$ für die Taktrate $8MHz$. Nur die positive Pulsweite kann mit einem Tastendruck verändert werden. Mit einem kurzen Tastendruck wird die positive Pulsweite um 1% bis zu 99% erhöht. Mit einem längeren Tastendruck wird die Pulsweite um 10% erhöht. Wenn die Pulsweite Werte über 99% erreicht, wird 100 vom

Ergebnis abgezogen. Die Pulsweite 0% erzeugt einen sehr kurzen positiven Puls.

Kapitel 7

Bekannte Fehler und ungelöste Probleme

Software-Version 1.12k

1. Germanium-Dioden (AC128) werden nicht in allen Fällen entdeckt. Ursache ist vermutlich der Reststrom. Das Kühlen der Diode reduziert den Reststrom.
2. Bei bipolaren Transistoren wird eine Kollektor - Emitter Schutzdiode nicht erkannt, wenn der Kollektor-Reststrom ICE0 hoch ist. Das Problem trat bisher nur bei Germanium-Transistoren mit einer externen Diode auf.
3. Der Stromverstärkungsfaktor von Germanium-Transistoren kann zu hoch gemessen werden wegen dem hohen Reststrom. In diesem Fall ist die gemessene Basis-Emitter-Spannung auffällig klein. Das Kühlen des Transistors kann helfen, einen realistischeren Stromverstärkungsfaktor zu bestimmen.
4. Bei Leistungs-Doppeldioden vom Schottky-Typ wie MBR3045PT kann bei Anschluss einer Einzeldiode keine Kapazitätsmessung in Sperr-Richtung durchgeführt werden. Der Grund ist ein zu hoher Reststrom. Der Fehler kann manchmal durch Kühlen (Kältespray) umgangen werden.
5. Es ist gelegentlich zu einer Falscherkennung einer 2,5V Präzisionsreferenz gekommen, wenn der Anschluss PC4 (Pin 27) unbeschaltet ist. Abhilfe ist möglich mit einem zusätzlichen Pull-Up-Widerstand nach VCC.
6. Die Diodenfunktion des Gates eines Triac kann nicht untersucht werden.
7. Vereinzelt ist von Problemen mit der Brown-Out-Schwelle 4,3V für ATmega168- oder ATmega328-Prozessoren berichtet worden. Dabei kommt es zu Resets bei der Kondensatormessung. Eine Ursache ist nicht bekannt. Der Fehler verschwindet, wenn man die Brown-Out-Schwelle auf 2,7V einstellt.
8. Mit der Benutzung des Schlafzustandes des Prozessors schwankt die VCC-Stromaufnahme mehr als in älteren Softwareversionen. Sie sollten die Abblock-Kondensatoren überprüfen, wenn Sie irgendwelche Probleme feststellen. Keramische $100nF$ Kondensatoren sollten in der Nähe der Poweranschlüsse des ATmega angeschlossen sein. Sie können die Benutzung des Schlafzustandes auch mit der Makefile-Option INHIBIT_SLEEP_MODE verhindern.
9. Das Messen von Tantal-Elektrolyt-Kondensatoren macht oft Probleme. Sie können als Diode erkannt werden oder auch gar nicht erkannt werden. Manchmal hilft ein Polaritätswechsel.
10. Die Anschlüsse Source und Drain können bei JFETs nicht richtig ermittelt werden. Die Ursache liegt in dem symmetrischen Aufbau dieser Halbleiter. Erkennen kann man dieses Problem, dass die Anzeige mit den ermittelten Parametern gleich bleibt, wenn die Anschlüsse vertauscht

angeschlossen werden. Leider kenne ich keinen Weg, Source und Drain korrekt zu ermitteln. Aber das Vertauschen von Source und Drain in irgendeiner Schaltung sollte normalerweise kein Problem verursachen.

Kapitel 8

Spezielle Softwareteile

Es wurden zahlreiche Veränderungen durchgeführt um Flashspeicher zu sparen. Die LCD-Ausgabe von Test-Anschlussnummern wurde in der Form „lcd_data('1'+Pin)“ durchgeführt. Um den Platz für die zusätzliche Addition für jeden Aufruf zu sparen, wurde die Funktion „lcd_testpin(uint8_t pin)“ in die Datei lcd_routines.c eingefügt.

Die Pseudofunktionen in der Form _delay_ms(200) sind keine Bibliotheksfunktionen sondern es werden für jede Aufrufstelle Warteschleifen in das Programm eingebaut. Das verbraucht viel Speicher, wenn man viele Aufrufe an unterschiedlichen Programmstellen hat. Alle solche Aufrufe wurden durch Aufrufe einer speziellen in Assembler-Sprache geschriebenen Bibliothek ersetzt, welche nur 74 Bytes des Flashspeichers (bei 8MHz) verbraucht, aber Aufrufe von wait1us() bis wait5s() in den Stufen 1,2,3,4,5,10,20... zur Verfügung stellt. Die Routinen enthalten den „Watch Dog Reset“-Befehl für alle Aufrufe über 50ms. Jeder Aufruf benötigt nur eine Anweisung (2 Byte). Warte-Aufrufe mit Zwischenwerten wie 8ms benötigen zwei Aufrufe (wait5ms() und wait3ms() oder zwei wait4ms()-Aufrufe). Ich kenne keine Lösung, die ökonomischer wäre, wenn man viele Warteaufrufe im Programm benutzt. Die Aufrufe benutzen keine Register, nur der Stapelzeiger (Stack Pointer) wird für die Rückkehr-Adressen im RAM (maximal 28 Byte Stack-Platz) benutzt.

Die vollständige Liste der Funktionen ist:

```
wait1us(), wait2us(), wait3us(), wait4us(), wait5us(), wait10us(),
wait20us(), wait30us(), wait30us(), wait40us(), wait50us(), wait100us(),
wait200us(), wait300us(), wait400us(), wait500us(), wait1ms(),
wait2ms(), wait3ms(), wait4ms(), wait5ms(), wait10ms(),
wait20ms(), wait30ms(), wait40ms(), wait50ms(), wait100ms(),
wait200ms(), wait300ms(), wait400ms(), wait500ms(), wait1s(),
wait2s(), wait3s(), wait4s() und wait5s();
```

Das sind 36 Funktionen, die nur 37 Anweisungen inklusive dem Watch Dog Reset benutzen! Da gibt es keinen Weg diese Bibliothek zu verkürzen. Zu guter Letzt halten die Aufrufe die exakte Zeit ein, wenn der unterste Aufruf (wait1us) dies tut. Nur die Warte-Aufrufe über 50ms sind einen Takt pro 100ms zu lang wegen des zusätzlich eingebauten Watch Dog Reset.

Zusätzlich wurde die oft benutzte Folge „wait5ms(); ReadADC(...);“ durch einen einzelnen Aufruf „W5msReadADC(...);“ ersetzt. Dasselbe wurde auch für die Folge „wait20ms(); ReadADC(...);“ gemacht, die durch den Aufruf „W20msReadADC(...);“ ersetzt wurde. Die Funktion ReadADC wurde zusätzlich in Assembler portiert, sodass diese Erweiterung sehr effektiv integriert werden konnte. Eine funktionell identische C-Version der ReadADC-Funktion ist auch beigelegt.

Kapitel 9

Arbeitsliste und neue Ideen

1. Ergänze mehr und bessere Dokumentation.
2. Darüber nachdenken, wie man den wirklichen Innenwiderstand der Port-B-Ausgänge (Widerstands-Schalter) bestimmen kann, anstatt anzunehmen, dass die Ports gleich sind.
3. Kann das Entladen von Kondensatoren beschleunigt werden, wenn der Minus-Pol zusätzlich mit dem 680Ω - Widerstand nach VCC (+) geschaltet wird?
4. Prüfe, ob der Tester Fließkommadarstellung von Werten gebrauchen kann. Das Überlaufrisiko (overflow) ist geringer. Man braucht keine Konstruktion mit Multiplikation oder Division, um einen Faktor mit einer gebrochenen Zahl nachzubilden. Aber ich weiß nicht, wieviel Platz für die Bibliothek gebraucht wird.
5. Schreibe eine Gebrauchsanweisung zum Konfigurieren des Testers mit den Makefile-Optionen und beschreibe den Ablauf bis zum fertigen Prozessor.
6. Wenn der Haltestrom eines Thyristors nicht mit dem 680Ω Widerstand erreicht werden kann, ist es ungefährlich für eine sehr kurze Zeit die Kathode direkt auf GND und die Anode direkt auf VCC zu schalten? Der Strom kann mehr als $100mA$ erreichen. Wird der Port beschädigt? Was ist mit der Spannungsversorgung (Spannungsregler)?
7. Prüfe die Ports nach dieser Aktion mit der Selbsttest-Funktion!
8. Idee für ein neues Projekt: USB-Version ohne LC-Display, Power vom USB-Port, Kommunikation zum PC über eine USB-Serial-Brücke.
9. Ersatz der samplingADC Funktion mit Benutzung des Counter1, um den Zeitversatz des ADC S&H einzustellen.
10. Untersuchung der Meßgenauigkeit für kleine Kondensatoren mit der SamplingADC Methode.
11. Untersuchung der Meßgenauigkeit für kleine Spulen mit der SamplingADC Methode.

Literaturverzeichnis

- [1] <https://github.com/mikrocontroller-net/transistorTester/> *Quellen des TransistorTesters.* 02-2021
- [2] Markus Frejek *AVR-TransistorTester*,. Embedded Projects Journal, 11. Ausgabe, 2011
- [3] <https://github.com/kubi48/> *TransistorTester Dokumentation der Autors.*
- [4] Atmel Corporation *8-bit AVR with 8KBytes In-System Programmable Flash - ATmega8(L)*,. Manual, 2486Z-AVR-02/11, 2011
- [5] Atmel Corporation *8-bit AVR with 4/8/16/32KBytes In-System Programmable Flash - ATmega48 - ATmega328*,. Manual, 8271D-AVR-05/11, 2011
- [6] Atmel Corporation *Atmel AVR126: ADC of megaAVR in Single Ended Mode*,. Application Note, 8444A-AVR-10/11, 2011
- [7] Atmel Corporation *Atmel AVR121: Enhancing ADC resolution by oversampling*,. Application Note, 8003A-AVR-09/05, 2005
- [8] http://en.wikibooks.org/wiki/LaTeX_LaTeX_documentation,. Guide to the LaTeX markup language, 2012
- [9] http://en.wikibooks.org/wiki/Gnuplot_Gnuplot_documentation,. Documentation for the plotting tool gnuplot, 2012
- [10] Wikipedia http://de.wikipedia.org/wiki/Equivalent_Series_Resistance *Erläuterung zum ESR.* Normung und Ersatzschaltbild eines Kondensators, 2012
- [11] <http://www.xfig.org/userman> *Xfig documentation*,. Documentation of the interactive drawing tool xfig, 2009
- [12] <http://docs.gimp.org/2.6/de/gimp> *gimp documentation*. Documentation of the GNU Image Manipulation Program, 2010
- [13] <http://www.mikrocontroller.net/articles/AVR-TransistorTester> *Online Dokumentation des TransistorTester*, Online Article, 2009-2011
- [14] <http://www.mikrocontroller.net/articles/AVRDUDER> *Online Dokumentation des avrdude programmer interface*, Online Article, 2004-2011
- [15] <http://www.mikrocontroller.net/topic/131804> *Thread von Markus F.*, Forum thread, 2009
- [16] http://www.mikrocontroller.net/articles/AVR_TransistorTester *Kurze Eigenschaftenbeschreibung des TransistorTester von Karl-Heinz K.*, Online Article, 2012

- [17] <http://www.mikrocontroller.net/topic/248078> Thread von Karl-Heinz K., Thread und Software Versionen, 2012
- [18] <http://www.mikrocontroller.net/articles/WinAVR> Artikel über WinAVR, Online Artikel, 2012
- [19] <http://sourceforge.net/projects/winavr/files> Quelle für WinAVR, Download Quelle, 2012
- [20] <http://www.mikrocontroller.net/topic/248078?page=5#2922341> Patch für WinAVR, Setzen der Fuses mit avrdude, Download Quelle, 2012
- [21] <http://www.orientdisplay.com/pdf/ST7565.pdf> Datenblatt des grafischen Controllers ST7565, Download Quelle, 2014
- [22] Maxim Integrated Products, Inc. <http://maximintegrated.com> DS3231: Extremely Accurate I²C-Integrated RTC/TCXO/Crystal, Data Sheet, 19-5170;Rev 10; 3/15, 2015
- [23] Maxim Integrated Products, Inc. <http://maximintegrated.com> DS3231M: 5ppm I²C Real-Time Clock, Data Sheet, 19-5312;Rev 7; 3/15, 2015