

TransistorTester mit AVR Mikrocontroller
und ein wenig mehr
Version 0.99k

Karl-Heinz Kübbeler
kh_kuebbeler@web.de

14. August 2012

Inhaltsverzeichnis

1	Eigenschaften	3
2	Hardware	5
2.1	Die Schaltung des TransistorTesters	5
2.2	Hinweise für den Aufbau des TransistorTesters	6
2.3	Programmierung des Mikrocontrollers	7
2.4	Fehlersuche	7
3	Bedienungshinweise	9
4	Konfigurieren des TransistorTesters	11
5	Beschreibung des Meßverfahrens	16
5.1	Messung von Halbleitern	18
5.1.1	Messung eines PNP-Transistors oder eines P-Kanal MOSFETs	18
5.1.2	Messung eines NPN Transistors oder eines N-Kanal-MOSFET	20
5.1.3	Messung von Dioden	22
5.1.4	Ergebnisse der verschiedenen Messungen	22
5.2	Widerstands Messung	24
5.2.1	Widerstandsmessung mit den 680 Ohm Widerständen	24
5.2.2	Widerstandsmessung mit den 470 kOhm Widerständen	26
5.2.3	Ergebnisse der Widerstands-Messung	27
5.3	Messen von Kondensatoren	29
5.3.1	Entladen der Kondensatoren	29
5.3.2	Messung von großen Kapazitäten	29
5.3.3	Messen von kleinen Kapazitäten	31
5.3.4	Ergebnisse der Kondensator-Messung	33
5.3.5	Automatischer Abgleich der Kondensator-Messung	38
5.4	Selbsttest Funktion	43
6	Bekannte Fehler und ungelöste Probleme	48
7	Spezielle Software Teile	49
8	Arbeitsliste und neue Ideen	50

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. 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. [1] 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 Speicher 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 (flash) 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 [5] 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 LCD-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 8 gesammelt. Inzwischen kann ich unter dem Linux Betriebssystem auch den EEPROM-Speicher des ATmega8 einwandfrei beschreiben.

Kapitel 1

Eigenschaften

1. Arbeitet mit ATmega8, ATmega88, ATmega168 oder ATmega328 Mikrocontrollern.
2. Anzeige der Meßergebnisse auf einem 2x16 Zeichen LCD-Display.
3. Ein Tasten Bedienung mit automatischer Abschaltfunktion.
4. Batterie-Betrieb ist möglich, weil der Strom im abgeschalteten Zustand nur etwa 20nA beträgt.
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, Thyristors und Triacs.
7. Automatische Erkennung der Pin Anschlüsse 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-Spannungen und der Gate Kapazitätswerte von MOSFETs.
12. Bis zu zwei Widerstände werden gemessen und mit Symbolen  und den Widerstandswerten 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 0,1Ω, Werte von bis zu 50MΩ werden erkannt.
14. Ein Kondensator kann erkannt und gemessen werden. Der wird mit dem Symbol  und Kapazitätswert angezeigt. Der Wert kann zwischen 25pF (bei 8MHz Takt, 50pF bei 1MHz Takt) bis 40mF mit einer Auflösung von bis zu 1pF (bei 8MHz Takt).
15. Bis zu vier Dezimalstellen werden für die Kapazitäts und Widerstands-Werte in der richtigen Dimension angezeigt.
16. Bis zu zwei Dioden werden mit dem Symbol  oder dem Symbol  in der richtigen Reihenfolge angezeigt. Zusätzlich werden die Schwellspannungen angezeigt.

17. Eine LED wird als Diode erkannt, die Schwellspannung ist viel höher als bei einer normalen Diode. Doppeldioden werden als zwei Dioden erkannt.
18. 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!
19. 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 nach der anderen.
20. 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.
21. Die Anschlüsse einer Gleichrichter-Brücke können mit nur einer Messung herausgefunden werden.
22. 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.
23. Die Meßzeit beträgt ungefähr zwei Sekunden, nur Kapazitätsmessungen können länger dauern.
24. Die Software kann für Meßserien mit vorgebbarer Wiederhol-Zahl konfiguriert werden, bevor die automatische Abschaltung ausschaltet.
25. Eingebaute Selbsttest-Funktion inklusive einem optionalen 50Hz Frequenz-Generator um die Genauigkeit der Taktfrequenz und der Verzögerungszeiten zu überprüfen.
26. Wählbare Möglichkeit, den Nullabgleich für die Kondensatormessung und die Innenwiderstandsbestimmung für die Portausgänge beim Selbsttest abzugleichen.

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! Es ist möglich, daß nicht alle Optionen in zukünftigen Versionen erhalten bleiben, weil diese Software immer noch im Teststadium ist. Alle Möglichkeiten sind nur bei Mikrocontrollern mit mehr Programmspeicher wie ATmega168 konfigurierbar.

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.

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 [1] 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 bereitete 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 platziert werden. Weil der PD7 Eingang und der PC6 (RESET) Anschluß die einzigen Anschlüsse sind, wo „pull-up“ Widerstände gebraucht werden, wurde ein zusätzlicher $27k\Omega$ Widerstand an dem PD7 (Pin 12) Anschluß 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 Bereich für den ADC umschalten. Die Umschalt-Geschwindigkeit wird durch den externen Kondensator C1 am AREF (21) Pin 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 nach. Einige unterschiedliche Versionen von R11 / R12 Kombinationen zirkulieren im Internet. Ich habe die Software an den Original-Entwurf von Markus F. [1] mit $10k\Omega$ und $3.3k\Omega$ angepaßt. Die zusätzliche 2,5V Präzisions-Spannungsreferenz, die an Pin PC4 (ADC4) angeschlossen ist, ist für zukünftige Software-Versionen geplant und wird derzeit noch nicht benutzt. Ein zusätzlicher ISP-Anschluß wurde hinzugefügt um leichter neue Software-Versionen laden zu können.

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!

2.3 Programmierung des Mikrocontrollers

Ich gebe die Software für den Mikrocontroller in Quell-Text heraus. Die Entwicklung wurde mit dem Linux Betriebssystem (Ubuntu) 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 LiesMich.txt Datei im Verzeichnis Sourcecode/default und in das Konfigurations-Kapitel 4. Das Ergebnis der Übersetzung hat die Dateierweiterung .hex und .eep. Üblicherweise heißen die Dateien TransistorTester.hex und TransistorTester.eep. Die .hex Datei enthält die Daten für den Programm Speicher (flash) des ATmega Prozessors. Die .eep Datei enthält die Daten für den EEPROM Speicher 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 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-Variante, wenn Sie keinen 8MHz 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 Programmdateien benutzen, aber das kann man später korrigieren! Aber falsch gesetzte fuses können die spätere ISP-Programmierung verhindern.

2.4 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 Programmdateien für den flash Speicher 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 Stromverstärkungsfaktor 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. 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-Kontroller 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.

Einiges, aber nicht alles ist auf der LCD-Anzeige lesbar Überprüfen Sie ob die .eep Daten in den EEPROM Speicher des ATmega geladen wurden. Wenn alle Programmdateien richtig geladen wurden, sollten Sie die Taktrate ihrer Programmdateien (Makefile) und die ATmega Prozessor 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 Überprüfe 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 5.4. Anderenfalls prüfen sie Ihre Platine visuell und prüfen Sie die Widerstandswerte mit einem Ohmmeter. Sie können die Pinne des ATmega für diese Prüfung benutzen, zum Beispiel können Sie der 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.

Kapitel 3

Bedienungshinweise

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 0,3 V und maximal 1,3 V 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 länger dauern.

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

Einzelmessung Wenn der Tester für Einzelmessung konfiguriert ist, schaltet der Tester nach einer Anzeigezeit von 10 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.

Dauermessung Einen Sonderfall stellt die Konfiguration ohne die automatische Abschaltfunktion dar. 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. 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 erkanntem 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 drei Sekunden. Wenn der Starttaster bis zum Erscheinen der ersten Meldung festgehalten wurde, beträgt die Anzeigezeit wie bei der Einzelmessung zehn Sekunden. Ein vorzeitiger neuer Meßbeginn ist aber während der Anzeigezeit durch erneutes Drücken des Starttasters möglich.

Wenn die Software mit der Selbsttestfunktion konfiguriert ist, kann der Selbsttest durch einen Kuzschluß aller drei Testports gestartet werden. Hier werden die im Selbsttest-Kapitel 5.4 beschriebenen Tests ausgeführt. Die achtmalige Testwiederholung kann vermieden werden, wenn der Starttaster gedrückt gehalten wird. So kann man uninteressante Tests schnell beenden und sich durch Loslassen des Starttasters interessante Tests achtmal wiederholen lassen. Wenn die Funktion `AUTO_CAL` in der Makefile gewählt ist, wird beim Selbsttest eine Kalibration der Nullwertes für die Kondensatormessung und eine Kalibration des Innenwiderstandes der Port-Ausgänge durchgeführt. Hierfür müssen aber alle acht Testwiederholungen durchgeführt werden, um die Mittelwerte zu bilden.

Kapitel 4

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 einen Ubuntu Linux Betriebssystem 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 (Version 5.11svn) von der Makefile benutzt, wenn man „make upload“ aufruft. Das Programm avrdude ist für Linux und Windows Betriebssysteme verfügbar. Der GNU C-Kompiler wird auch von der AVR studio Software unter Windows benutzt. Sie können die Programm Daten (.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 Signatur Bytes 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 ATmega88. Die Software, die für einen ATmega168 übersetzt wurde, läuft nicht auf einem ATmega88, sogar wenn der ATmega88 genug Speicher hat! Sind Sie vorsichtig, wenn Sie nicht meine 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
m48 or m48p = ATmega48
m88 or m88p = ATmega88
m168 or m168p = ATmega168
m328 or m328p = ATmega328
Beispiel: PARTNO = m168

UI_LANGUAGE gibt die Sprache für den Tester an

LANG_ENGLISH, LANG_GERMAN, LANG_POLISH, LANG_CZECH, LANG_SLOVAK und LANG_SLOVENE sind derzeit verfügbar.

Beispiel: UI_LANGUAGE = LANG_ENGLISH

LCD_CYRILLIC wird nur gebraucht, wenn man ein LCD-Display mit kyrillischem Zeichensatz benutzt. Das μ und Ω Zeichen ist im kyrillischen Zeichensatz nicht enthalten. Wenn Sie diese Option angeben, werden beide Zeichen von der Software in das LCD geladen.

Beispiel: CFLAGS += -DLCD_CYRILLIC

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

AUTO_CAL Der Nullabgleich für die Kondensatormessung und der Innenwiderstand der Port-Ausgänge wird beim Selbsttest zusätzlich ins EEprom geschrieben und ist damit für die weiteren Messungen abgeglichen.
Beispiel: CFLAGS += -DAUTO_CAL

FREQUENCY_50HZ Zum Ende des Selbsttests wird bis zu einer Minute lang ein 50 Hz Signal auf Port 2 und Port 3 erzeugt.
Beispiel: CFLAGS += -DFREQUENCY_50HZ

R_MESS schaltet die Widerstandsmessung ein. Diese Option sollte immer angegeben werden.
Beispiel: CFLAGS += -DR_MESS

C_MESS schaltet die Kondensatormessung ein. Diese Option sollte immer angegeben werden.

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 kleinen Kapazitäten (unter $50\mu F$) 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. Ein Wert von 10 ergibt etwa 1 Prozent kleinere Meßergebnisse.
Beispiel: CFLAGS += -DREF_C_KORR=14

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

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.
Beispiel: CFLAGS += -DAUTOSCALE_ADC

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

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.
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 8MHz Betrieb wird wegen der besseren Auflösung der Kondensator-Messung empfohlen.

Beispiel: OP_MHZ = 8

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

Beispiel: CFLAGS += -DUSE_EEPROM

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. Für den 5V Betrieb sind die Werte 44, 22 und 11 möglich. Die 44 ergibt die beste Genauigkeit aber die längste Meßzeit. Eine ADC-Messung mit dem Wert 44 braucht etwa 4,7ms.

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 installierten Schalttransistoren die Option POWER_OFF weglassen haben, können Sie die Mess-Wiederholung dennoch abbrechen, wenn Sie den Start-Knopf einige Sekunden gedrückt halten während das Meßergebnis angezeigt wird bis die „Timeout“ Meldung erscheint. Wenn Sie den Knopf loslassen, schaltet der Tester den Strom ab. Sie können 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 10 Sekunden angezeigt, bei der Mehrfachmessung wird die Anzeigezeit auf 3 Sekunden reduziert (wird in config.h gesetzt). Wenn der Startknopf beim ersten Einschalten lange gedrückt wird, wird das Meßergebnis auch bei der Mehrfachmessung 10 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 Versions-Nummer der Software angezeigt. Diese Option ist hilfreich um bei Batterie betriebenen Tester Versionen an den Batterie Wechsel 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 wegen des Verpolungs-Schutzes 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 100mV (0,1V) Wert. Die Warn-Spannung ist immer 1V höher 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 wegen der Tief-Entladung 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=54`

Beispiel für 7805 type Regler (6.4V): `CFLAGS += -DBAT_POOR=64`

UF_OUT_MV setzt das Ausgabeformat für die Diodenflußspannung auf mV Einheit. Anderenfalls wird die Flußspannung in Volt mit zwei Nachkommastellen ausgegeben. Die Ausgabe in Volt spart eine Stelle auf dem LCD, die mV Ausgabe braucht weniger Programmspeicher. Die 16 Stellen des LCD sind manchmal zu knapp für die zusätzliche Angabe der Kapazität, deshalb sollte die Volt Ausgabe bevorzugt werden.

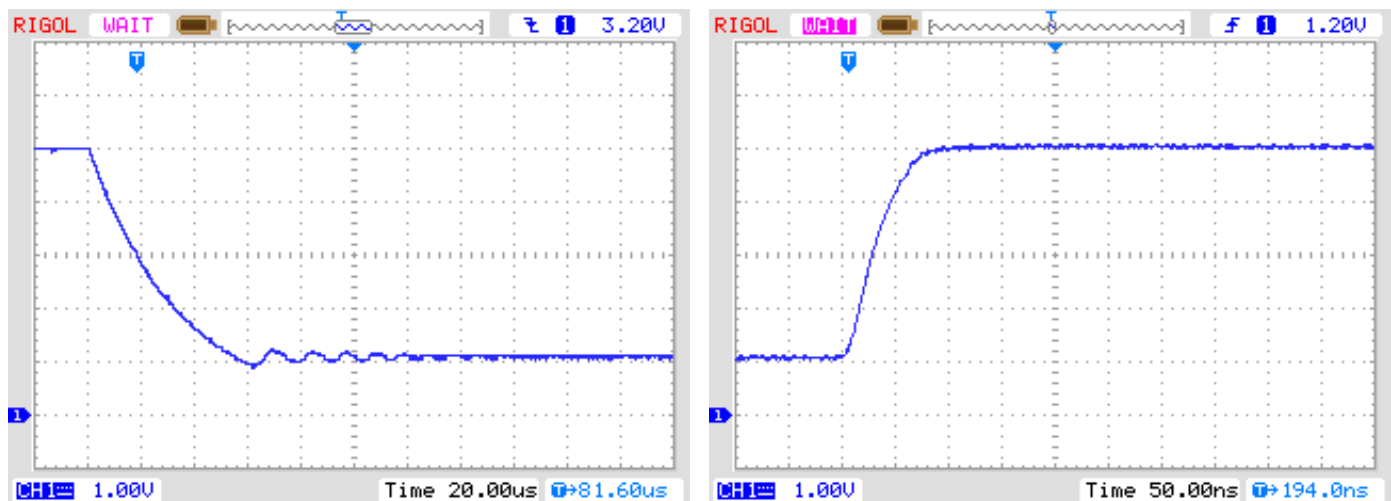
Beispiel: `CFLAGS += -DUF_OUT_MV`

PROGRAMMER stellt den Programmer Typ für das avrdude Schnittstellenprogramm ein. Eine richtige Einstellung des Programmer Typs (und Ports) ist notwendig, wenn Sie den „make upload“ oder „make fuses“ Aufruf dieser Makefile benutzen. Für weitere Informationen schauen Sie bitte in die Handbuch Seiten von avrdude oder in die online Dokumentation [11].

Beispiel: `PROGRAMMER=avrisp2`

PORT stellt die verwendete Schnittstelle ein, wo avrdude den Mikrocontroller (ATmega) erreichen kann. Für weitere Informationen schauen Sie bitte in die Handbuch Seiten von avrdude.

Beispiel: `PORT=usb`



(a) from 5V to 1.1V

(b) from 1.1V to 5V

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

Zusätzliche Parameter können in den Dateien `Transistortester.h` und `config.h` gesetzt werden. Die Datei `Transistortester.h` enthält globale Variablen und definiert die Port / Pin Konstellation sowie

die Widerstandswerte, die für die Messung benutzt werden. Die Datei config.h setzt Parameter für die verschiedenen Prozessor Typen, Wartezeiten und die Taktfrequenz für den ADC. Normalerweise brauchen diese Werte nicht ohne Grund geändert werden.

Kapitel 5

Beschreibung des Meßverfahrens

Ein vereinfachtes Schaltbild eines Eingang/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 Ausgabe Wert (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.

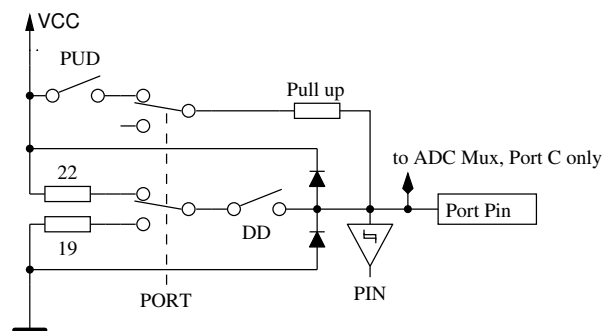


Abbildung 5.1: vereinfachtes Schaltbild jedes ATmega Port Pins

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

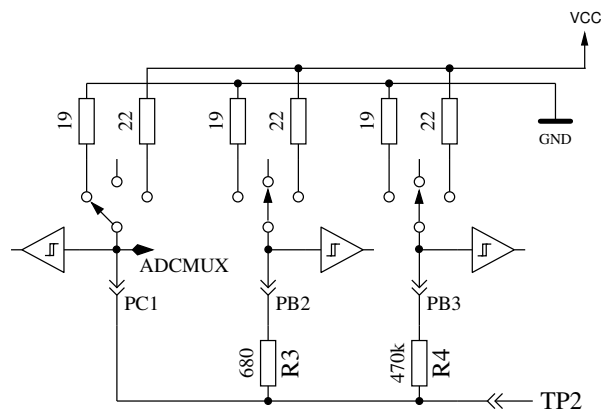


Abbildung 5.2: vereinfachtes Schaltbild des Testpins TP2

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 einen 680Ω Widerstand oder einen $470k\Omega$ Widerstand mit entweder GND oder VCC verbunden werden. Tabelle 5.1 zeigt alle denkbaren Meßmöglichkeiten. Beachte, 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

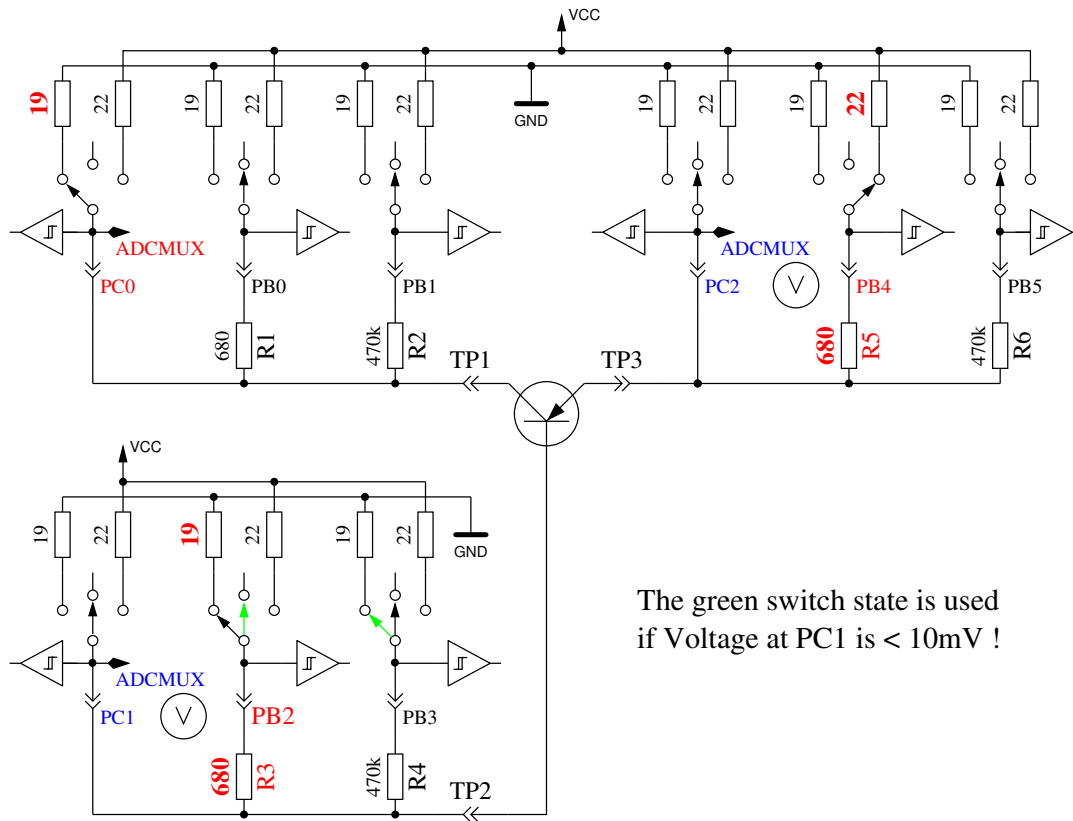
Wenn die Kondensatormessung des Testers konfiguriert ist, versucht der Tester vor allen Messungen erst einmal, die Kondensatoren an allen Anschlußpins zu entladen. Wenn das nicht gelingt, also die Restspannung zu hoch bleibt, wird das Entladen nach etwa 12 Sekunden mit der Meldung „Cell!“ abgebrochen. 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

Ein Testpin wird als negative Seite des Bauteils angenommen. Ein anderer Pin wird als positive Seite des Bauteils angenommen. Als erster Test wird die positive Seite des Bauteils direkt mit VCC verbunden. Die negative Seite wird mit dem 680Ω Widerstand nach GND verbunden. Der Testpin (dritter Pin, auch TriStatePin genannt) wird zuerst mit dem 680Ω Widerstand für 10ms mit GND verbunden. Die Spannung des negativen Testpins wird gemessen, während der TriStatePin auf Eingang geschaltet ist. Es wird angenommen, daß das getestete Bauteil ein P-Kanal MOSFET sein kann und daß das Gate entladen sein sollte. Wenn die gemessene Spannung über 976mV ist, nimmt der nächste Test an, daß das getestete Bauteil auch ein P-Kanal MOSFET sein könnte und dafür wird der 680Ω Widerstand für 10ms zur VCC Seite geschaltet. Auch für diesen Fall wird die Spannung des negativen Pins mit stromlosen TriStatePin gemessen. Wenn die Spannung des negativen Pins größer als 92mV ist, werden zusätzliche Tests gemacht um N-Kanal JFET oder D-MOSFET (Verarmungs Typ) und P-Kanal JFET oder P-MOSFET zu unterscheiden. Die MOSFET Versionen können unterschieden werden durch den stromlosen Zustand des TriStatePins in jedem Zustand. Wenn das Bauteil keinen Strom zwischen dem positiven Pin und dem negativen Pin ohne ein Signal auf dem TriStatePin hat, sind die nächsten Tests im nächsten Unterkapitel 5.1.1 beschrieben. Wenn Strom festgestellt wird, sind die nächsten Tests in dem Dioden Unterkapitel 5.1.3 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 5.3 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 UB Spannung 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. Weil der Stromverstärkungsfaktor für Darlington-Transistoren sehr hoch werden kann, wird der Faktor auf 65535 ($2^{16} - 1$) limitiert.



The green switch state is used if Voltage at PC1 is < 10mV !

Abbildung 5.3: h_{FE} 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 5.4 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 die Basis-Spannung größer als 0,97V ist, muß es ein PNP sein. Für die Messung des Stromverstärkungsfaktors wird anstelle des 680Ω Widerstandes der 470kΩ Widerstand als Basis-Widerstand genommen. Der Stromverstärkungsfaktor wird berechnet mit $h_{FE} = \frac{U_C \cdot 470000}{U_B \cdot (680 + 19)}$. 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 470kΩ Widerstand rauf und runter gezogen wird bis die Drain Seite schaltet und dann die Spannung am Gate gemessen wird.

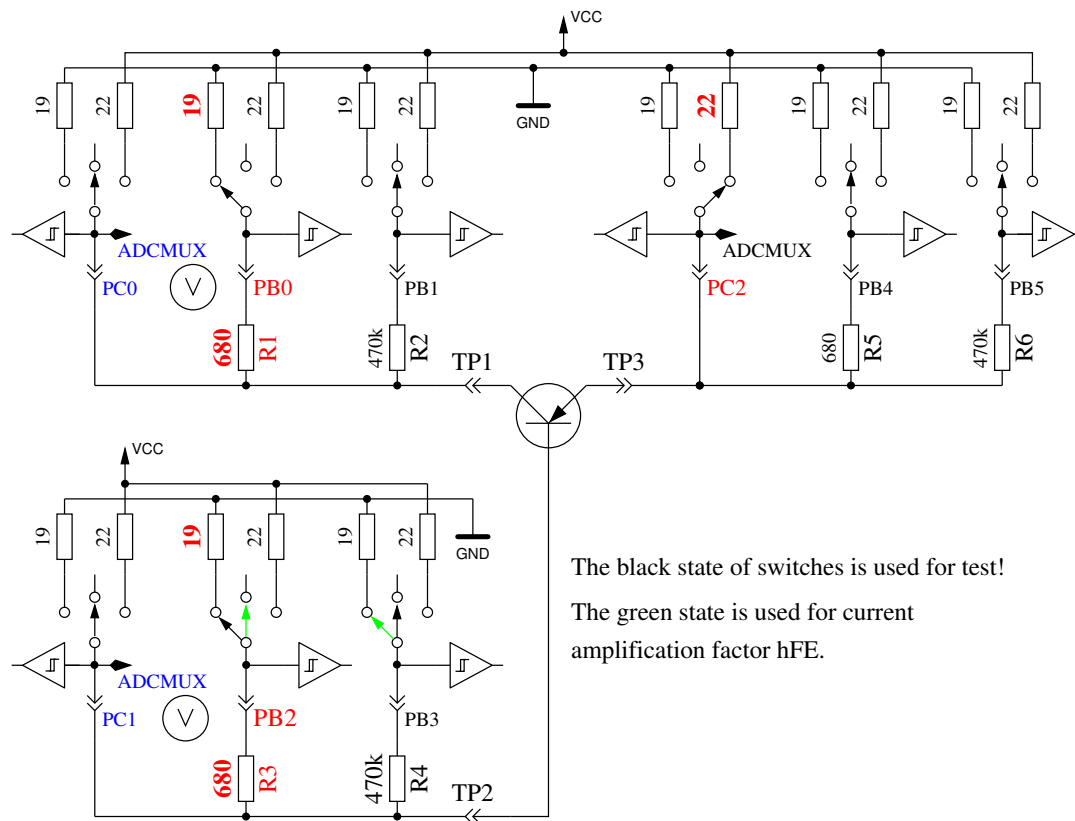


Abbildung 5.4: Prüfung und h_{FE} Messung eines PNP Transistors in der Kollektor-Schaltung

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 mit der Messung des Stromverstärkungsfaktors in der Kollektor-Schaltung. Zuerst wird die Messung mit einem nach VCC geschalteten 680Ω Basis Widerstand gemacht. Wenn die Spannung an dem Basis-Widerstand zu klein ist, wird stattdessen der $470k\Omega$ Widerstand genommen. Der Stromverstärkungsfaktor wird auf 65535 (16Bit) begrenzt. Die Messungen werden dann in der Emitter-Schaltung fortgeführt, wie in Abbildung 5.5 gezeigt.

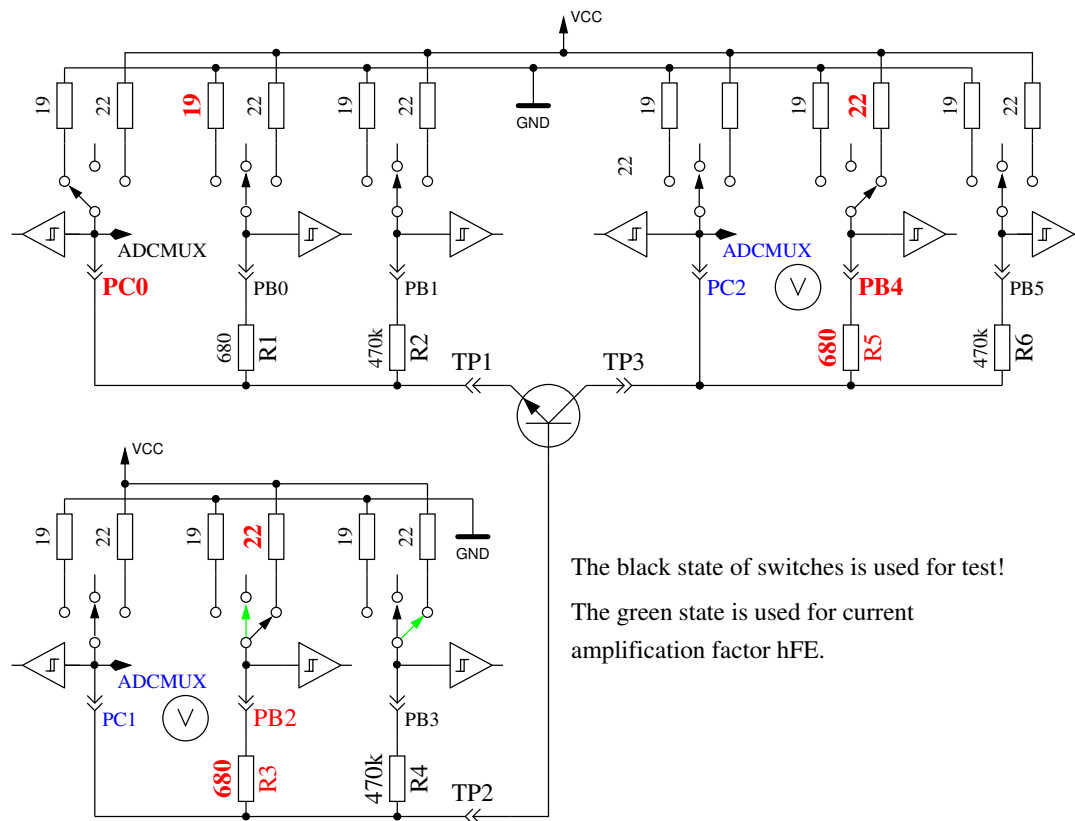


Abbildung 5.5: Prüfung und hFE Messung eines NPN Transistors in Emitter-Schaltung

Wenn die Spannung auf der Kollektor Seite unter 1,6V liegt, wenn der 680Ω Basis Widerstand 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 $hFE = \frac{(VCC-UC) \cdot 470000}{(VCC-UB) \cdot (680+22)}$ gebildet. Wenn die Spannung an der Basis zeigt, daß da 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 Eingangssignal auf der Drain Seite schaltet, wobei dann die Gate Spannung gelesen wird. Die Messung wird elf Mal wiederholt wie in Abbildung 5.6 gezeigt und die Ergebnisse addiert. Diese Summe wird mit vier multipliziert und durch neun geteilt um eine Auflösung in mV zu erhalten.

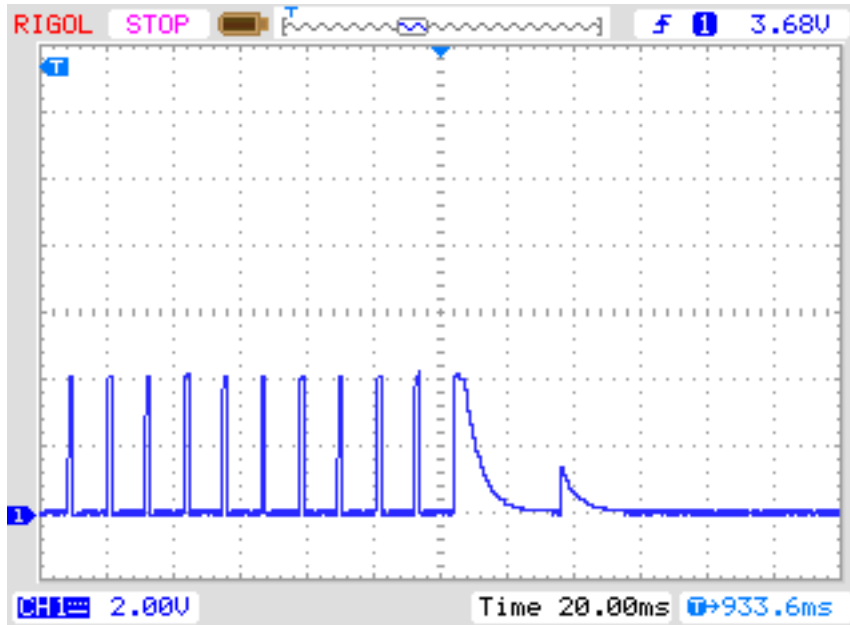


Abbildung 5.6: Messung der Schwellspannung eines N-Kanal-MOSFET

5.1.3 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 acht Mal die Flußspannung mit dem $470k\Omega$ Widerstand muß größer als die Flußspannung mit dem 680Ω Widerstand sein. Ich hoffe, daß dieses Verhalten immer eine Diode ist.

5.1.4 Ergebnisse der verschiedenen Messungen

Die folgenden drei Tabelle zeigen die Ergebnisse verschiedener Bauteile mit verschiedenen Software Konfigurationen von ATmega8 und ATmega168 Prozessoren.

Diode Type	Mega8@8MHz Signatur 1E9307 WITH_AUTO_REF	Mega168 @8MHz Signatur 1E9406	Mega168 @8MHz Signatur 1E9406 WITH_AUTO_REF AUTOSCALE_ADC
1N4148	Diode, .71V, 1pF	Diode, .71V, 0pF	Diode, .72V, 1pF
1N4150	Diode, .67V, 2pF	Diode, .67V, 0pF	Diode, .67V, 2pF
BA157	Diode, .62V, 19pF	Diode, .62V, 19pF	Diode, .62V, 19pF
BY398	Diode, .54V, 17pF	Diode, .54V, 16pF	Diode, .54V, 16pF
1N4007	Diode, .65V, 13pF	Diode, .65V, 12pF	Diode, .66V, 12pF
LED green	Diode, 1.95V, 6pF	Diode, 1.95V, 5pF	Diode, 1.96V, 5pF
ZPD2,7	2xDi, .74V, 2.53V	2xDi, .74V, 2.53V	2xDi, .74V, 2.52V
BU508A B+E	Diode, .61V, 5190pF	Diode, .61V, 5374pF	Diode, .61V, 5232pF
BU508A B+C	Diode, .58V, 257pF	Diode, .59V, 267pF	Diode, .59V, 260pF

Tabelle 5.2: Meßergebnisse der Dioden-Tests

Transistor Type	Mega8@8MHz Signatur 1E9307 WITH_AUTO_REF	Mega168 @8MHz Signatur 1E9406	Mega168 @8MHz Signatur 1E9406 WITH_AUTO_REF AUTOSCALE_ADC
BU508A	NPN, B=10, .60V	NPN, B=10, .61V	NPN, B=10, .61V
2N3055	NPN, B=21, .62V	NPN, B=22, .61V	NPN, B=22, .61V
BC546B	NPN, B=385, .77V	NPN, B=386, .77V	NPN, B=384, .78V
BC556B	PNP, B=306, .79V	PNP, B=429, .79V	PNP, B=262, .79V
BC639	NPN, B=180, .72V	NPN, B=228, .72V	NPN, B=228, .72V
BC640	PNP, B=216, .72V	PNP, B=240, .72V	PNP, B=185, .72V
AC128 (Ge.)	PNP, B=64, .27V	PNP, B=65, .27V	PNP, B=58, .27V
BC517	NPN, B=28296, 1.40V	NPN, B=29330, 1.41V	NPN, B=28224, 1.40V
BC516	PNP, B=65535, 1.41V	PNP, B=65535, 1.41V	PNP, B=65535, 1.40V
BRY55/200	Thyristor	Thyristor	Thyristor

Tabelle 5.3: Meßergebnisse der Tests mit bipolaren Transistoren

FET Type	Mega8@8MHz Signatur 1E9307 WITH_AUTO_REF	Mega168 @8MHz Signatur 1E9406	Mega168 @8MHz Signatur 1E9406 WITH_AUTO_REF AUTOSCALE_ADC
BS170	N-E-MOS,D, 2.62V 71pF	N-E-MOS,D, 2.55V 73pF	N-E-MOS,D, 2.55V 72pF
J310	N-JFET	N-JFET	N-JFET
IRFU120N	N-E-MOS,D, 4.14V 916pF	N-E-MOS,D, 4.14V 943pF	N-E-MOS,D, 4.14V 918pF
IRFU9024	P-E-MOS,D, 3.51V 956pF	P-E-MOS,D, 3.52V 977pF	P-E-MOS,D, 3.52V 942pF
ZVP2106A	P-E-MOS,D, 3.20V 118pF	P-E-MOS,D, 3.22V 122pF	P-E-MOS,D, 3.21V 118pF
ZVNL120A	N-E-MOS,D, 1.56V 142pF	N-E-MOS,D, 1.52V 147pF	N-E-MOS,D, 1.51V 143pF

Tabelle 5.4: Meßergebnisse der MOS Transistor 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 680 Ω Präzisionswiderständen durchgeführt. Das Schaltbild dieser Messungen mit Testpin 1 (TP1) und Testpin 3 (TP3) werden vereinfacht in Abbildung 5.7 und Abbildung 5.8 als ein Beispiel von den sechs Kombinationsmöglichkeiten gezeigt.

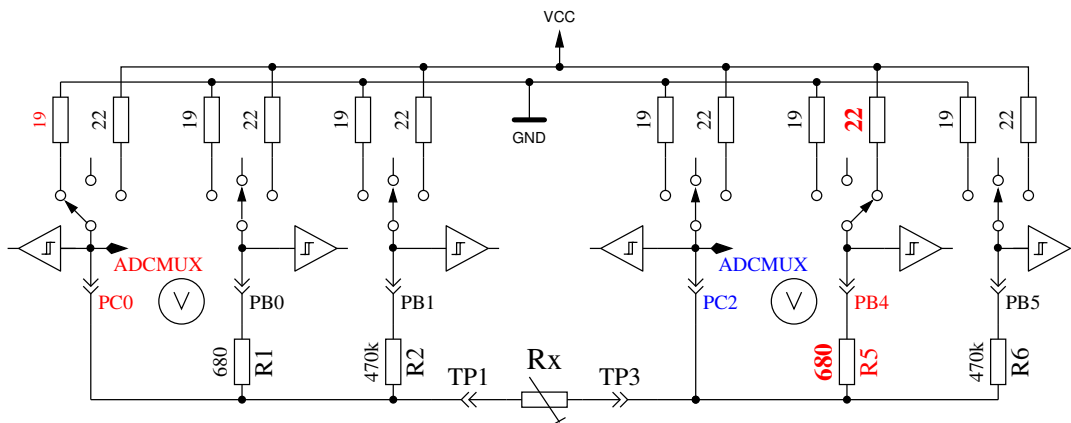


Abbildung 5.7: Messung Type 1 mit 680 Ω

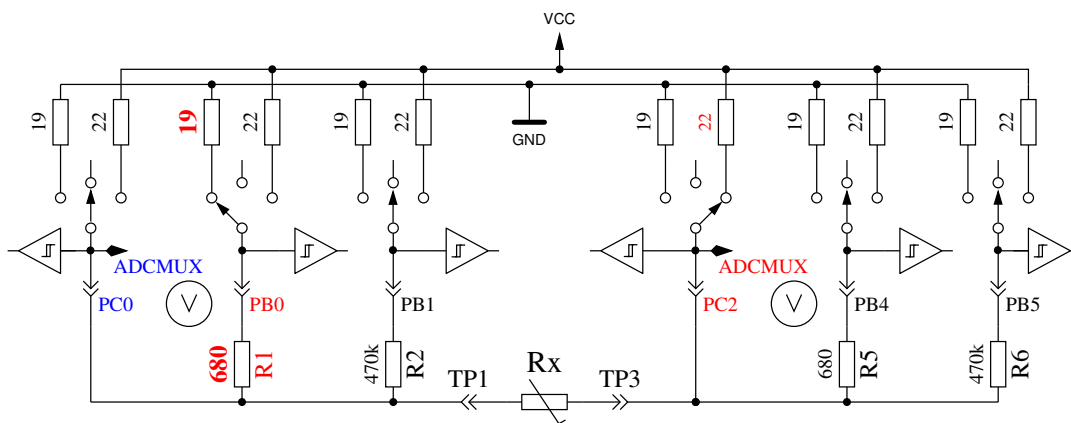


Abbildung 5.8: Messung Type 2 mit 680 Ω

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 5.9 gezeigt, wobei die Widerstandswerte in logarithmischer Skalierung dargestellt sind.

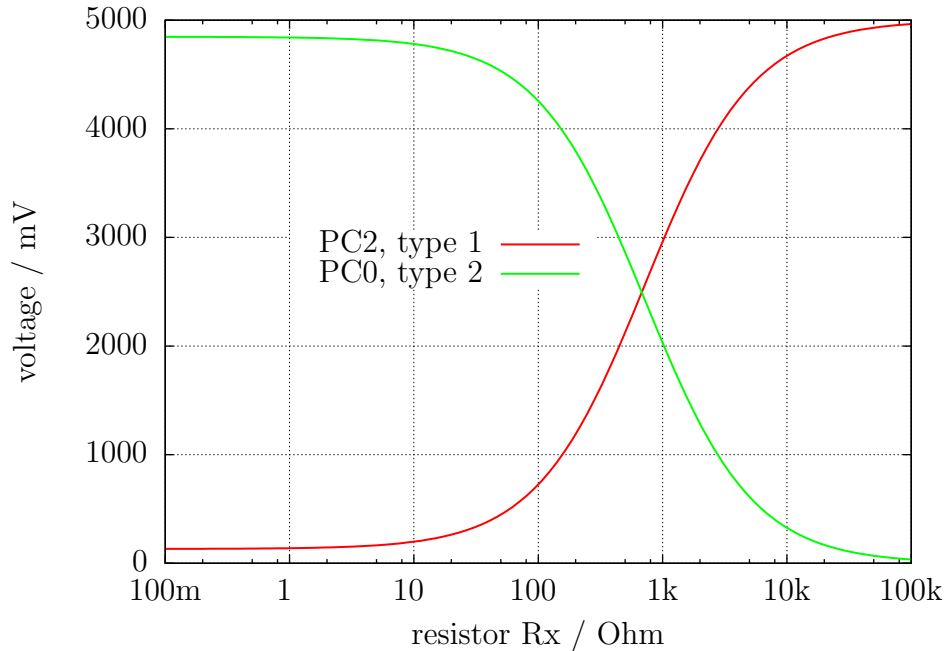
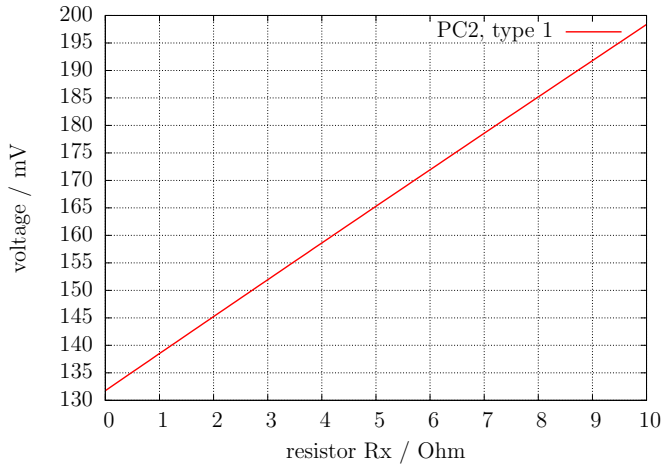


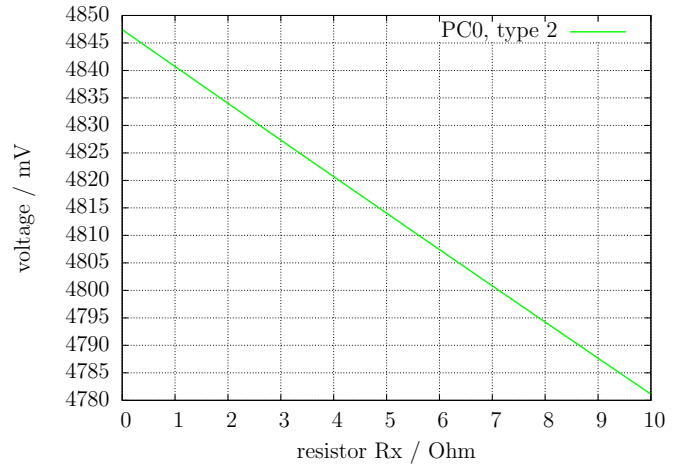
Abbildung 5.9: Spannung von Type 1 und Type 2 Messungen mit 680Ω

Die Verlauf für Messung Type 1 wird in Abbildung 5.10a mit gespreizter Darstellung für die unteren Widerstandswerte gezeigt. Wie man sehen kann, braucht man eine bessere ADC Auflösung als die Standard $4,9\text{mV}$ 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 5.10b gezeigt. Unglücklicherweise kann man nicht die höhere ADC Auflösung für Meßmethode Type 2 benutzen, weil die Spannung zu hoch ist und unser ATmega keine differenziellen ADC Eingänge besitzt. Die Messungen mit den 680Ω Widerständen werden bis zu einem Widerstandswert von $20\text{k}\Omega$ (Spannung ist unter 169mV) zur Bestimmung des Meßergebnisses verwendet.

Für höhere Widerstandswerte werden Messungen mit den $470\text{k}\Omega$ Widerständen benutzt. Der Mittelwert von beiden Messungen wird für den angezeigten Widerstandwert 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 $0,98\text{V}$ liegt, wird ein gewichteter Mittelwert mit Faktor vier für die Messung mit der Spannung unter $0,98\text{V}$ 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 ATmega88, ATmega168 und ATmega328 Prozessoren verwendet, für ATmega8 wird ein Faktor zwei als Wichtung benutzt wenn die Spannung unter $0,98\text{V}$ ist, weil die ADC Referenzspannung hier $2,54\text{V}$ statt $1,1\text{V}$ beträgt.



(a) Type 1 Messung



(b) Type 2 Messung

Abbildung 5.10: Ausschnitt des theoretischen Spannungsverlauf von 0Ω bis 10Ω

5.2.2 Widerstandsmessung mit den $470\text{ k}\Omega$ Widerständen

Die nächsten Abbildungen 5.11 und 5.12 zeigen die gleichen Meßmethoden für die Messungen mit $470\text{ k}\Omega$ Präzisionswiderständen. Weil $470\text{ k}\Omega$ sehr groß ist in Relation zu den Port Widerständen 22Ω und 19Ω , können die Port-Widerstände für die Berechnung des Widerstandswertes R_x vernachlässigt werden.

Für beide Meßmethoden mit den $470\text{ k}\Omega$ 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 Spannungs-Verlauf wird in Abbildung 5.13 gezeigt, wobei die Widerstandswerte wieder in logarithmischer Skalierung gezeigt werden. Der theoretische Verlauf in diesem Diagramm endet bei $100\text{ M}\Omega$, aber das Ergebnis des Testers wird auf $60\text{ M}\Omega$ begrenzt, anderenfalls nimmt der Tester an, daß kein Widerstand angeschlossen ist. Als Ergebnis wird der gewichtete Mittelwert von beiden Meßmethoden verwendet 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 700Ω addiert wird. Dieser Offset kann mit der `RH_OFFSET` Konstante (define) in der Datei `config.h` angepaßt werden.

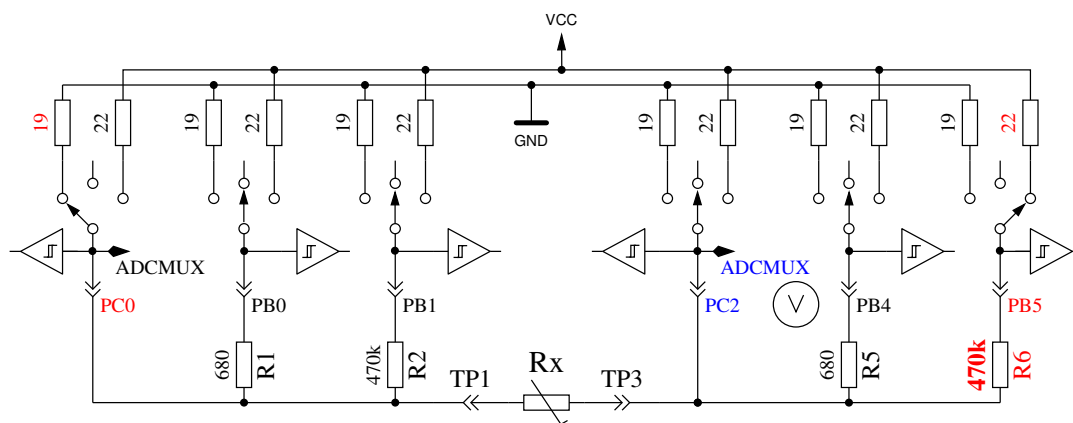


Abbildung 5.11: Messung Type 3 mit $470\text{ k}\Omega$

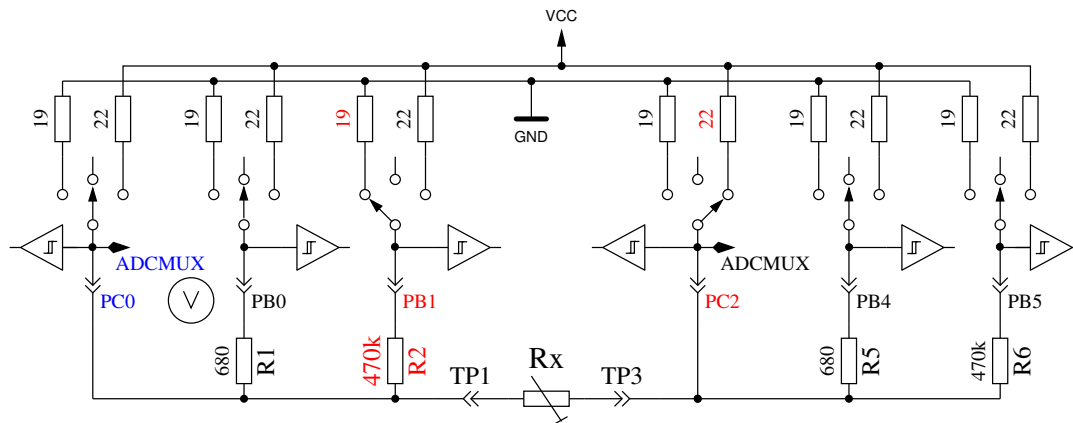


Abbildung 5.12: Messung Type 4 mit $470k\Omega$

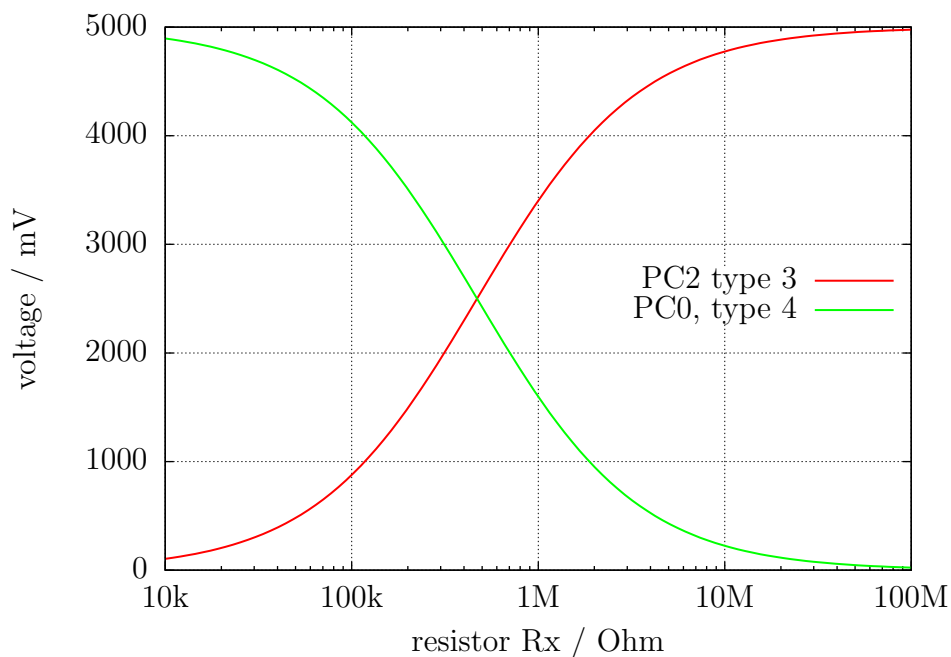


Abbildung 5.13: Spannungen von Type 3 und Type 4 Messungen mit $470k\Omega$

5.2.3 Ergebnisse der Widerstands-Messung

Abbildung 5.14 zeigt den relativen Fehler der Widerstandsmessung mit einem ATmega8 Mikrocontroller. Mega8 sind die Ergebnisse ohne AUTOSCALE_ADC Option, Mega8as sind die Ergebnisse mit der Option AUTOSCALE_ADC und Mega8orig sind die gleichen Messungen mit der original Software von Markus F. ohne meine Verbesserungen. Die Ergebnisse für die AUTOSCALE_ADC Version zeigen hier kaum Verbesserungs-Potential. Abbildung 5.15 zeigt die gleichen Messungen mit einem ATmega168 Mikrocontroller. Mega168 sind die Ergebnisse ohne die AUTOSCALE_ADC Option, Mega168as die mit der AUTOSCALE_ADC Option.

Mit dem ATmega168 Mikrocontroller 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 Widerstandwert hat. Es ist besser, den Widerstand direkt mit den Anschlußklemmen zu verbinden. Wenn das nicht möglich ist, sollte man der Widerstandswert der kurzgeschlossenen Prüfklemmen vom Meßergebnis abziehen. Zum Beispiel, wenn der Widerstand einen aufgedruckten Wert von 30Ω

hat, der Tester zeigt einen Wert von $30,6\Omega$ an und die beiden kurzgeschlossenen 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%!

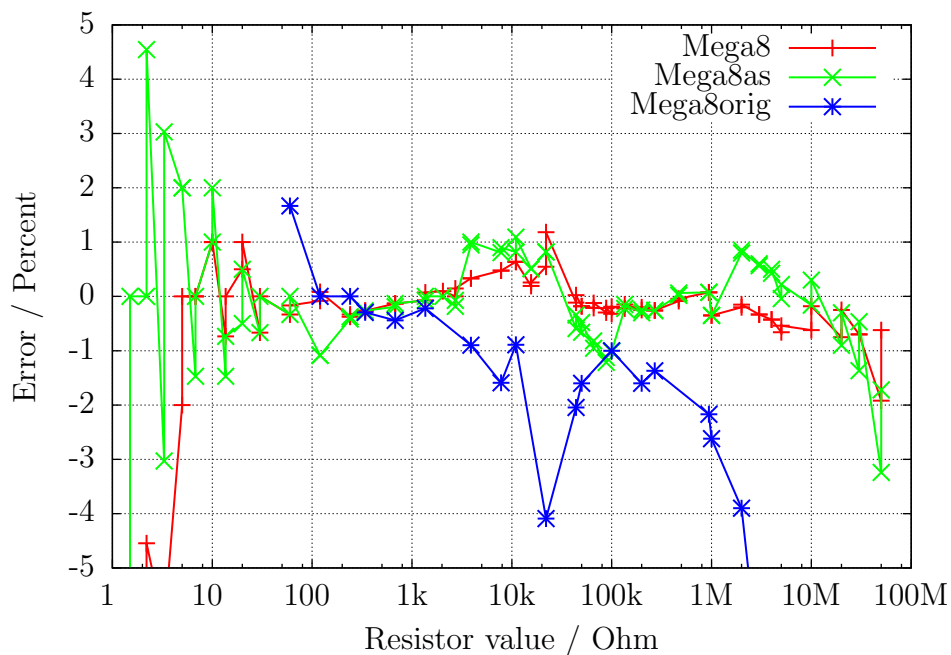


Abbildung 5.14: Relativer Fehler in % für Widerstands-Messungen mit ATmega8

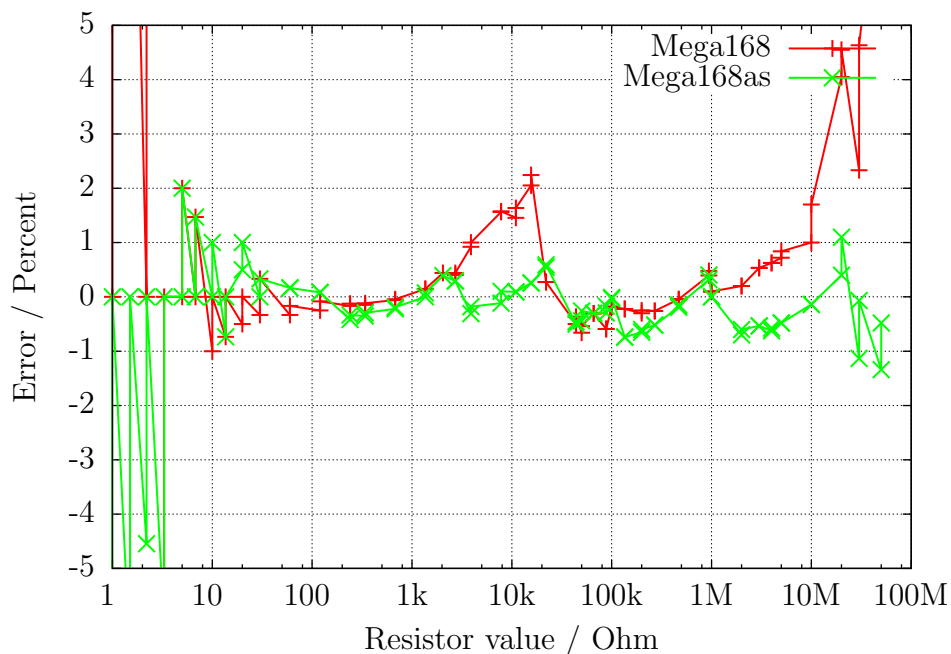


Abbildung 5.15: Relativer Fehler in % für Widerstand-Messungen mit ATmega168

5.3 Messen von Kondensatoren

Die Messung von Kapazitätswerten wird nach allen anderen Messungen als separater Teil mit einer Ladezeitmessung durchgeführt. Die original Software vom Markus F. hat das mit einer Programmschleife 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 zu erhalten in nur drei Kombinationsmöglichkeiten für die drei Testpins. 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 legal, weil jeder Portausgang einen Innenwiderstand von ungefähr 20Ω hat. Die Abbildung 149 (Seite 258) im Atmega8 Datenblatt [2] 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 15mF Kondensatoren 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 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 der 10 ms Ladepuls bis zu 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 kann. Abbildung 5.16 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.

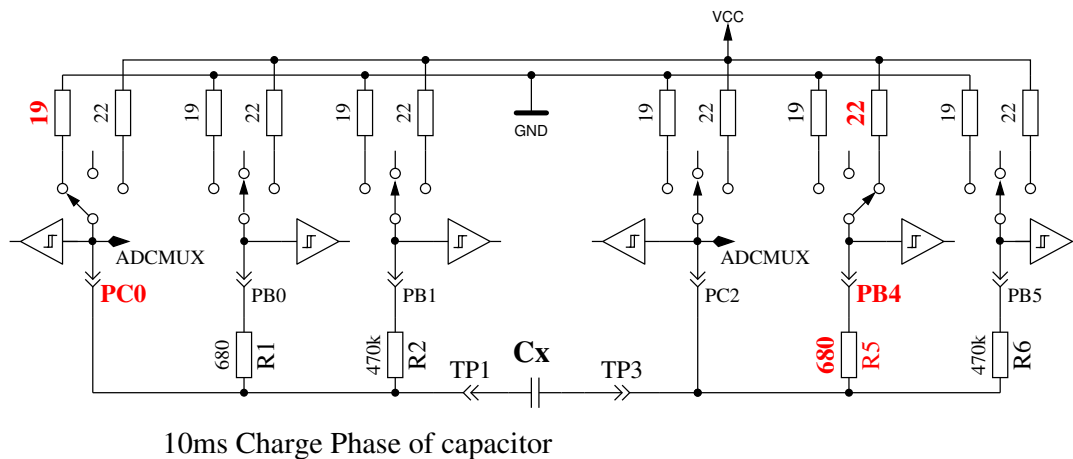
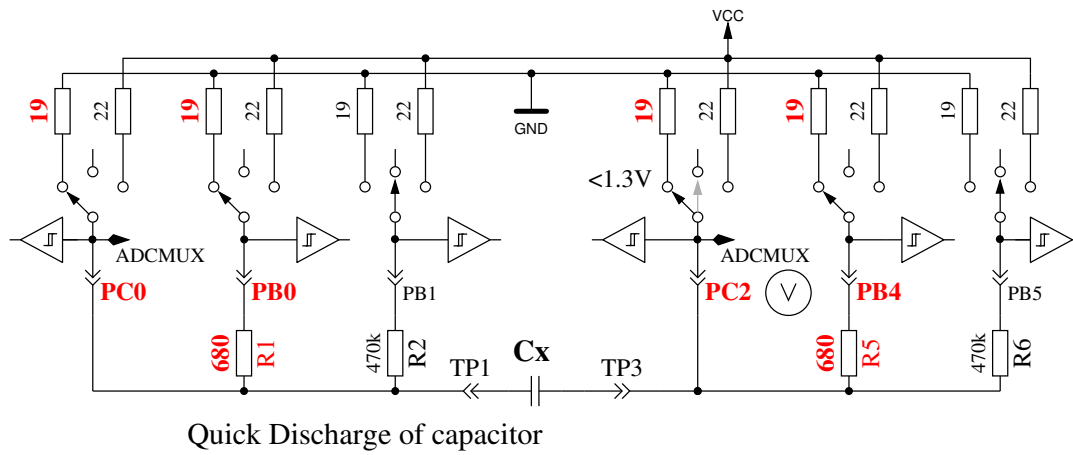


Abbildung 5.16: Kondensator Entladung und Ladung mit 10ms Ladepulsen bis Spannung $>300\text{mV}$

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. Abbildung 5.17a zeigt das Laden und Entladen eines $229\mu\text{F}$ Kondensators. Das flache Dach der Meßkurve bis zum Entladebeginn ist durch die Meßzeit und Berechnungszeit des ATmega verursacht. Abbildung 5.17b 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 5.18 zeigt die Messung eines 15mF Kondensators.



(a) $229\mu F$ Kondensator



(b) $5mF$ Kondensator

Abbildung 5.17: Laden und Entladen von großen Kondensatoren für die Messung



Abbildung 5.18: Laden und Entladen eines $15mF$ Kondensators für die Messung

5.3.3 Messen von kleinen Kapazitäten

Wenn der erste 10 ms Ladepuls den Kondensator zu hoch aufgeladen hat, wird eine andere Meßtechnik benutzt. Der ATmega Prozessor 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. Schaltbild 5.19 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 hat einen Abstand von 50mV Schritten und wird interpoliert entsprechend der aktuellen Referenz-Spannung. Diese Tabelle wird nur mit der Makefile Option `WITH_AUTO_REF` aktiviert. Vom Kapazitätswert ziehe ich eine experimentell herausgefundene vordefinierte Konstante oder eine im Selbsttest herausgefundene 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 immer etwas zu klein gemessen wird, deshalb kann man einen Zusatz mit der Makefile Option `REF_C_KORR` angeben. Die gemessene Referenz-Spannung wird dann mit dem angegebenen Korrekturwert (mV) korrigiert (addiert). Wenn die Option `WITH_AUTO_REF` nicht benutzt wird, werden die Referenz-Spannungen entsprechend den Angaben in den Datenblättern [2] [3] der ATmega8, ATmega88, ATmega168 und ATmega328 Prozessoren berücksichtigt. Eine Beispielmessung von diesem Typ ist in Abbildung 5.20 dargestellt. Die Meßzeit für den $22\mu F$ Kondensator ist 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.

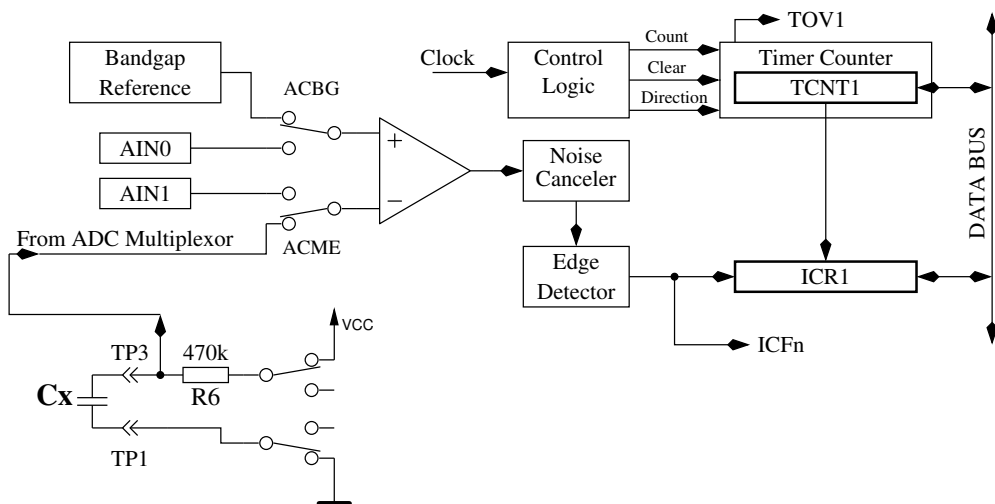


Abbildung 5.19: Messung kleiner Kapazitätswerte mit dem Komparator

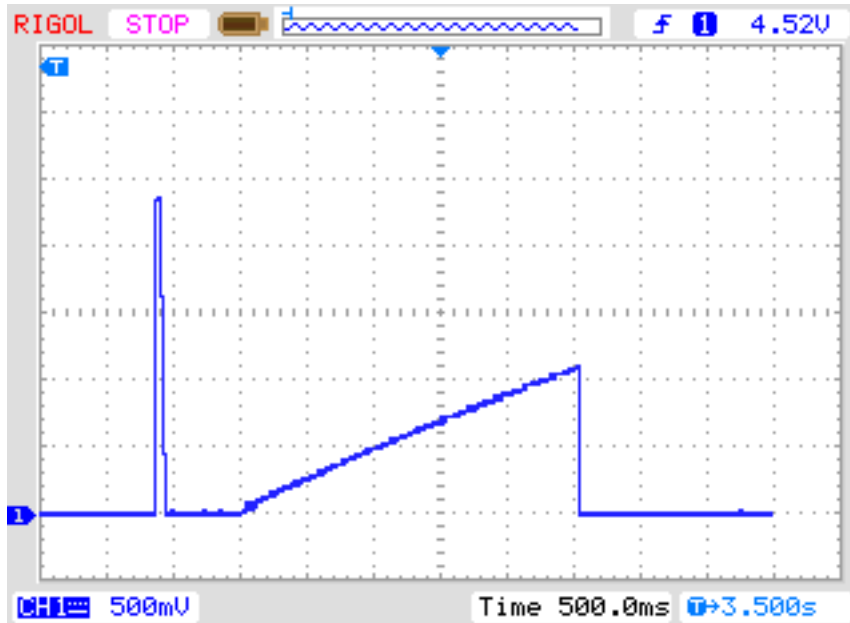


Abbildung 5.20: Laden und Entladen eines $22\mu F$ Kondensators für die Messung

Im Prinzip kann diese Technik auch mit dem 680Ω Widerstand benutzt werden, aber weil während dem Komparatorbetrieb der ADC nicht benutzt werden kann, habe ich 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. Die Methode, die in der aktuellen Software für große Kondensatoren in Kapitel 5.3.2 verwendet wird, vermeidet diesen Konzeptfehler.

5.3.4 Ergebnisse der Kondensator-Messung

Die Ergebnisse meiner Messungen werden in Abbildung 5.21 für den ATmega8 Prozessor ohne und mit der AUTOSCALE_ADC Option dargestellt. Zusätzlich sind einige Werte der Original-Software mit einem Korrekturfaktor von 0,88 (-12%) dargestellt. Die Ergebnisse der Messung der gleichen Kondensatoren mit einem ATmega168 werden in Abbildung 5.22 gezeigt. Die Referenz für die Fehlerberechnung sind die Meßwerte eines PeakTech 2170 RCL-Meßgerätes, nicht die aufgedruckten Werte der Bauteile. Die größeren relativen Meßfehler bei großen Kondensatoren liegen zum Teil an der zu hohen Meßfrequenz (100 Hz) des RCL-Meßgerätes für große Elektrolytkondensatoren, zum anderen spielt auch die schlechte Güte der Elektrolytkondensatoren eine Rolle.

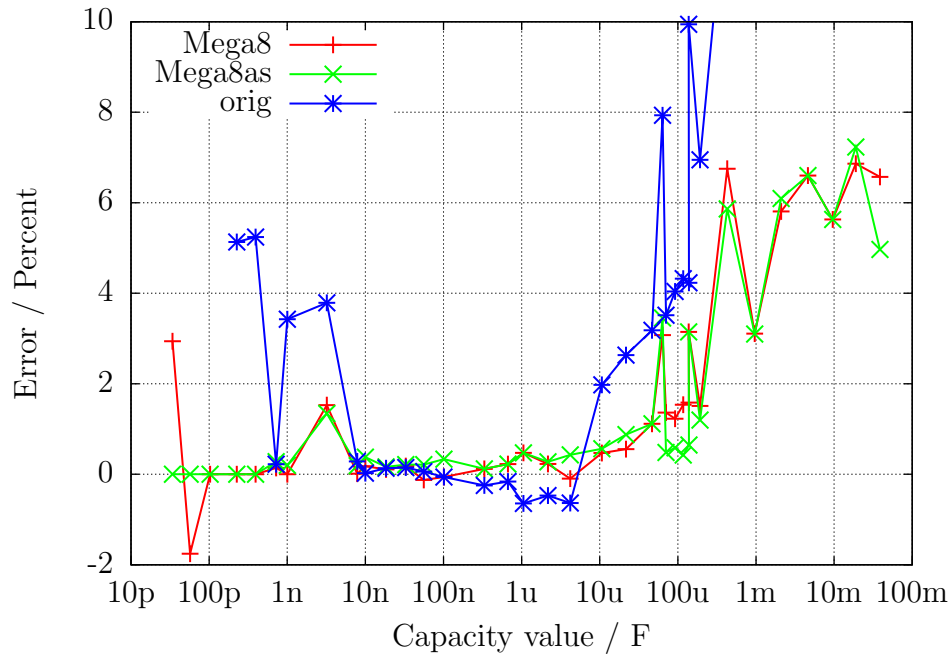


Abbildung 5.21: Prozentualer Fehler der Kondensator-Messungen mit dem ATmega8

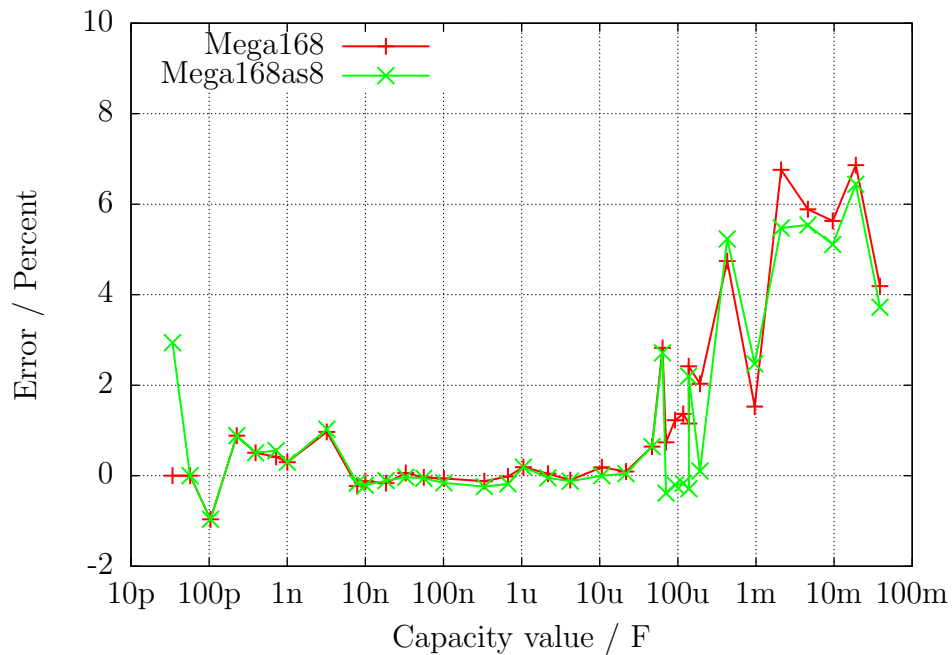


Abbildung 5.22: Prozentualer Fehler der Kondensator-Messungen mit dem ATmega168

Wie schwierig es ist, für Kondensatormessungen den richtigen Bezugswert zu finden, soll die Abbildung 5.23 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 dem wirklichen Kapazität entspricht.

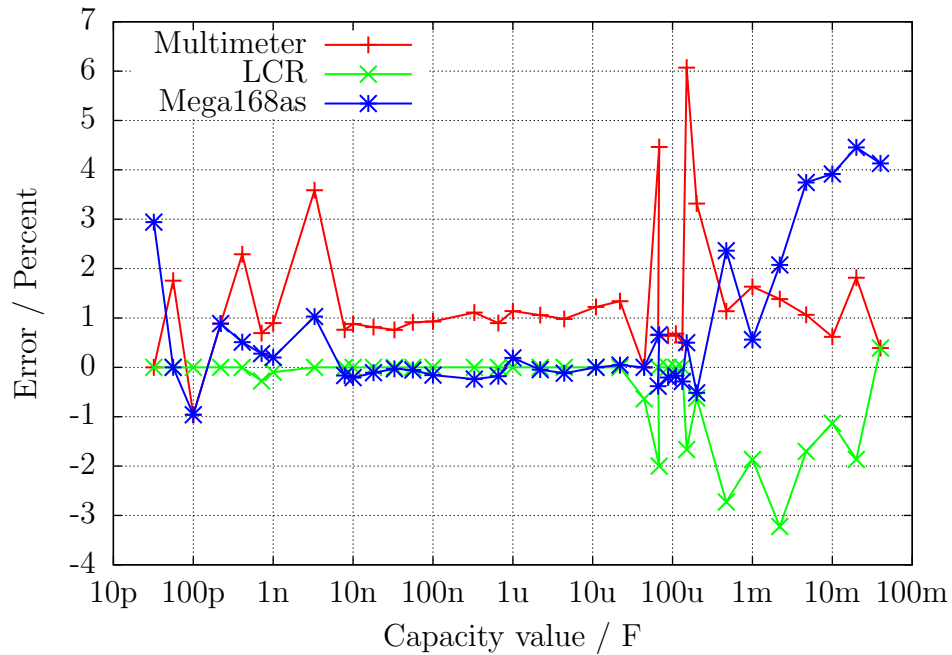


Abbildung 5.23: Vergleich der Kondensator-Messungen mit Multimeter, LCR-meter und ATmega168

Die Abweichungen der Meßergebnisse von drei verschiedenen ATmega168 Prozessoren werden in Abbildung 5.24 dargestellt. Hier wurde die Messung des LCR-Meters als Vergleichsbasis genommen. Entsprechend werden die Meßergebnisse von drei verschiedenen ATmega168A Prozessoren in Abbildung 5.25, von drei verschiedenen ATmega168PA Prozessoren in Abbildung 5.26 und von drei verschiedenen ATmega328 in Abbildung 5.27 sowie von drei ATmega328P in Abbildung 5.28 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 12 cm langen Anschlußleitungen mit den Klemmen verursacht werden. Auch das Board Layout hat Einfluß auf diesen Nullwert. Diesen Nullwert habe ich mit der Board Version „DG2BRS V 5.2.1“ ermittelt.

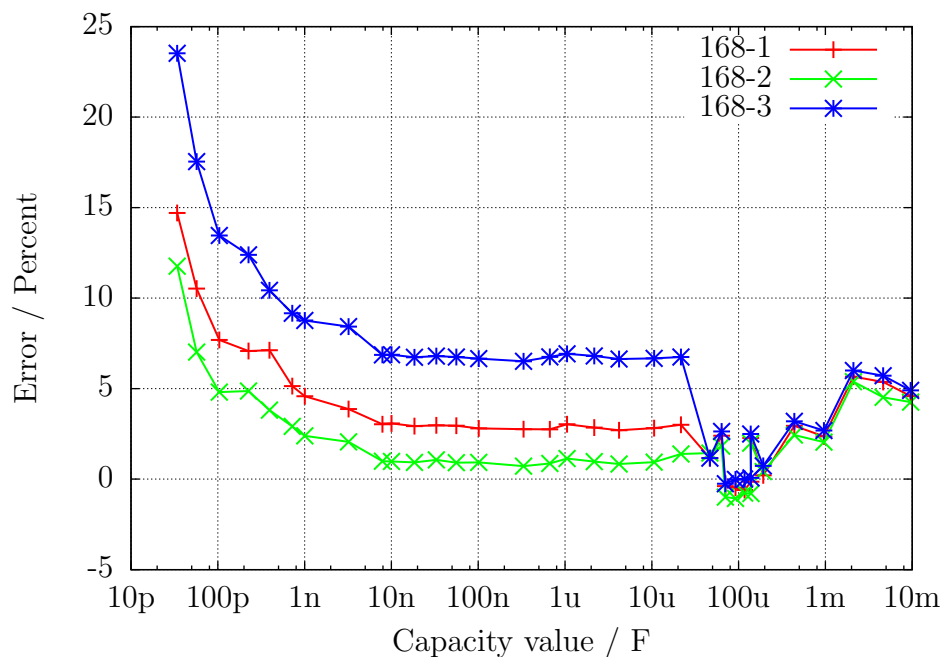


Abbildung 5.24: Kondensator-Messfehler von drei ATmega168, unkalibriert

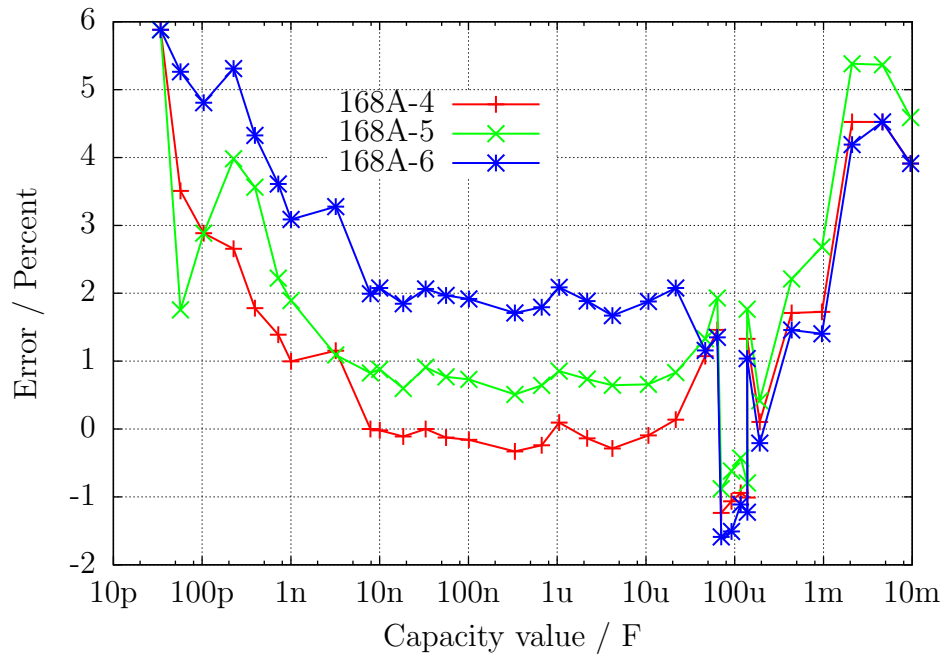


Abbildung 5.25: Kondensator-Messfehler von drei ATmega168A, unkalibriert

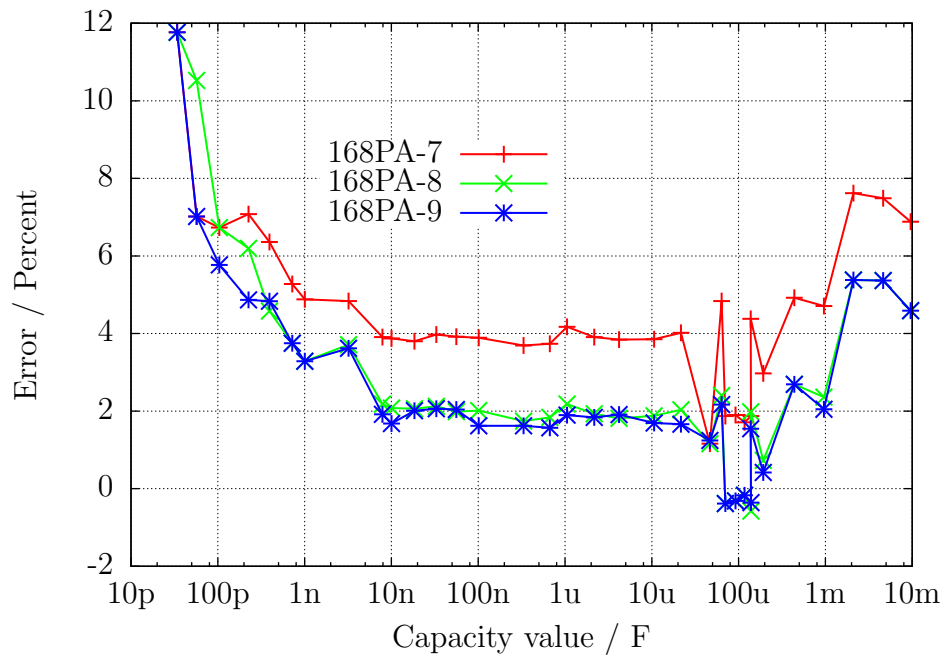


Abbildung 5.26: Kondensator-Messfehler von drei ATmega168PA, unkalibriert

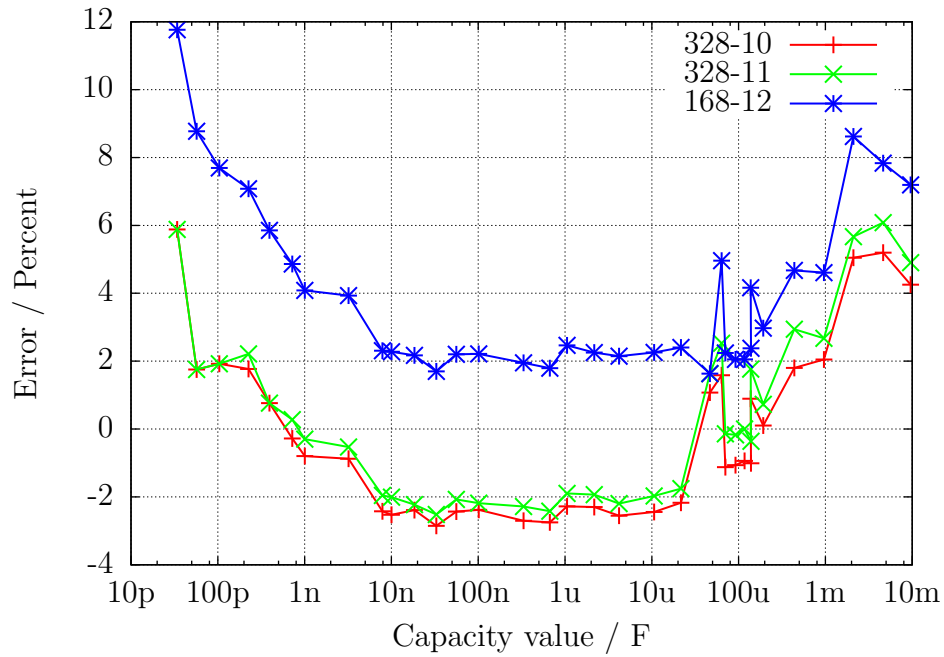


Abbildung 5.27: Kondensator-Messfehler von drei ATmega328, unkalibriert

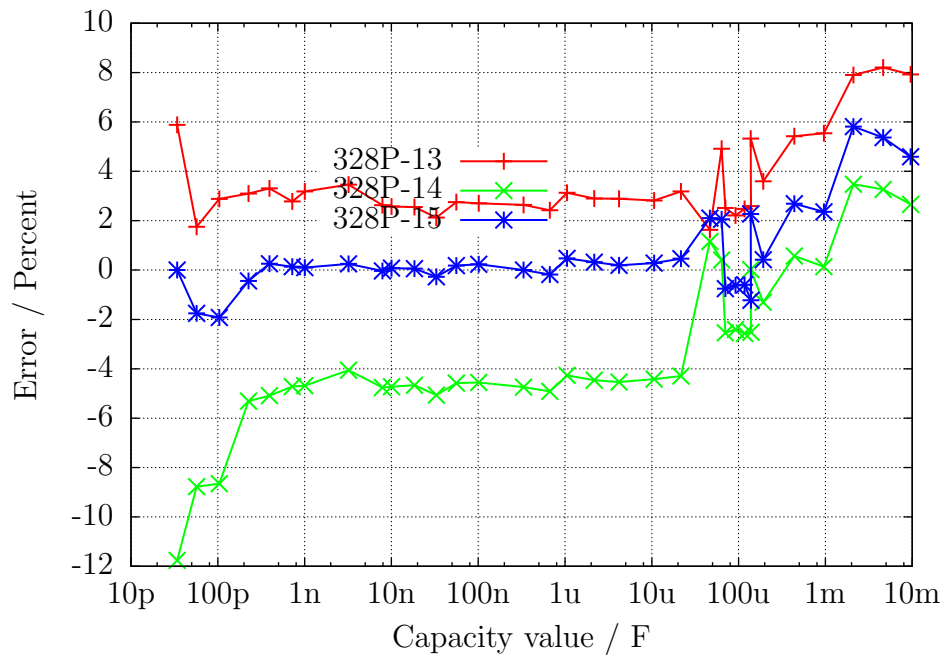


Abbildung 5.28: Kondensator-Messfehler von drei ATmega328P, unkalibriert

Zum Erreichen der besten Meßgenauigkeit muß die Software an die individuellen Eigenschaften des ATmega Exemplars 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 1 mV 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 Prozessoren habe ich bei kleinen Kapazitätswerten einen Meßfehler beobachtet, der von der Anstiegsgeschwindigkeit der Spannung beim Ladevorgang abhängig ist. Die Abbildung 5.29 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 (COMP_SLEW1=4000 und COMP_SLEW2=220) und des Selbstentladeverhaltens der großen Kondensatoren. Der Spannungsanstiegs-Korrekturfaktor berechnet sich nach $\frac{COMP_SLEW1}{cval+COMP_SLEW2} - \frac{COMP_SLEW1}{COMP_SLEW2}$, wobei cval der gemessene Kapazitätswert in pF ist.

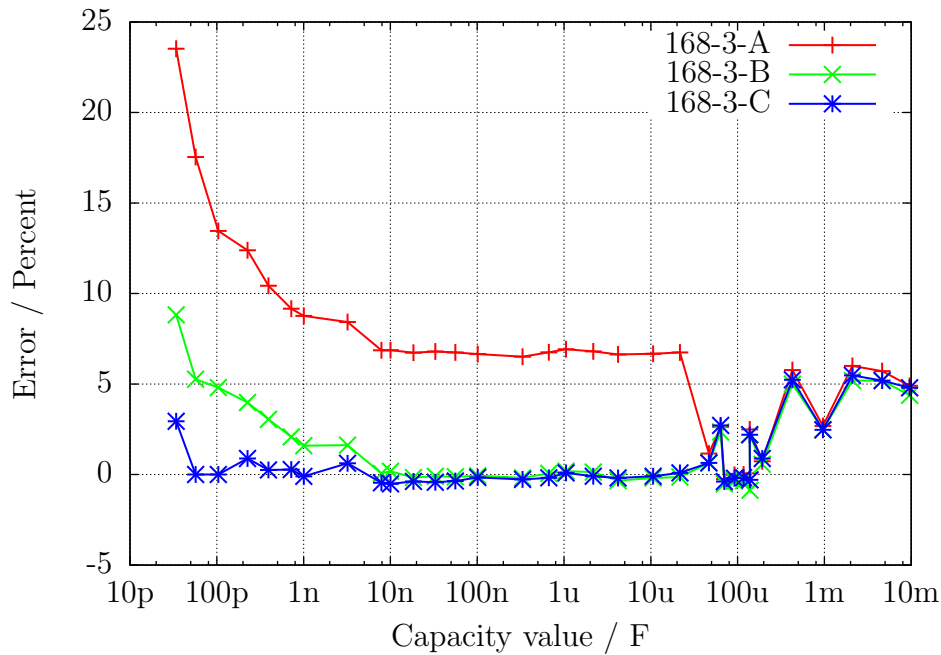


Abbildung 5.29: Optimierung der Kondensator-Messung eines ATmega168

5.3.5 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 war die Beseitigung der Exemplarstreuungen bei der Messung von Kondensatoren bis etwa $40\mu F$, wie sie in den Abbildungen 5.24, 5.25 und 5.26 gezeigt wurden. Als wesentliche Ursache wurde das unterschiedliche Verhalten (Offset Spannung) des analogen Komparators herausgefunden.

In Abbildung 5.30 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 5.24, 5.25 und 5.26 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. Mit einem mittleren Wert für diff kann man also einen guten Schätzwert für die Korrektur der Kondensatormessung aus der Differenz der

Kondensatorspannung nach dem Laden zur internen Referenzspannung berechnen. Im zweiten Teil der Abgleichprozedur muß also ein Kondensator hoher Güte zwischen Pin 1 und Pin 3 mit mindestens $100nF$ angeschlossen werden. Es sollte ein Folienkondensator, nach Möglichkeit kein keramischer Kondensator und schon gar kein Elektrolyt-Kondensator sein. Der Kapazitätswert kann bis zu $22\mu F$ betragen, braucht aber für diesen Abgleich nicht genau bekannt zu sein.

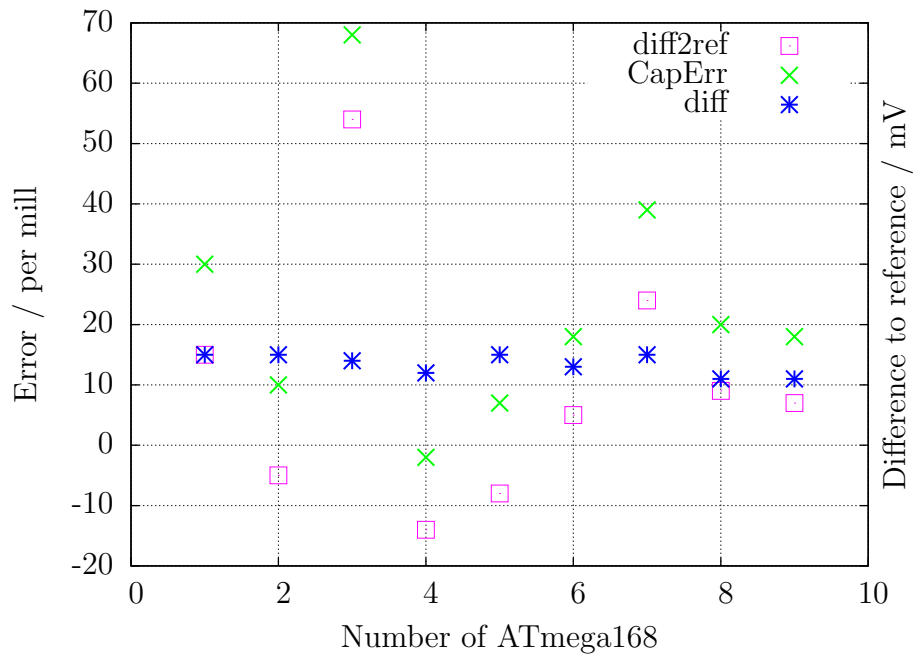


Abbildung 5.30: Daten von neun ATmega168 Prozessoren

Die Diagramme 5.31, 5.32, 5.33, 5.34 und 5.35 zeigen die Meergebnisse der gleichen Prozessoren in einer Standard-Softwarekonfiguration nach der Autokalibration. Die Prozessoren wurden alle mit der selben Software programmiert, lediglich für die ATmega168PA mußte die Makefile wegen „PARTNO = m168p“ 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.

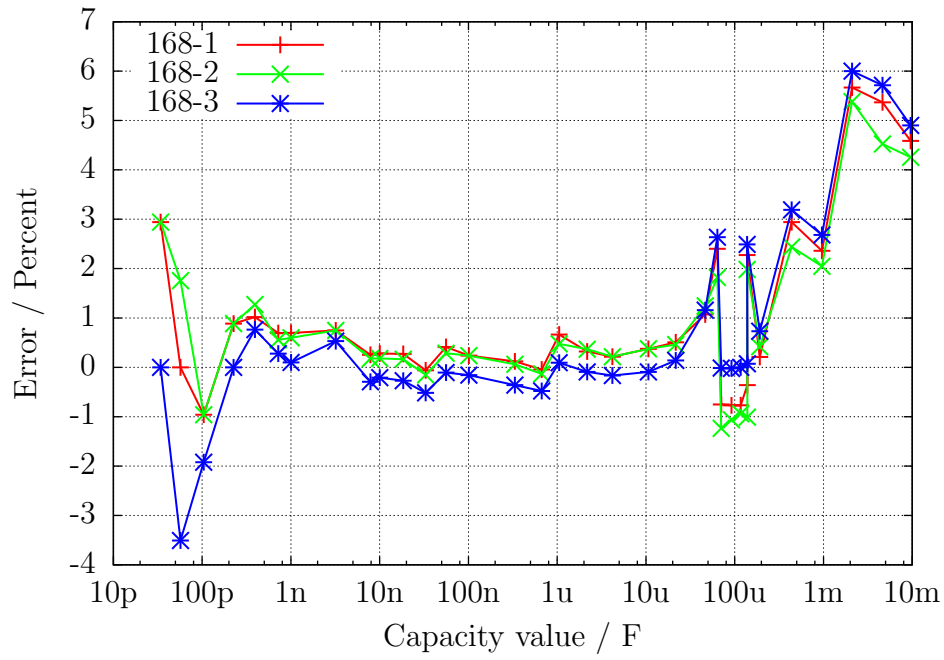


Abbildung 5.31: Kondensator-Messfehler von drei ATmega168, kalibriert

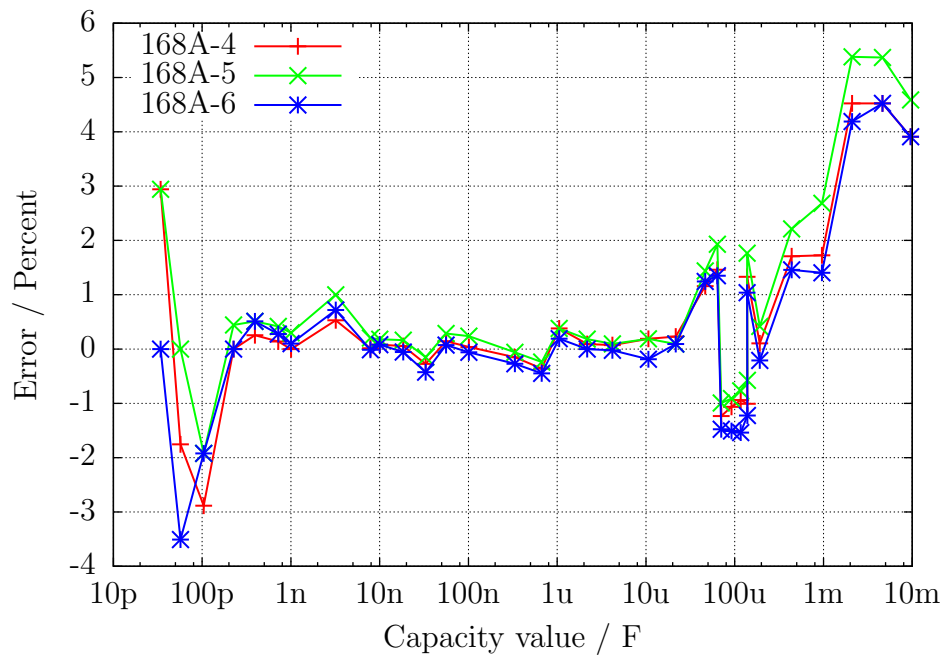


Abbildung 5.32: Kondensator-Messfehler von drei ATmega168A, kalibriert

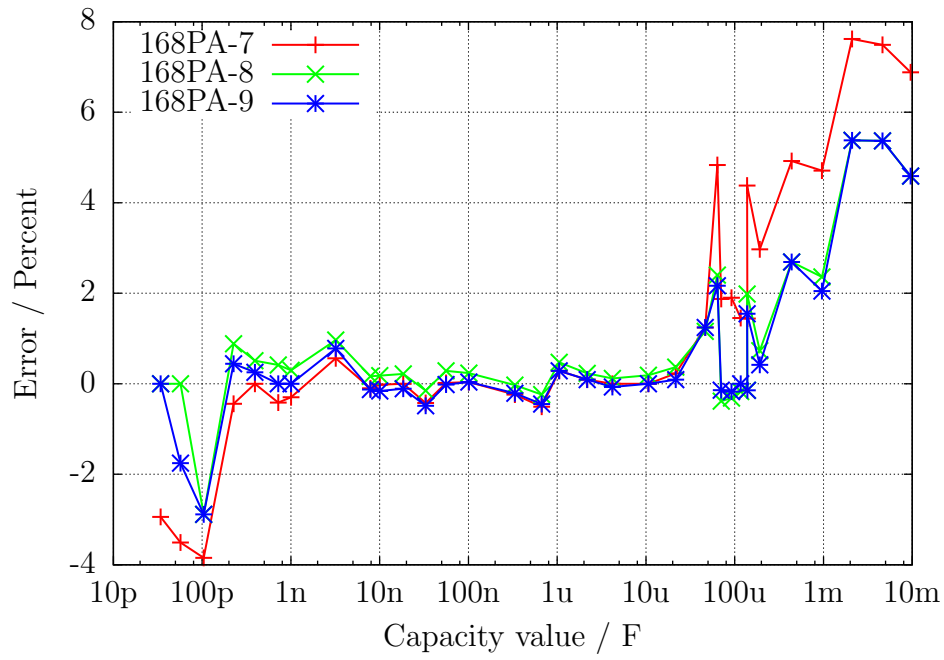


Abbildung 5.33: Kondensator-Messfehler von drei ATmega168PA, kalibriert

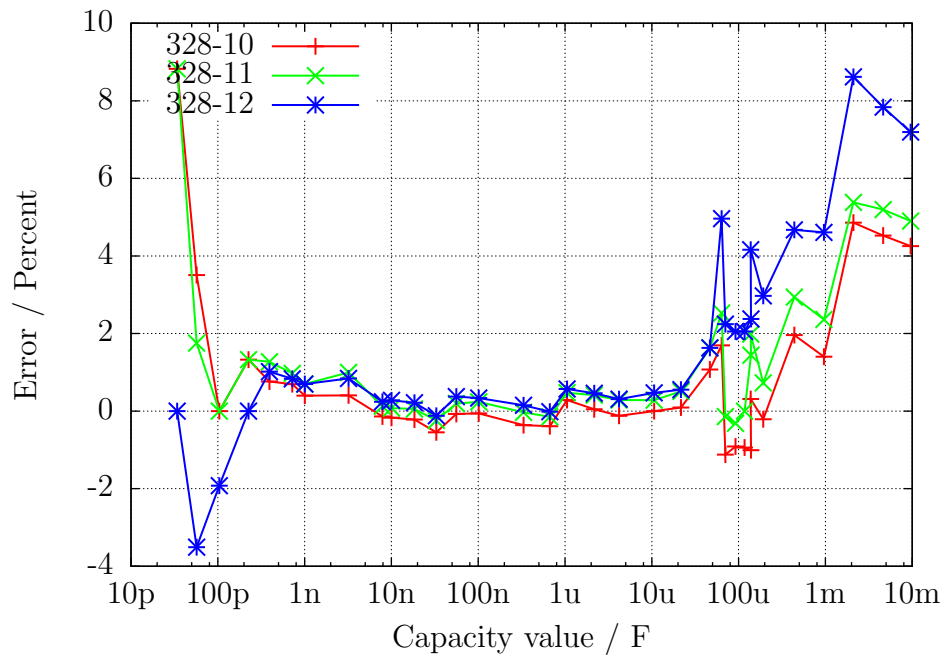


Abbildung 5.34: Kondensator-Messfehler von drei ATmega328, kalibriert

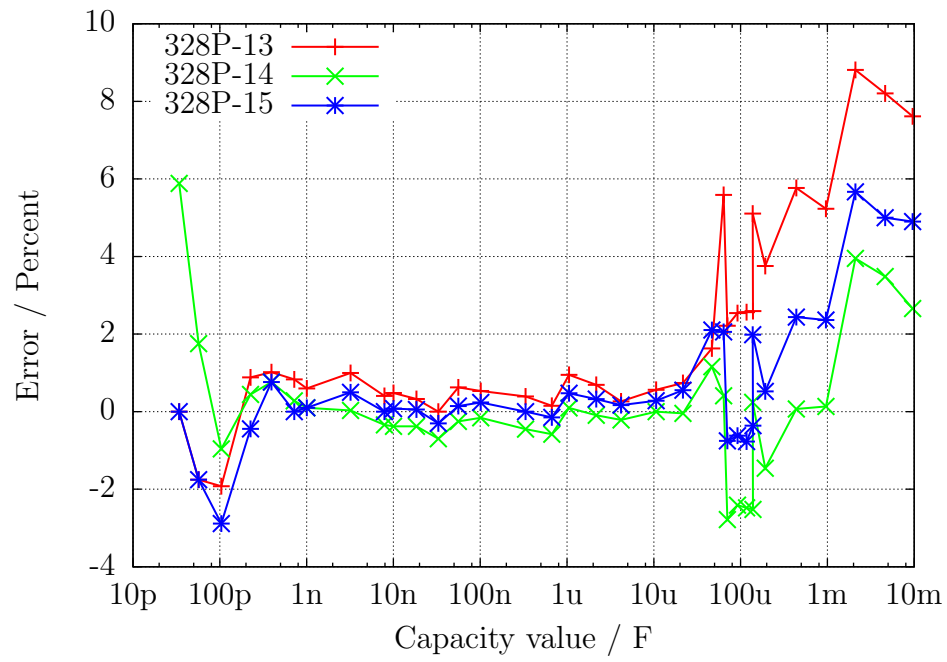


Abbildung 5.35: Kondensator-Messfehler von drei ATmega328P, kalibriert

5.4 Selbsttest Funktion

Beginnend mit Version 0.9k habe ich eine Selbsttest-Funktion eingebaut. Die Benutzung ist sehr 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. Nach Beenden des Selbsttestes wird der Tester mit normaler Messung fortfahren. Wenn kein Bauteil angeschlossen ist, wird der TransistorTester mit der Meldung „kein,unbek. oder defektes Bauteil“ enden. Die unglückliche Seite der Selbsttest-Funktion ist, daß der 8k flash Speicher fast verbraucht ist. Die Länge der ATmega8 Version ist ungefähr 8000 Bytes. Man kann die Selbsttest-Funktion zusammen mit allen anderen Optionen nur für einen ATmega168 oder ATmega328 wählen. Für einen ATmega8 muß man wegen des begrenzten flash Speichers wenigstens die Option AUTOSCALE_ADC weglassen. Die einzelnen Schritte der Selbsttestfunktion werden immer in Zeile 1 des LCD Displays mit dem Buchstaben T gefolgt von der Schritt-Nummer dargestellt. Jeder Schritt wird achtmal 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.

In jedem Schritt werden 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) Referenz Spannung (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 470kΩ Widerstand.
2. **Vergleich der 680Ω Widerstände.** In Zeile 1 wird der kryptische Text „+RL- 12 13 23“ angezeigt. Das bedeutet das folgende: Das 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. 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.

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. Ich möchte daran erinnern, daß die Auflösung des ADC etwa 4,88mV beträgt! Die Meßsituation wird auch in Abbildung 5.36 dargestellt. Alle diese Kombinationen sollten im Hinblick auf die internen Port-Widerstände folgendes ergeben:

$$\frac{5001 \cdot (19+680)}{(19+680+680+22)} = 2493 .$$

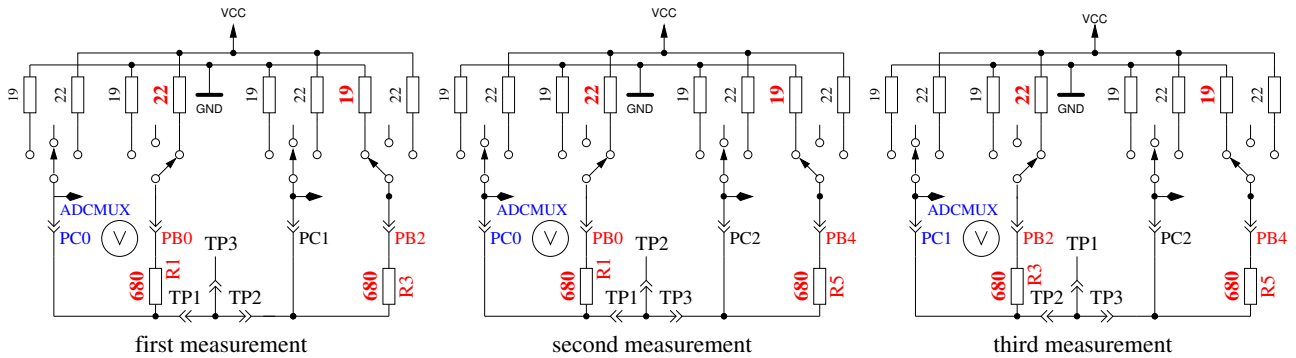


Abbildung 5.36: Vergleich der 680Ω Widerstände

3. **Vergleich der 470kΩ Widerstände.** Jetzt zeigt das Display in Zeile 1 „+RH- 12 13 23“. Das gleiche Verfahren wie in Schritt wird wiederholt mit den 470kΩ Widerständen (Symbol RH). Die Ergebnisse sollten nahe bei $\frac{5001 \cdot (19 + 470000)}{(19 + 470000 + 470000 + 22)} = 2500$ für alle Kombinationen sein.
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).
5. Dieser Schritt testet die **Fähigkeit von mit GND (-) verbundenen 470kΩ 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 470kΩ Widerständen (H) die Testpins auf VCC zu ziehen**. Zeile 1 zeigt den Text „RH+“. Der bestmögliche mV Wert für die drei Messungen wäre 5001 in Zeile 2. Große Abweichungen vom Idealwert für Schritt 5 und 6 sind Fehler wie Isolations Problem, Flußmittelreste oder beschädigter Port.
7. **Messung des Innenwiderstandes eines auf GND geschalteten Ausganges.** Der Text in der ersten Zeile des LCD ist „Ri.Lo = (mV)“. In der zweiten Zeile des LCDs werden drei Spannungswerte angezeigt. Die internen Port C Widerstände von auf GND (-) geschalteten Ausgängen werden mit dem Strom der nach VCC (+) geschalteten 680Ω Widerstände gemessen, siehe Abbildung 5.37. Es werden nur die drei Pinne des ADC Ports vermessen, die Widerstands-Ports PB0, PB2 und PB4 können nicht ohne Veränderung der Hardware gemessen werden. Es wird angenommen, daß die Port Widerstände von unterschiedlichen Ports nahezu identisch sind. Um die Widerstandswerte zu erhalten, müssen die angezeigten mV Werte durch ungefähr 7 geteilt werden (siehe Schritt 8).

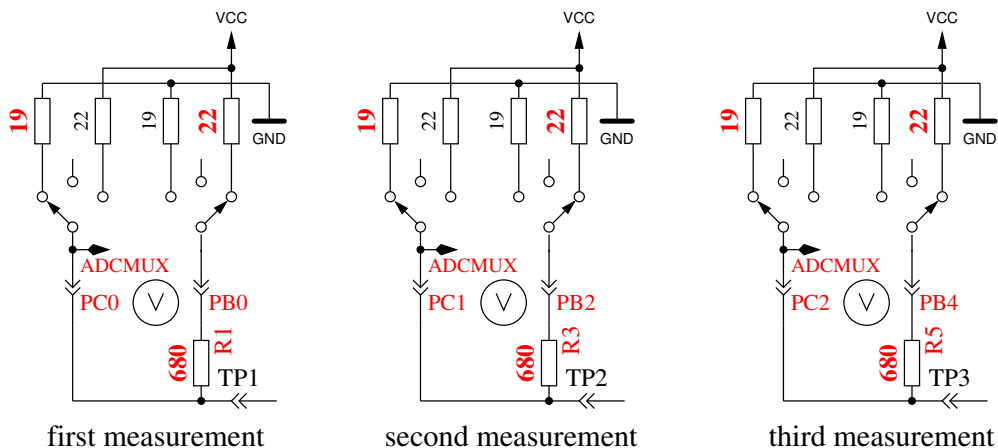


Abbildung 5.37: Messung des Innenwiderstandes von auf GND geschalteten Port C Ausgängen.

8. **Messung des Innenwiderstandes von auf VCC geschalteten Port Ausgängen.** Der benötigte Strom wird von auf GND geschalteten 680Ω Widerständen geliefert. Der Text in der ersten LCD Zeile ist „Ri.Hi= (mV)“, wenn keine AUTO_CAL Option gesetzt ist. Mit gesetzter AUTO_CAL Option wird nur „Ri.Hi=“ ausgegeben. In der zweiten Zeile des LCDs werden drei Spannungswerte angezeigt als Differenz zu VCC. Es ist die gleiche Messung wie die in Test 7 zur anderen Seite, wie in Abbildung 5.38 gezeigt wird. Mit folgenden Schritten können Sie den Innenwiderstand selbst berechnen: Um den Strom zu berechnen: $(5001 - (\text{Ergebnis von Test 7}) - (\text{Ergebnis von Test 8})) / 680$. Sie können beide Widerstandswerte erhalten, indem Sie die jeweilige Spannung (Ergebnisse von Test 7 oder Test 8) durch diesen Strom dividieren. Wenn die AUTO_CAL Option gewählt wurde und der vorige und dieser Test achtmal wiederholt wurden, wird der Mittelwert der Innenwiderstände der Ausgabe-Ports im EEprom festgehalten. Das Ergebnis für diesen Test wird dann in Zeile 1 in $d\Omega$ angezeigt, der letzte Wert der Zeile 2 ist dann das Ergebnis für den vorigen Test (auf GND geschaltet). Diese Werte stehen dann für weitere Messungen zur Verfügung.

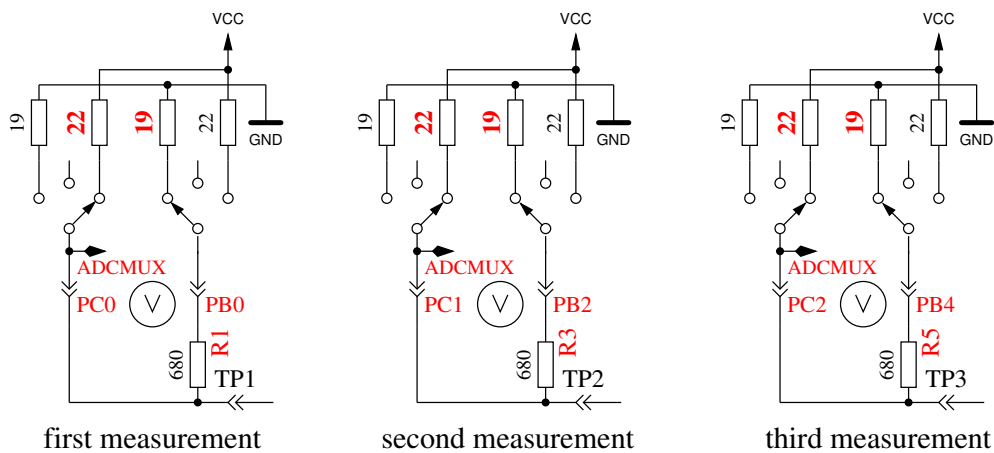


Abbildung 5.38: Messung des Innenwiderstandes des auf VCC geschalteten Port C Ausgängen.

9. **Messung des Nulloffsets der Kondensatormessung.** Für die Pin-Kombinationen 1:3, 2:3 und 1:2 wird der Nullwert der Kondensatormessung in pF ausgegeben. In der Software wird ein Vorgabewert von circa 39 pF für die normale Meßwertausgabe berücksichtigt. Für die Ausgabe dieses Tests ist keine Korrektur berücksichtigt, es wird auch kein Nulloffset abgezogen. Wenn die AUTO_CAL Option gewählt wurde und der Test achtmal wiederholt wurde, wird der Mittelwert der gefundenen Nulloffsets im EEprom festgehalten, wenn der Null Offset kleiner als 70pF ist. Der gefundene Null Offset wird für die weiteren Kapazitäts-Messungen Pin abhängig berücksichtigt. Angezeigt wird das durch die Ausgabe „EE0“ in Zeile 1, wobei die Zeile 2 die gespeicherten Null Offsets für die Pin-Kombinationen 1:3, 2:3 und 1:2 anzeigt. Bitte beachten Sie, daß Veränderungen des Meßaufbaus einen Neuabgleich sinnvoll machen. So kann sich der Nulloffset durch Verwenden von Kabeln mit Klemmen um circa 3 pF gegenüber einem leeren Sockel erhöhen.
10. **Warten auf den Anschluß eines Kondensators an Pin 1 und Pin 3.** 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 mindestens $100nF$ sein. Kapazitätswerte von bis zu $22\mu F$ können problemlos verwendet werden. Auf keinen Fall sollte man Elektrolyt-Kondensatoren verwenden.
11. **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 2 mit dem Text „REF_C=“ angezeigt und in den EEprom Speicher geschrieben.

An Ende der Selbsttest-Funktion wird der Text „Auto Test End“ in Zeile 1 und die Versionsnummer 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 Anzeige 50Hz am Ende der Zeile 1. Diese Ausgabe wird dreizigmal mit jeweils 2 Sekunden Dauer durchgeführt. Man kann die Zeit der Verzögerungs-Aufrufe prüfen, wenn man ein Oszilloskop oder einen Frequenz-Zähler besitzt. 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 gedrücktthalten des Start-Tasters vorzeitig abgebrochen werden.

Dann fährt das Programm mit der normalen Meßfunktion fort.

Test Nr.	1. Ergebnis	2. Ergebnis	3. Ergebnis
Test 1 Referenz	band gap Ref 1228 RHfakt 756		
Test 2 Vergleich 680Ω bester Wert: 2493	RL1+ RL2- 2489	RL1+ RL3- 2488	RL2+ RL3- 2488
Test 3 Vergleich 470kΩ bester Wert: 2500	RH1+ RH2- 2493	RH1+ RH3- 2493	RH2+ RH3- 2493
Test 4	isolate probe		
Test 5 470kΩ isolation bester Wert: 0	RH1- 0	RH2- 0	RH3- 0
Test 6 470kΩ isolation bester Wert: 5001	RH1+ 4996	RH2+ 4995	RH3+ 4995
Test 7 Pin Widerstand Low	TP1- RL1+ 132	TP2- RL2+ 132	TP3- RL3+ 137
Test 8 Pin Widerstand High	TP1+ RL1- 151	TP2+ RL2- 151	TP3+ RL3- 151
Test 9 Kondensator Offset	C0 1:3 34	C0 2:3 35	C0 1:2 33

Tabelle 5.5: Selbsttest Mega8, Signature 1E9307, WITH_AUTOREF

Test No.	1. Ergebnis	2. Ergebnis	3. Ergebnis
Test 1 Referenz	band gap Ref 1051 RHfakt 902		
Test 2 Vergleich 680 Ω bester Wert: 2493	RL1+ RL2- 2493	RL1+ RL3- 2493	RL2+ RL3- 2493
Test 3 Vergleich 470k Ω bester Wert: 2500	RH1+ RH2- 2494	RH1+ RH3- 2498	RH2+ RH3- 2498
Test 4	isolate probe		
Test 5 470k Ω isolation bester Wert: 0	RH1- 1	RH2- 1	RH3- 1
Test 6 470k Ω isolation bester Wert: 5001	RH1+ 5000	RH2+ 5000	RH3+ 5000
Test 7 Pin Widerstand Low	TP1- RL1+ 138	TP2- RL2+ 139	TP3- RL3+ 139
Test 8 Pin Widerstand High	TP1+ RL1- 156	TP2+ RL2- 156	TP3+ RL3- 156
Test 9 Kondensator Offset	C0 1:3 39	C0 2:3 39	C0 1:2 38

Tabelle 5.6: Selbsttest Mega168, Signature 1E9406, WITH_AUTOREF

Kapitel 6

Bekannte Fehler und ungelöste Probleme

Software Version 0.99k

1. Kondensatoren mit Kapazitäten von mehr als 40mF werden als Widerstand mit einem Wert von 2Ω erkannt. Weil Widerstände zuerst entdeckt werden, wird die Kapazitätsmessung dann nicht mehr durchgeführt. Praxisgerechtere Kondensatoren haben meist Werte von bis zu 10mF. Deshalb könnte die Kapazitätsmessung auch auf 40mF anstelle der geplanten 100mF begrenzt werden, wenn dieses Problem nicht gelöst werden kann.
2. Entladeprozedur von Kondensatoren endet nicht richtig. Es ist ein Fall bekannt geworden, wo ein ATmega8 Prozessor, der mit AUTOSCALE_ADC Option arbeiten sollte, nicht zum regulären Entlade-Ende kam. Die Spannungsgrenze ist mit unter 3mV sehr niedrig gesetzt. Bei meinen eigenen Versuchen hat die Grenze immer gereicht. Der Fehler äußert sich so, daß nach einer Zeit von etwa 12s die Meldung „Cell!“ im Display erscheint. Man kann die Option CAP_EMPTY_LEVEL in der Makefile auf höhere Werte als 3mV setzen, um den Fehler zu beseitigen, wenn keine Ursache gefunden wird.
3. Leistungs-Doppeldioden vom Schottky Typ wie MBR20100CT werden als Doppeldiode nicht erkannt.

Kapitel 7

Spezielle Software Teile

Es wurden zahlreiche Veränderungen durchgeführt um flash Speicherplatz 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_ziff1(uint8_t pin)“ in die Datei lcd_routines.c eingefügt.

Die Pseudo Funktionen 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 flash-Speichers (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). Warteaufrufe mit Zwischenwerten wie 8ms benötigen zwei Aufrufe (wait5ms() und wait3ms() oder zwei mal 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 Sprache portiert, so daß diese Erweiterung sehr effektiv integriert werden konnte. Eine funktionell identische C-Version der ReadADC Funktion ist auch beigelegt.

Kapitel 8

Arbeitsliste und neue Ideen

1. Ergänze mehr und bessere Dokumentation.
2. Prüfe, ob der TransistorTester besser „interpolierte“ ADC Werte erhalten kann, wenn zusätzliches Rauschen zum Signal oder zur Referenz hinzugefügt wird (siehe ATMEL Dokument AVR121 [5]: Enhancing ADC resolution by oversampling). Wenn alle Werte gleich sind, kann keine Verbesserung der Auflösung durch Überabtasten stattfinden. Kann genügend Rauschen mit den ATmega Zählern erzeugt werden? Natürlich kann diese Methode nicht alle ADC Fehler beseitigen.
3. Diese Methode kann getestet werden, indem ein langsam ansteigendes Eingangssignal erzeugt wird und dieses Signal beobachtet wird. Das ansteigende Signal kann erzeugt werden indem ein großer Kondensator mit dem $470k\Omega$ Widerstand geladen wird. Das Anwachsen der Spannung kann dann mit dem LCD-Display in einem speziellen Teil des Selbst-Testes beobachtet werden. Die Unterschiede der ReadADC Alternativen $\frac{44}{9}$, $\frac{22 \cdot 2}{9}$ oder $\frac{11 \cdot 4}{9}$ können ebenfalls beobachtet werden.
4. Der ADC arbeitet mit einer Taktrate von 125kHz. Die Spezifikation erlaubt einen Betrieb mit 200kHz bei voller Genauigkeit. Aber 200kHz Taktfrequenz kann mit dem Vorteiler nicht erzeugt werden, wenn die CPU-Taktrate 1MHz oder 8MHz beträgt. Wie viel Genauigkeit geht verloren, wenn die ADC Taktrate auf 250kHz gesetzt wird? Die Messungen könnten in etwa der halben Zeit erledigt werden, wenn der 250kHz Betrieb toleriert werden könnte. Bisherige Tests waren nicht ermutigend.
5. Darüber nachdenken, wie man den wirklichen Innenwiderstand der Port B Ausgänge (Widerstands-Schalter) bestimmen kann, anstatt anzunehmen, daß die Ports gleich sind.
6. Strom Idss von Drain to Source messen (oder Innenwiderstand?) mit Gate auf Source Potential for JFET.
7. Kann das Entladen von Kondensatoren beschleunigt werden, wenn der Minus-Pol zusätzlich mit dem 680Ω Widerstand nach VCC (+) geschaltet wird?
8. Können Induktivitäten gemessen werden?
9. Wie verändern sich die Meßergebnisse, wenn die Betriebs-Spannung zwischen 4,5V und 5V variiert wird?
10. Prüfe, ob der Tester Fließkomma Darstellung von Werten gebrauchen kann. Das Überlaufsrisiko (overflow) ist geringer. Man braucht keine Multiplikation / Division Konstruktion um einen Faktor mit einer gebrochenen Zahl nachzubilden. Aber ich weiß nicht wieviel Platz für die Bibliothek gebraucht wird.

11. Schreibe eine Gebrauchsanweisung zum Konfigurieren des Testers mit den Makefile Optionen und beschreibe den Ablauf bis zum fertigen Prozessor.
12. 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)?
13. Prüfe die Ports nach dieser Aktion mit der Selbsttest-Funktion!
14. Können Spannungsregler geprüft werden? (Eingang, Ausgang, GND)
15. Können Optokoppler getestet werden?
16. Ist es möglich den ESR von Elektrolyt-Kondensatoren zu bestimmen? Zum Beispiel beim Entladen mit den Port C Ausgängen und beim Laden mit dem 680Ω Widerstand.
17. Ausgabe einer Warnung, wenn die relativ zu VCC gemessene Referenz-Spannung nicht plausibel im Hinblick auf das ATmega Modell ist.
18. Kann mit einer an PC4 angeschlossenen Präzisions-Spannungsreferenz die VCC Spannung und die interne Referenz kalibriert werden?
19. Wie wäre es mit einem neuen Tester mit einem größeren ATmega, der differenzielle ADC-Ports hat, mehr Programm (flash) Speicher ...? Es gibt keinen ATxmega, der eine Versorgungsspannung von 5V hat, nur die ATmega Linie ist möglich.
20. Idee für ein neues Projekt: USB Version ohne LCD-Display, Power vom USB Port, Kommunikation zum PC über eine USB-Serial Brücke.

Literaturverzeichnis

- [1] Markus Frejek *AVR-Transistortester*,. Embedded Projects Journal, 11. Ausgabe, 2011
- [2] Atmel Corporation *8-bit AVR with 8KBytes In-System Programmable Flash - ATmega8(L)*,. Manual, 2486Z-AVR-02/11, 2011
- [3] Atmel Corporation *8-bit AVR with 4/8/16/32KBytes In-System Programmable Flash - ATmega48 - ATmega328*,. Manual, 8271D-AVR-05/11, 2011
- [4] Atmel Corporation *Atmel AVR126: ADC of megaAVR in Single Ended Mode*,. Application Note, 8444A-AVR-10/11, 2011
- [5] Atmel Corporation *Atmel AVR121: Enhancing ADC resolution by oversampling*,. Application Note, 8003A-AVR-09/05, 2005
- [6] http://en.wikibooks.org/wiki/LaTeX_LaTeX_documentation,. Guide to the LaTeX markup language, 2012
- [7] http://en.wikibooks.org/wiki/Gnuplot_Gnuplot_documentation,. Documentation for the plotting tool gnuplot, 2012
- [8] <http://www.xfig.org/userman> *Xfig documentation*,. Documentation of the interactive drawing tool xfig, 2009
- [9] http://docs.gimp.org/2.6/de_gimp_documentation. Documentation of the GNU Image Manipulation Program, 2010
- [10] <http://www.mikrocontroller.net/articles/AVR-Transistortester> *Online Dokumentation des Transistortester*, Online Article, 2009-2011
- [11] <http://www.mikrocontroller.net/articles/AVRDUDE> *Online Dokumentation des avrdude programmer interface*, Online Article, 2004-2011
- [12] <http://www.mikrocontroller.net/topic/131804> *Thread von Markus F.*, Forum thread, 2009
- [13] http://www.mikrocontroller.net/articles/AVR_Transistortester *Kurze Eigenschaftenbeschreibung des TransistorTester von Karl-Heinz K.*, Online Article, 2012
- [14] <http://www.mikrocontroller.net/topic/248078> *Thread von Karl-Heinz K.*, Thread und Software Versionen, 2012