

TransistorTester with AVR microcontroller
and a little more
Version 0.99k

Karl-Heinz Kübbeler
`kh_kuebbeler@web.de`

September 11, 2012

Contents

1	Features	3
2	Hardware	5
2.1	Circuit of the TransistorTester	5
2.2	Hints for building the TransistorTester	6
2.3	Programming of the microcontroller	7
2.4	Troubleshooting	7
3	Instructions for use	9
4	Configuring the TransistorTester	11
5	Description of the measurement procedures	15
5.1	Measurement of Semiconductors	17
5.1.1	Measurement of PNP Transistor or P-Channel-MOSFET	17
5.1.2	Measurement of NPN Transistor or N-Channel-MOSFET	18
5.1.3	Measurement of Diodes	20
5.1.4	Results of different measurements	20
5.2	Resistor Measurement	22
5.2.1	Resistor Measurement with 680 Ohm Resistors	22
5.2.2	Resistor Measurement with 470 kOhm resistors	24
5.2.3	Results of the resistor measurements	25
5.3	Measurement of Capacitors	29
5.3.1	Discharging of Capacitors	29
5.3.2	Measurement of big Capacitors	29
5.3.3	Measurement of small Capacitors	31
5.3.4	Results of Capacitor measurement	33
5.3.5	Automatic calibration of the capacitor measurement	37
5.4	Selftest Function	41
6	Known errors and unsolved problems	48
7	Special Software Parts	49
8	To Do List and new ideas	50

Preface

Basically Motive

Every hobbyist knows the following problem: You disassemble a Transistor out of a printed board or you get one out of a collection box. If you find out the identification number and you already have a data sheet or you can get the documents about this part, everything is well. But if you don't find any documents, you have no idea, what kind of part this can be. With conventional approach of measurement it is difficult and time-consuming to find out the type of the part and parameters. It could be a NPN, PNP, N- or P-Channel-Mosfet etc. It was the idea of Markus F. to hand over the work to a AVR microcontroller.

As my work has started

My work with the software of the TransistorTester of Markus F. [1] has started, because I had problems with my programmer. I had bought a printed board and components, but I could not program the EEprom of the ATmega8 with the Windows driver without error messages. Therefore I took the software of Markus F. and changed all the accesses from the EEprom memory to flash memory accesses. By analysing the software in order to save memory at other places of program, I had the idea, to change the result of the ReadADC function from ADC units to millivolt (mV) units. The mV resolution is needed for any output of voltage values. If ReadADC returns directly the mV resolution, I can save the conversion for each output value. This mV resolution can be get, if you first accumulate the results of 22 ADC readings. The sum must be multiplied with two and divided by nine. Then we have a maximum value of $\frac{1023 \cdot 22 \cdot 2}{9} = 5001$, which matches perfect to the wanted mV resolution of measured voltage values. So I additionally had the hope, that the enhancement of ADC resolution by oversampling could help to improve the voltage reading of the ADC, as described in AVR121 [5]. The original version ReadADC has accumulated the result of 20 ADC measurements and divides afterwards by 20, so the result is equal to original ADC resolution. By this way never a enhancement of ADC resolution can take place. So I had to do little work to change the ReadADC, but this forced analysing the whole program and change of all "if-statements" in the program, where voltage values are queried. But this was only the beginning of my work!

More and more ideas to make measurement faster and more accurate has been implemented. Additionally the range of resistor and capacity measurements are extended. The output format for LCD-Display was changed, so symbols are taken for diodes, resistors and capacitors instead of text. For further details take a look to the actual feature list chapter 1. Planned work and new ideas are accumulated in the To Do List chapter 8. By the way, now I can program the EEprom of the ATmega with Linux operating system without errors.

Chapter 1

Features

1. Operates with ATmega8, ATmega88, ATmega168 or ATmega328 microcontrollers.
2. Displaying the results to a 2x16 character LCD-Display.
3. One key operation with automatic power shutdown.
4. Battery operation is possible since shutdown current is only about 20nA.
5. Low cost version is feasible without crystal and auto power off.
6. Automatic detection of NPN and PNP bipolar transistors, N- and P-Channel MOSFETs, JFETs, diodes, double diodes, Thyristors and Triacs.
7. Automatic detection of pin layout of the detected part.
8. Measuring of current amplification factor and Base-Emitter threshold voltage of bipolar transistors.
9. Darlington transistors can be identified by the threshold voltage and high current amplification factor.
10. Detection of the protection diode of bipolar transistors and MOSFETs.
11. Measuring of the Gate threshold voltage and Gate capacity value of MOSFETs.
12. Up to two Resistors are measured and shown with symbols  and values. All symbols are surrounded by the probe numbers of the Tester (1-3). So Potentiometer can also be measured. If the Potentiometer is adjusted to one of its ends, the Tester cannot differ the middle pin and the end pin.
13. Resolution of resistor measurement is now 0.1Ω , values up to $50M\Omega$ are detected.
14. One capacitor can be detected and measured. It is shown with symbol  and value. The value can be from 25pF (8MHz clock, 50pF @1MHz clock) to 40mF with a resolution of up to 1 pF (@8MHz clock).
15. Up to four digits are shown for capacitor and resistor values with the correct dimension.
16. Up to two diodes are shown with symbol  or symbol  in correct order. Additionally the flux voltages are shown.
17. LED is detected as diode, the flux voltage is much higher than normal. Two-in-one LEDs are also detected as two diodes.

18. Zener-Diodes can be detected, if reverse break down Voltage is below 4.5V. They are shown as two diodes, you can identify this part only by the voltages. The outer probe numbers, which surround the diode symbols, are identical in this case. You can identify the real Anode of the diode only by the one with break down (threshold) Voltage nearby 700mV!
19. If more than 3 diode type parts are detected, the number of founded diodes is shown additionally to the fail message. This can only happen, if Diodes are attached to all three probes and at least one is a Z-Diode. In this case you should only connect two probes and start measurement again, one after the other.
20. Measurement of the capacity value of a single diode in reverse direction. Bipolar Transistors can also be analysed, if you connect the Base and only one of Collector or Emitter.
21. Only one measurement is needed to find out the connections of a bridge rectifier.
22. Capacitors with value below 25pF are usually not detectet, but can be measured together with a parallel diode or a parallel capacitor with at least 25pF. In this case you must subtract the capacity value of the parallel connected part.
23. Testing time is about two seconds, only capacity measurement can cause longer period.
24. Software can be configured to enable series of measurements before power will be shut down.
25. Build in selftest function with 50Hz Frequency generator to check the accuracy of clock frequency and wait calls.
26. Selectable facility for ATmega168 or ATmega328, to calibrate the zero offset of capacity measurement and the internal port resistance of port output with the selftest. A external capacitor with at least $100nF$ between pin 1 and pin 3 is necessary to compensate the offset voltage of the analog comparator. This can reduce measurement errors of capacitors of up to $40\mu F$. With the same capacitor a correction voltage to the internal reference voltage is found to adjust the gain for ADC measuring with the internal reference.

Thyristors and Triacs can only be detected, if the test current is above the holding current. Some Thyristors and Triacs need as higher gate trigger current, than this Tester can deliver. The available testing current is only about 6mA! It is possible that not all options remains in future releases, because this software is currently in a testing stage, You can select all features only for microcontroller with mor program memory such as ATmega168.

Attention: Allways be shure to discharge capacitors before connecting them to the Tester! The Tester may be damaged before you have switched it on. There is only a little protection at the ATmega ports.

Chapter 2

Hardware

2.1 Circuit of the TransistorTester

The circuit of the TransistorTester in figure 2.1 is based on the circuit of Markus F. released in Abb. 1 of AVR-Transistortester report [1]. Changed or moved parts are marked with green color, optional parts are marked with red color. Some changes are made because the electronical power switch make problems in some implementations. Therefore the resistor R7 is reduced to $3.3k\Omega$. The capacitor C2 is reduced to 10nF and R8 is moved so that the PD6 output does not try to switch the C2 capacitor directly. Additional blocking capacitors are added and should be placed near the power connection of the Atmega and near the Voltage regulator. Because the PD7 input and PC6 (RESET) are the only pins, where pull up resistors where needed, one extra $27k\Omega$ resistor is added to the PD7 (pin 13) input. With this modification the software can disable all internal pull up resistors of the ATmega. The additional crystal with its 22pF capacitors are optional added. The accuracy of a crystal has the benefit of more stable time measurement for getting the capacitor values. New software version can use a scale switch of the ADC. The speed of switching is reduced by the external capacitor C1 at the AREF (21) pin of the ATmega. To avoid slowing down the measurement speedi more than necessary, the value of this capacitor should be reduced to 1nF. Removing of the capacitor C1 is also possible. For adapting the software to the actual circuit take a look to the Makefile options in the configuring chapter 4. Some different versions of R11 / R12 resistor combinations circulates in the internet. I have adapted my software to the original of Markus F. [1] with $10k\Omega$ and $3.3k\Omega$. The additional 2.5V precision voltage reference connected at pin PC4 (ADC4) is planned to use with future software versions and is currently not used. A optional ISP connector has been added to easier load new software versions to the tester.

2.3 Programming of the microcontroller

I release the software for the microcontroller with source code. The development is done with Linux operating system (Ubuntu) and is controlled with a Makefile. The Makefile makes shure, that your software will be compiled with the prior selected Makefile options. Some constellations are precompiled with the source. Please take a look to the ReadMe.txt file in the directory Sourcecode/default and to the chapter 4. The result of compilation have the extensions .hex and .eep . Usually the names will be TransistorTester.hex and TransistorTester.eep . The .hex file contains the data for the program memory (flash) of the ATmega processor. The .eep file contains the data for the EEprom memory of the ATmega. Both data files must be loaded to the correct memory. Additionally the operating state of the ATmega processor must be programmed with the “fuses”. If you can use my Makefile and additionally the program avrdude, you need no exact knowledge of the details about the fuses. You have only to type “make fuses” if you have no crystal or “make fuses-crystal” if you have installed the 8MHz crystal to your printed board. With the ATmega168 series of the microcontroller you can also use “make fuses-crystal-lp” to use a crytal with the low power mode. Never choose the crystal mode of clock generation, if you don’t have installed the 8MHz crystal. If you are not shure with the fuses, leave them as default set by manufactor and first bring the the tester to operation in this mode. Maybe your program runs too slow, if you use program data compiled for 8MHz operation, but you can correct this later! But a wrong set of fuses may inhibit later ISP-programming.

2.4 Troubleshooting

In most cases of problems you will miss the text output to the LCD-display. At first you should check, if the LED was illuminated weak, if you release the Test button.

Power does not switch on. If the LED is without light and the VCC power has correct 5V voltage during holding the Test button, the microcontroller does not switch the power correctly. The microcontroller should hold the power by switching the PD6 output to 5V, which is usually done as one of the first actions. If you hold the Test key pressed, the power is switched on anyway. So you can check the value of VCC power and additionally the voltage value of the PD6 output, if you hold the key pressed. If VCC voltage has correct value (5V), but PD6 voltage is below 4V, your microcontroller does not start the program. In this case you should check if the microcontroller flash has been loaded with proper data for your installed type and if ATmega is correctly configured with the fuses. If your ATmega put the PD6 output to 5V and the power does not stay if you release the Test key, it is more difficult to find the reason. First you can shorten the LED and try again. If your Tester now starts, your LED may be faulty or mounted with wrong polarity. If this is not the reason, the current amplification factor of your T3 transistor (BC557C) is insufficient. The current to the base of T3 is lower in the microcontroller state as in the “key pressed” state.

Nothing is readable on the LCD display Check the voltage at the contrast pin at the LCD display (pin 3). Adjust to correct value specified in the data sheet of your display and optimize by viewing. If you have a high temperature display type, you must provide a negative contrast voltage for operation. In this case you can use the ICL 7660 device for generating a negative voltage from positive 5V. If there is no output readable on the LCD and the background light is on, you should disconnect the power and check all four data plus the two control signal connections. If all connection are well, the only reason I see is a uncorrect timing of control signals. This can be caused by a slower LCD controller than expected by the software or the ATmega software runs at wrong clock speed. Please check for which clock speed your programming data was compiled and if the fuses of the ATmega are correct set to that speed. You find the clock parameter in the corresponding Makefile.

Something but not all is readable on the LCD display Check if the .EEP data are loaded to the EEPROM memory of ATmega. If all data are loaded correctly, you should check the clock speed of your programming data (Makefile) and ATmega processor settings (fuses).

Measurement is slow and Capacitors are measured about 8 times too small You run software compiled for 8MHz clock at real clock speed of 1MHz. Please set the fuses of the ATmega correctly.

Measurement has strangely values Check if your programmer is still connected to the ISP-plug. The ISP interface should be disconnected for measuring. Very often the reason of wrong measurements is the use of software compiled with the AUTOSCALE_ADC option and with the option NO_REF_CAP, but the capacitor at the AREF pin has still a value of 100nF. Wrong assembly of components or remaining soft solder flux can disturb the measurements too. Please check with the selftest function of your TransistorTester software if possible. For the details see Chapter 5.4. Otherwise inspect your board visually and check the resistor values with a ohmmeter. You can use the pins of the ATmega for this check, for example to check the R1 you can measure between pin 23 and pin 14. Take a look at the circuit diagram 2.1 for details. There is no need to remove the microcontroller, only battery or power supply should be removed before.

Chapter 3

Instructions for use

Using of the Transistor-Tester is simple. Anyway some hints are required. In most cases are wires with alligator clips connected to the test ports with plugs. Also sockets for transistors can be connected. In either case you can connect parts with three pins to the three test ports in any order. If your part has only two pins, you can connect this pins to any two of the tree test ports. Normally the polarity of part is irrelevant, you can also connect pins of electrolytical capacitors in any order. The measurement of capacity is normally done in a way, that the minus pole is at the test port with the lower number. But, because the measurment voltage is only between 0.3 V and at most 1.3 V, the polarity doesn't matter. When the part is connected, you should not touch it during the measurement. You should put it down to a nonconducting pad if it is not placed in a socket. You should also not touch to the isolation of wires connected with the test ports, the measurement results can be affected. Then you should press the start button. After displaying a start message, the measurement result should appear after two seconds. If capacitors are measured, the time to result can be longer corresponding to the capacity.

How the transistor-tester continues, depends on the configuration of the software.

Single measurement mode If the tester is configured for single measurement mode, the tester shut off automatical after displaying the result for 10 seconds for a longer lifetime of battery. During the display time a next measurement can be started by pressing the start button. After the shut off a next measurement can be started too of course. The next measurement can be done with the same or another part.

Endless measurement mode A special case is the configuration without automatical shut off. This configuration is normally only used without the transistors for the shut off function. A external off switch is necessary for this case. The tester will repeat measurements until power is switched off.

Multi measurement mode In this mode the tester will shut down not after the first measurement but after a configurable series of measurements. In the standard case the tester will shut down after five measurements without found part. If any part is identified by test, the tester is shut down after double of five (ten) measurements. A single measurement with unknown part after a series of measurement of known parts will reset the counter of known measuerements to zero. Also a single measurement of known part will reset the counter of unknown measurements to zero. This behavior can result in a nearly endless series of measurements without pressing the start button, if parts are disconnected and connected in periodical manner.

In this mode there is a special feature for the display period. If the start button is pressed only short for switching on the tester, the result of measurement ist only shown for three seconds. Buf if you press and hold the start button until the first message is shown, the further

measurement results are shown for ten seconds. The next measurement can started earlier by pressing the start button during the displaying of result.

If the software is configured with the selftest function, the selftest can be started by connecting all three test ports together. In this case all of the documented tests in the Selftest chapter 5.4 will be done. The repetition of the tests can be avoided, if the start button is hold pressed. So you can skip uninteresting tests fast and you can watch interesting tests by releasing the start button. If the function `AUTO_CAL` is selected in the Makefile, there will be calibrated the zero offset for the capacity measurement and the internal resistance of the port output. But for this feature the tests must be repeated the full eight times for building the mean values. If you connect a capacitor with at least $100nF$ to pin 1 and pin 3 during test step 10, the offset voltage of the analog comparator will be compensated for better measurement of capacity values and the gain for ADC measurements with the internal reference voltage will be adjusted too for better resistor measurement results with the `AUTOSCALE_ADC` option.

Chapter 4

Configuring the TransistorTester

The complete software for the TransistorTester is available in source code. The compilation of modules is controlled with a Makefile. The development was done at the Ubuntu Linux operating system with the GNU toolchain (gcc version 4.5.3). It should be possible to use other Linux operating systems without problems. To load the compiled data to the flash memory or the EEPROM memory, the tool avrdude (version 5.11svn) was taken by the Makefile, if you call “make upload”. The program avrdude is available for Linux and Windows operating system. The gnu C-compiler gcc is also taken by the AVR studio software at the Windows operating system. You can load the program data (.hex and .eep) also with other tools to the ATmega, but only my Makefile version takes care to load the correct data to the choosed processor. Avrdude loads only data to the ATmega if the Signature Bytes of the connected ATmega is identical to the choosed one. If you alter the Makefile, all the software will be compiled new, if you call a “make” or “make upload” command. The software compiled for a ATmega8 does not run on a ATmega88. The software compiled for a ATmega168 does not run on the ATmega88, even if the ATmega88 has enough flash memory! Be careful, if you don’t use my Makefile. With the correct options set, my software runs on the unchanged hardware of Markus F (PARTNO=M8, NO option NO_AREF_CAP and NO PULLUP_DISABLE option). The clock rate can also be set to 8 MHz with fuses, no crystal is required!

The following options in the Makefile are available to configure the software for your Tester.

PARTNO describes the target processor:

- m8 = ATmega8
- m48 or m48p = ATmega48
- m88 or m88p = ATmega88
- m168 or m168p = ATmega168
- m328 or m328p = ATmega328
- example: PARTNO = m168

UI_LANGUAGE specifies the favored Language

- LANG_ENGLISH, LANG_GERMAN, LANG_POLISH, LANG_CZECH, LANG_SLOVAK and LANG_SLOVENE is currently available
- example: UI_LANGUAGE = LANG_ENGLISH

LCD_CYRILLIC is only needed for a LCD-display with cyrillic character set. The μ and Ω character is not available with the cyrillic character set. If you specify this option, both characters are loaded to the LCD with software.

- example: CFLAGS += -DLCD_CYRILLIC

WITH_SELFTEST If you specify this Option, software will include a selftest function. Selftest will be started, if you connect all three probes together and start measurement.

- example: CFLAGS += -DWITH_SELFTEST

AUTO_CAL The zero offset for capacity measurement and the port output resistor values will be written additionally to the EEprom with the selftest routine. Additionally the offset voltage of the analog comparator (REF_C_KORR) and the voltage offset of the internal reference voltage (REF_R_KORR) will be measured automatically, if you connect a capacitor with alt least $100nF$ to pin 1 and pin 3 in the test No. 10. All found values will be written to EEprom and will be used for further measurements automatically.

example: CFLAGS += -DAUTO_CAL

FREQUENCY_50HZ At the end of selftest a 50 Hz Signal will be generated on Port 2 and Port 3 for up to one minute.

example: CFLAGS += -DFREQUENCY_50HZ

R_MESS enables the resistor measurement. This option should allways be set.

example: CFLAGS += -DR_MESS

C_MESS enables the capacity measurement. This option should allways be set.

example: CFLAGS += -DC_MESS

CAP_EMPTY_LEVEL This option defines the voltage level for discharged capacitor (mV units).

You can set the level to higher value as 3mV, if the tester does not finish discharging of capacitors. In this case the tester ends after longer time with the message "Cell!".

example: CFLAGS += -DCAP_EMPTY_LEVEL=3

WITH_AUTO_REF specifies, that reference voltage is read to get the actual factor for capacity measuring of low capacity values (below $50\mu F$).

example: CFLAGS += -DWITH_AUTO_REF

REF_C_KORR specifies a offset for readed reference voltage in mV units. This can be used to adjust the capacity measurement of little capacitors. A correction value of 10 results to about 1 percent lower measurement results.

example: CFLAGS += -DREF_C_KORR=14

C_H_KORR specifies a correction value for the measurement of big capacitor values. A value of 10 results to 1 percent lower measurement results. If the option AUTO_CAL is selected together with the WITH_SELFTEST option, the REF_C_KORR will be a offset to the measured voltage difference of the test capacitor and the internal reference voltage.

example: CFLAGS += -DREF_C_KORR=10

AUTOSCALE_ADC enables the automatic scale switchover of the ADC to either VCC or internal reference. Internal reference gives a 2.56V scale for ATmega8 and a 1.1V scale for other processors.

example: CFLAGS += -DAUTOSCALE_ADC

NO_AREF_CAP tells your Software, that you have no Capacitor installed at pin AREF (pin 21).

This enables a shorter wait-time for the AUTOSCALE_ADC scale switching of the ADC. A 1nF capacitor was tested in this mode without detected errors. Figure 4.1a and 4.1b show the switching time with a 1nF capacitor. As you can see the switching from 5V to 1.1V is much slower than switching back to 5V. If you have still installed the 100nF, switching time will be about factor 100 longer!

example: CFLAGS += -DNO_AREF_CAP

REF_R_KORR specifies a offset for the internal ADC-reference voltage in mV units. With this offset a difference by switching from VCC based ADC reference to internal ADC reference for

resistor measurement can be adjusted. If you select the AUTO_CAL option of the selftest section, this value is a offset to the found voltage difference in the AUTO_CAL function.

example: CFLAGS += -DREF_R_KORR=10

OP_MHZ tells your software at which Clock Frequency in MHz your Tester will operate. The software is tested only for 1 MHz, 8MHz and additionally 16MHz. The 8MHz operation is recommended for better resolution of capacity measurement.

example: OP_MHZ = 8

USE_EEPROM specifies if you wish to locate fix text and tables in EEprom Memory. Otherwise the flash memory is used. Recommended is to use the EEprom (option set).

example: CFLAGS += -DUSE_EEPROM

PULLUP_DISABLE specifies, that you don't need the internal pull-up resistors. You must have installed a external pull-up resistor at pin 13 (PD7) to VCC, if you use this option. This option prevents a possible influence of pull-up resistors at the measuring ports (Port B and Port C).

example: CFLAGS += -DPULLUP_DISABLE

ANZ_MESS this option specifies, how often an ADC value is read and accumulated. You can select any value between 5 and 200 for building mean value of one ADC measurement. Higher values result to better accuracy, but longer measurement time. One ADC measurement with 44 values takes about 4.7ms.

example: CFLAGS += -DANZ_MESS=25

POWER_OFF This option enables the automatic power off function. If you don't specify this option, measurements are done in a loop infinitely until power is disconnected with a ON/OFF switch. If you have the tester without the power off transistors, you can deselect the option POWER_OFF. If you have NOT selected the POWER_OFF option with the transistors installed, you can stop measuring by holding the key several seconds after a result is displayed until the time out message is shown. After releasing the key, the tester will be shut off by timeout. You can also specify, after how many measurements without a founded part the tester will shut down. The tester will also shut down the power after twice as much measurements are done in sequence without a single failed part search. If you have forgotten to unconnect a test part, total discharging of battery is avoided. Specify the option with a form like CFLAGS += -DPOWER_OFF=5 for a shut off after 5 consecutive measurements without part found. Also 10 measurements with any founded part one after another will shut down. Only if any sequence is interrupted by the other type, measurement continues. The result of measurement stay on the display for 10 seconds for the single measurement, for the multiple measurement version display time is reduced to 3 seconds (set in config.h). If the start key is pressed a longer time on power on time, the display time is also 10 seconds for the multiple measurement. The maximum value is 255 (CFLAGS += -DPOWER_OFF=255).

example 1: CFLAGS += -DPOWER_OFF=5

example 2: CFLAGS += -DPOWER_OFF

BAT_CHECK enables the Battery Voltage Check. If you don't select this option, the version number of software is output to the LCD instead. This option is usefull for battery powered tester version to remember for the battery change.

example: CFLAGS += -DBAT_CHECK

BAT_OUT enables Battery Voltage Output on LCD (if BAT_CHECK is selected). If your 9V supply has a diode installed, use the BAT_OUT=600 form to specify the threshold voltage (mV) of your diode to adjust the output value. Also the voltage loss of transistor T3 can be respected

with this option. threshold level does not affect the voltage checking levels (BAT_POOR).

example 1: CFLAGS += -DBAT_OUT=300

example 2: CFLAGS += -DBAT_OUT

BAT_POOR sets the poor level of battery voltage to the specified 100mV (1/10 Volt) value. The warning level of battery voltage is always 1V higher than the specified poor level. Setting the poor level to low values such as 5.4V is not recommended for rechargeable 9V batteries, because this increase the risk of battery damage by the reason of the deep discharge! If you use a rechargeable 9V Battery, it is recommended to use a Ready To Use type, because of the lower self-discharge.

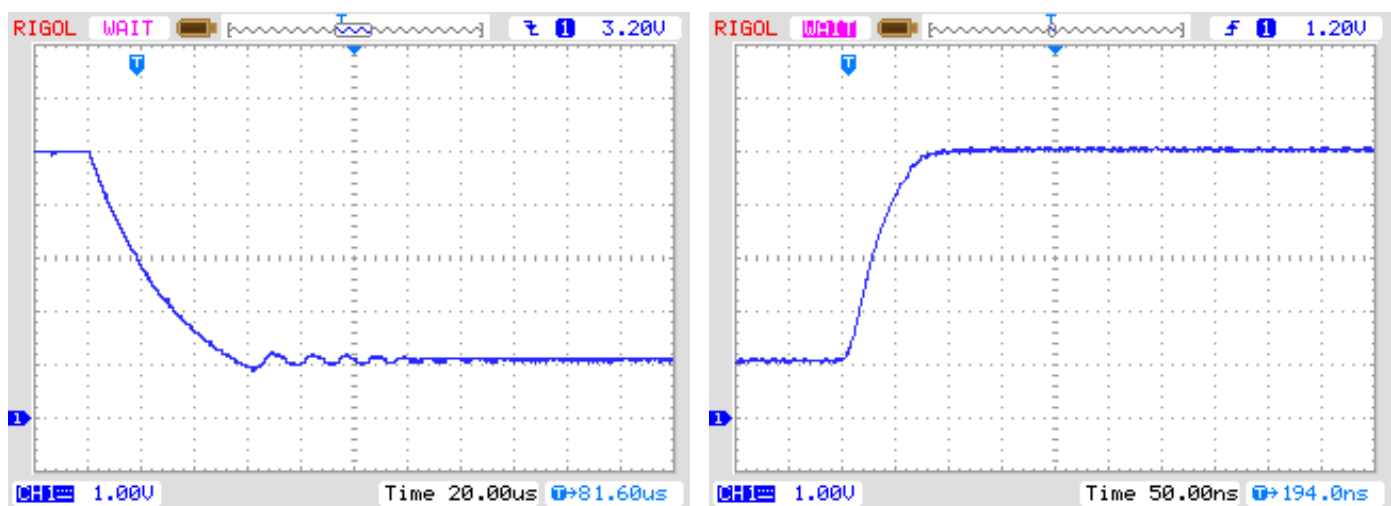
example for low drop regulator (5.4V): CFLAGS += -DBAT_POOR=54 example for 7805 type regulator (6.4V): CFLAGS += -DBAT_POOR=64

PROGRAMMER select your programmer type for avrdude interface program. The correct selection of this option is needed, if you use the “make upload” or “make fuses” call of this Makefile. For further information please look to the manual pages of avrdude and online documentation [11].

example: PROGRAMMER=avrisp2

PORT select the port where avrdude can reach your microcontroller (atmega). and port if you will use avrdude For further information please look to the manual pages of avrdude.

example: PORT=usb



(a) from 5V to 1.1V

(b) from 1.1V to 5V

Figure 4.1: AREF switching with a $1nF$ Capacitor

Additional parameters can be set in the files `transistortester.h` and `config.h`. The file `transistortester.h` contains global variables and defines the port / pin constellation and the resistor values used for measurement. The file `config.h` specifies parameter for different processor types, wait times and the clock frequency of the ADC. Normally there is no reason to change these values.

Chapter 5

Description of the measurement procedures

The simplified schematic of a Input/Output-Port pin of the ATmega is shown in figure 5.1. The PUD switch isolates all “pull up” resistors of the ATmega. The output of a pin can be switched off with the DD switch. The Input can operate regardless to the state of the switch DD. The PORT switch usually defined the output level, but also switches the pull up resistor. Because the Switches PORT and DD can not be changed at the same time but only one after another, the pull up resistors can disturb the measurement. Therefore I prefer to disable the pull up resistors with the PUD switch. Of course all the switches are electronic type and the resistors 19Ω and 22Ω are approximated values.

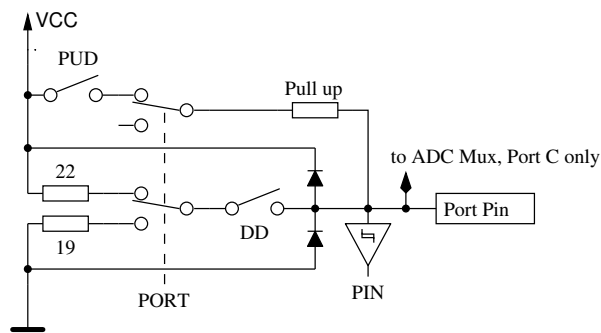


Figure 5.1: simplified diagram of each ATmega port pin

Every of the three terminal probes of your Transistor Tester is build with three ATmega port pins, which is shown as simplified diagram for the terminal probe TP2 (middle of three pins) in figure 5.2.

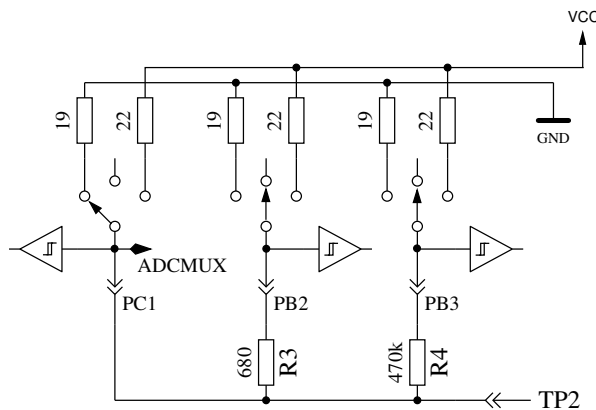


Figure 5.2: simplified circuit of each measurement terminal probe TP

Every test pin (measurement port) can be used as digital or analog input. This measurement capability is independent of using the port as output. Every test pin can be switched to output and in this mode it can be directly connected to GND (0V) or VCC (5V), or it can be connected via a 680Ω resistor or a $470k\Omega$ resistor to either GND or VCC. Table 5.1 shows all possible combination of measurements. Notice, that the positive state can be switched directly to VCC (Port C) or it can be connected with the 680Ω resistor to VCC (Port B). The same possibility has the negative state of terminal probe to the GND side. The test state means, that probe can be open (Input), connected with the $470k\Omega$ resistor to VCC or GND, or that the probe can be connected with the 680Ω resistor to VCC or GND.

	state pin 1	state pin 2	state pin 3
1.	positive	negative	test
2.	positive	test	negative
3.	test	negative	positive
4.	test	positive	negative
5.	negative	test	positive
6.	negative	positive	test

Table 5.1: all combinations of measurement

If the capacitor measuring is configured for the tester, the tester will try to discharge the capacitors connected at all test pins. If discharge will fail, that means the remaining voltage is too high, the discharging will be aborted after about 12 seconds with the message "Cell!". This can also happen, if no capacitor is connected to any test pin. The cause for this can be, that the cut-off voltage is chosen too low for this ATmega. You can choose a higher voltage with the Makefile option CAP_EMPTY_LEVEL.

5.1 Measurement of Semiconductors

One probe pin is assumed to be the negative side of the component. Another pin is assumed to be the positive side of the component. For a first test, the components positive side is directly connected to VCC. The negative side is connected with the 680Ω resistor to GND. The test probe (third pin, also called TriStatePin) is first connected with the 680Ω resistor for 10 ms to GND. The voltage of the negative probe pin is read, during the TriStatePin is switched to Input (High Impedance). It is assumed that the tested part can be a N-Channel MOSFET and the gate should be discharged. If the readed voltage is above 976mV, the next test assume, that the tested part can also be a P-Channel MOSFET and for this a 10ms switch of the TriStatePin with the 680Ω resistor to the VCC side is done. Also for this case the voltage at the negative Probe pin is read. If the voltage of the negative pin is greater than 92mV with the currentless TriStatePin, additional tests are made to differ N-Channel JFET or D-MOSFET (depletion) and P-Channel JFET or P-MOSFET. MOSFET versions can be differed by the missing of current in any state of the TriStatePin. If the component has no current between positive probe and negative probe without signal at the TristatePin, the next tests are specified in the next section 5.1.1. If current was detected, the next test is described in the diode section 5.1.3.

5.1.1 Measurement of PNP Transistor or P-Channel-MOSFET

First the current amplification factor is measured with common collector for the assumed PNP transistor. The measuring situation is shown in figure 5.3. If the measured voltage at the Base (UB) is above 9mV with the 680Ω resistor, the hFE is build as $hFE = \frac{UE-UB}{UB}$. The voltage UE is the difference of the Emitter-voltage to VCC. The difference between the 22Ω and 19Ω resistors are not respected. If the UB voltage is below 10mV, the measurement is done with the 470kΩ resistor at the base. In this case the current amplification factor is build as $hFE = \frac{UE \cdot 470000}{UB \cdot (680 + 22)}$.

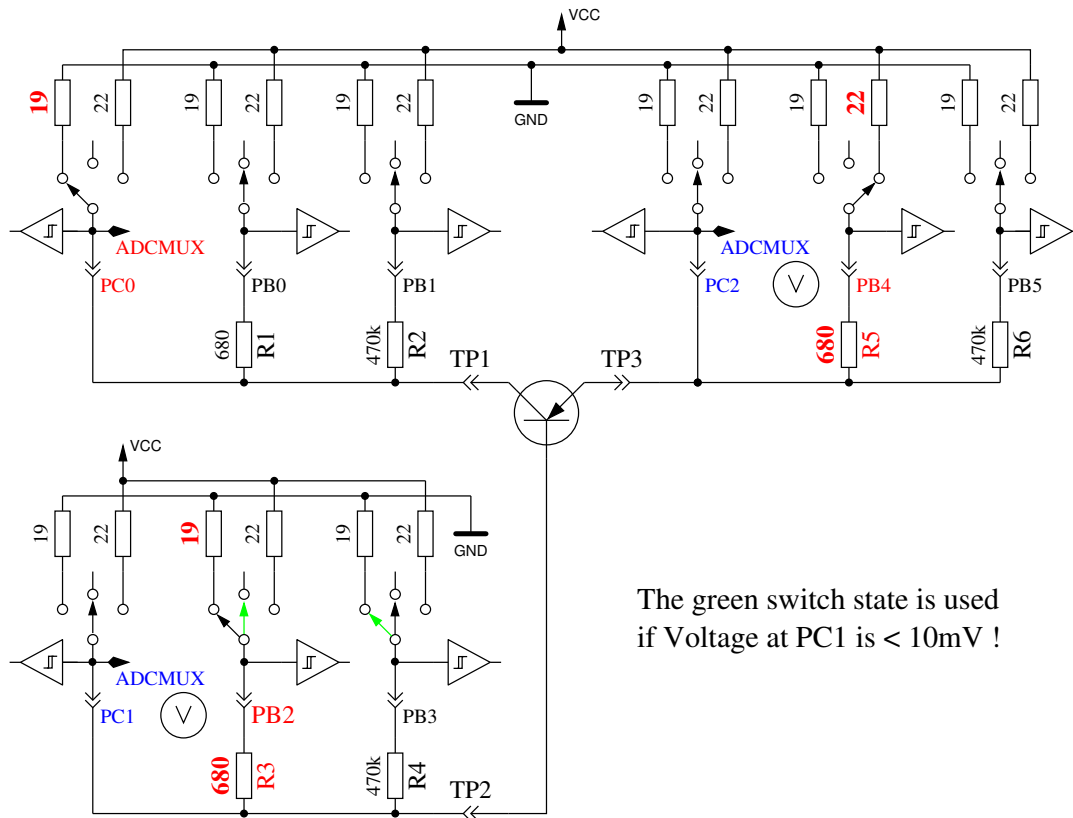


Figure 5.3: hFE measurement of PNP transistor with common collector circuit

Next the tests with common emitter are done for the assumed PNP transistor. The positive side of component is now direct connected to VCC, the negative side 680Ω resistor is connected to GND as shown in Figure 5.4. If the negative side of component has a voltage of above 3.4V, when the base side 680Ω resistor was connected to GND, it must be a PNP transistor or a P-Channel FET. This can be easy find out by analysing the base voltage. If the base voltage is greater 0.97V, it must be a PNP. For measuring the current amplification factor, the 470kΩ resistor is taken as Base resistor instead of the 680Ω. The current amplification factor is build by $hFE = \frac{UC \cdot 470000}{UB \cdot (680 + 19)}$. The higher current amplification factor is assumed to be the right one, this one or the one found with the common collector circuit. The values found for the PNP are only valid, if a second set of measurements is done. In order to prevent detecting the PNP in the inverse mode (collector and emitter are swapped), the measurement with the higher current amplification is taken as the right one. If base voltage is lower than 0.97V, it must be a P-E-MOS. In this case the gate threshold voltage is measured by switching the gate slowly with the 470kΩ resistor up and down, waiting for a digital input signal change of the Drain side and then read the voltage of the gate pin.

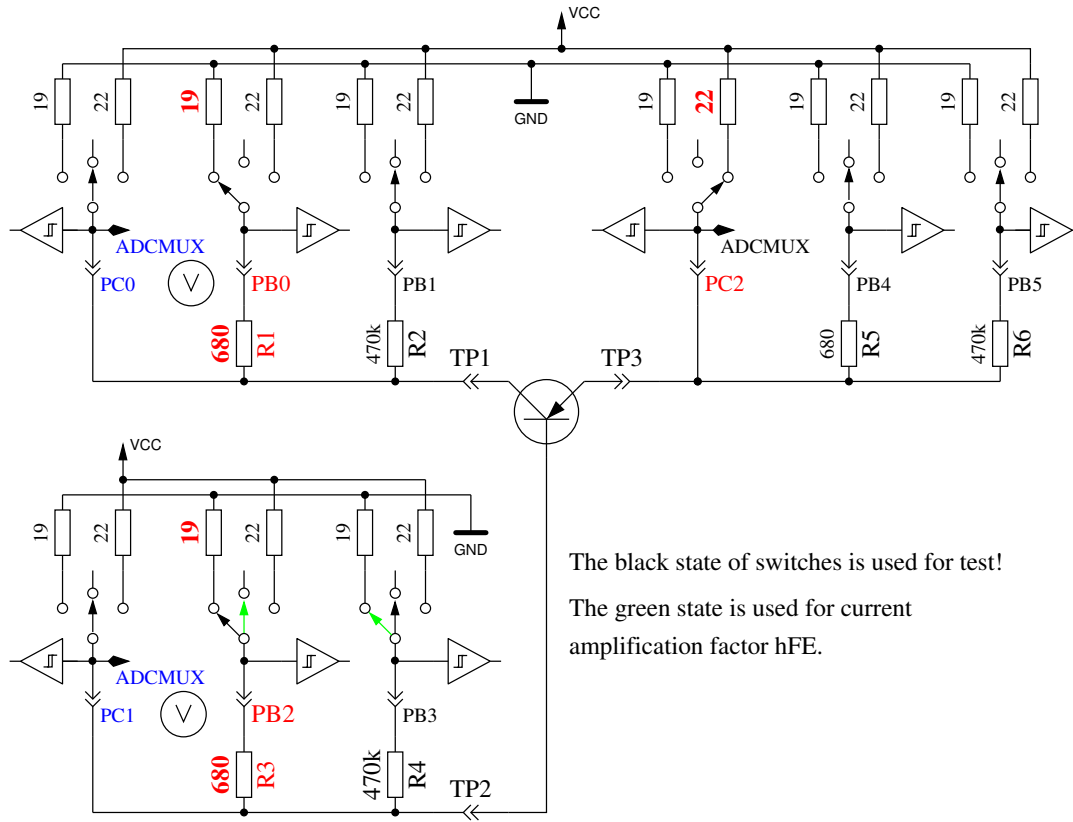


Figure 5.4: test and hFE measurement of PNP transistor with common emitter circuit

5.1.2 Measurement of NPN Transistor or N-Channel-MOSFET

The measuring of NPN-Transistors begin in the same way as PNP-Transistors with measuring the current amplification factor in the common collector circuit. First measurement is done with a 680Ω base resistor switched to VCC. If the voltage at the base resistor ist too low, the 470kΩ resistor is taken instead. Measurement then continues with the common emitter circuit as shown in figure 5.5.

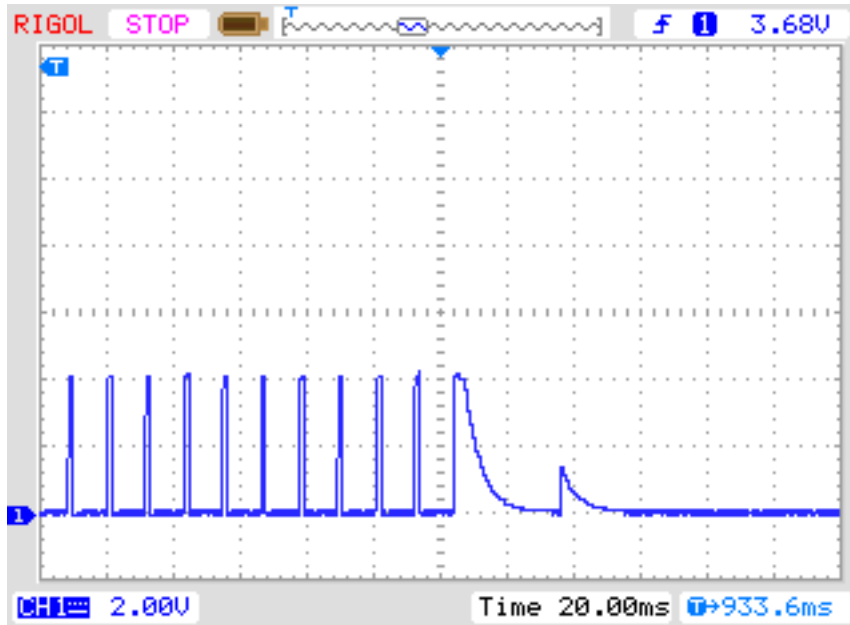


Figure 5.6: measuring of threshold voltage of N-Channel-MOSFET

5.1.3 Measurement of Diodes

If current is detected with the pre-tests, the behavior of the part will be checked to be a diode. The flow voltage with the 680Ω resistor must be between 0.15V and 4.64V. The flux voltage with the 680Ω must be greater than 1.125 times the flux voltage with the $470k\Omega$ resistor and eight times the flux voltage with the $470k\Omega$ must be greater than the flux voltage with the 680Ω resistor. I hope, that this behavior is always a diode.

5.1.4 Results of different measurements

The following three tables shows results of different test probes with different software configurations with the ATmega8 and ATmega168 processors.

Diode Type	Mega8@8MHz signature 1E9307 WITH_AUTO_REF	Mega168 @8MHz signature 1E9406	Mega168 @8MHz signature 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, 6pF	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

Table 5.2: measurement results of diode testing

Transistor Type	Mega8@8MHz signature 1E9307 WITH_AUTO_REF	Mega168 @8MHz signature 1E9406	Mega168 @8MHz signature 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=25.5k, 1.20V
BC516	PNP, B=65535, 1.41V	PNP, B=65535, 1.41V	PNP, B=81.5k, 1.17V
BRY55/200	Thyristor	Thyristor	Thyristor

Table 5.3: measurement results of bipolar transistor testing

FET Type	Mega8@8MHz signature 1E9307 WITH_AUTO_REF	Mega168 @8MHz signature 1E9406	Mega168 @8MHz signature 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

Table 5.4: measurement results of MOS transistor testing

5.2 Resistor Measurement

Each resistor is measured with four different types of measurement in one current direction. The same resistor is also tested with the same four measurement types in the other current direction. The measurement in the opposite direction is only used to identify a resistor. If mismatch between both measurements is too big, it's not a resistor.

5.2.1 Resistor Measurement with 680 Ohm Resistors

The measurement of a unknown resistor R_x is done in two ways with the build in precision 680 Ω resistors. The diagram of this measurements for test pin 1 (TP1) and test pin 3 (TP3) are simplified shown in figure 5.7 and figure 5.8 as a example of the six choices of probe combinations.

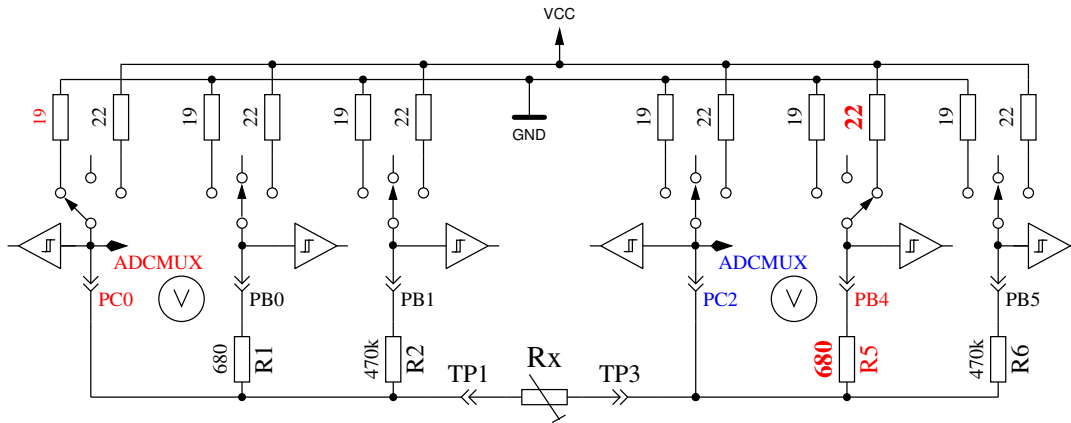


Figure 5.7: Measurement type 1 with 680 Ω

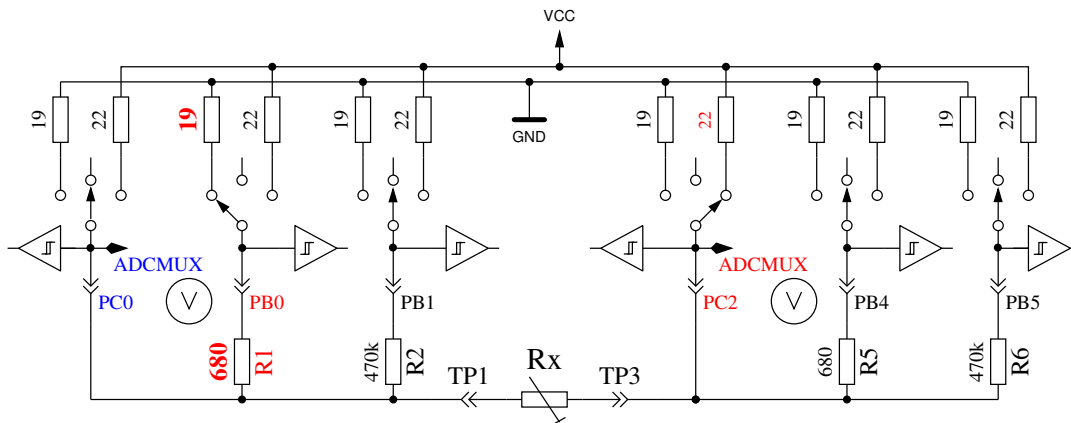


Figure 5.8: Measurement type 2 with 680 Ω

On the left side test pin 1 is shown and on the right side you can see test pin 3. In both diagrams you see, that the terminal 3 (right side) is connected to VCC, the left side is connected to GND. The direction of current flow through the resistor R_x is always the same. The values of ports switched to output are shown with red color, the values of ports used as Input are shown in blue color, the inactive ports are black. In both shown measurement types the current should have the same value, because the sum of resistor values between VCC and GND is identical (if the build in resistors are identical). Usually the measured voltage is not the same, because the sequence of resistors has changed.

The V symbol within the circle marks the ports used for voltage measurement. In both configurations the value of resistor Rx can be computed with the known resistor values and the measured voltages, if the relation of resistor Rx and the 680Ω is not too high. The theoretical voltage gradient is shown in figure 5.9, where resistor values are shown in logarithmic scale.

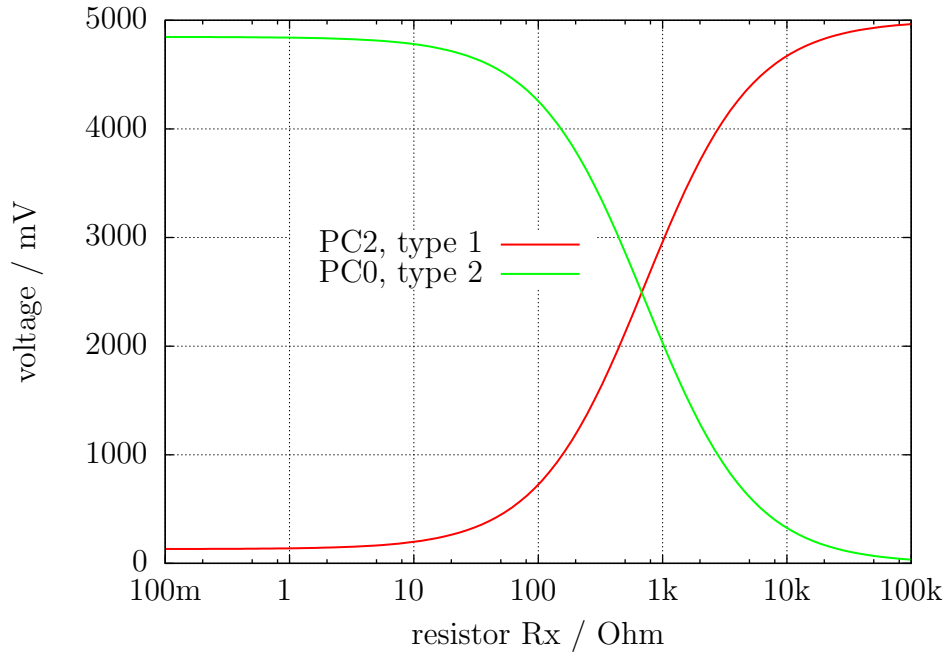
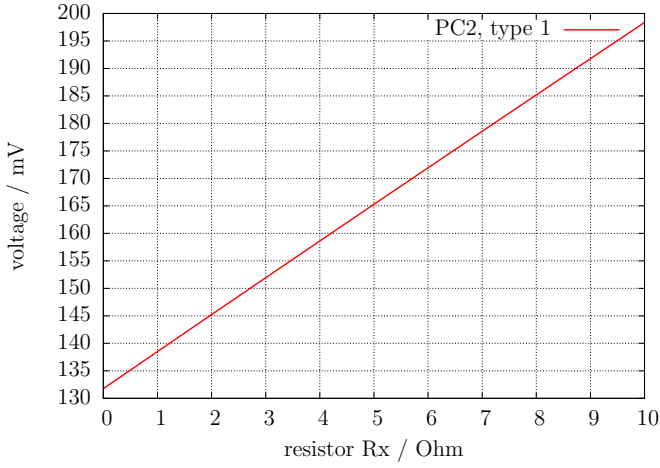


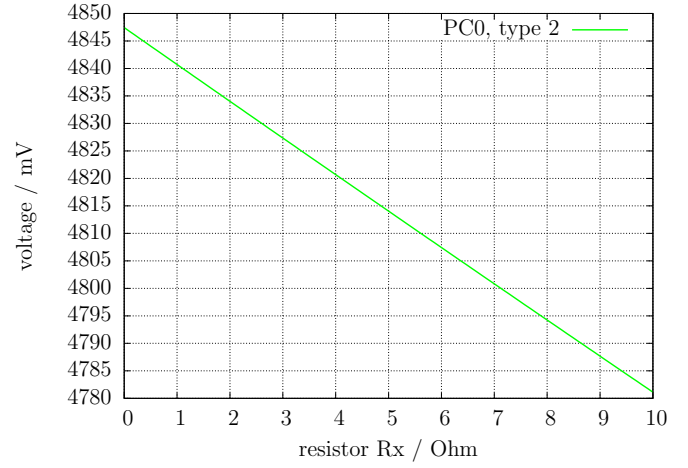
Figure 5.9: Voltages of type 1 and type 2 measurements with 680Ω

The graph of measurement type 1 is shown in figure 5.10a with zoomed scale for the lower resistor range. You can see, that you need a better ADC resolution than the standard 4.9mV resolution at the 5V ADC reference, to get the right resistor value from measured voltage below 2Ω . There are only three ADC steps from 0Ω to 2Ω . The range switching with the AUTOSCALE_ADC option can help in this case. The same zoomed range of measurement type 2 shows the figure 5.10b. Unfortunately we can not use the higher ADC resolution for measurement type 2 in this range, because the voltage is too high and our ATmega have no differential ADC input. Measurements with the 680Ω resistors are taken for building the result of measurements up to $20\text{k}\Omega$ (Voltage of measurement typ 2 will be below 169mV).

For higher resistor values the measurements with the $470\text{k}\Omega$ resistors are used. The mean value of both measurements is taken as displayed resistor value, if all tests attests, that is is no other type of part. If the AUTOSCALE_ADC function is selected and one of the voltages of the both measurement types is below 0.98V , a weighted average is build with factor four for this value. The other value is weighted with factor one. This is done to respect the factor four better resolution of this measurement. Factor four is only taken for ATmega88, ATmega168 and ATmega328 processors, for the ATmega8 two is taken as weighting factor if voltage is below 0.98V , because the reference voltage for the ADC is here 2.54V instead of 1.1V .



(a) Type 1 measurement



(b) Type 2 measurement

Figure 5.10: Cut-out of theoretical Voltage from 0Ω to 10Ω

5.2.2 Resistor Measurement with 470 kOhm resistors

The next figures 5.11 and 5.12 shows the same measurement procedure for the measurement with the precision $470k\Omega$ resistors. Because the $470k\Omega$ is very big in relation to the port resistor values 22Ω and 19Ω , the port resistor values are ignored for the computing of the resistor value R_x .

For both measurement types with the $470k\Omega$ resistors only one Voltage is measured, because the current is so low, that no voltage difference at the internal port resistors of the ATmega can be measured (as expected). The theoretical voltage gradient is shown in figure 5.13 where the resistor values are again shown in logarithmic scale. The theoretical gradient in this diagram ends at $100M\Omega$, but the resulting value of the Tester is limited to $60M\Omega$, otherwise the Tester assumes that no resistor is connected. The weighted average of both measurement types is taken as result with the same rules described for the measurements with the 680Ω resistors. For all ATmega processors I had found, that the measured results with the $470k\Omega$ resistors are more exactly, if a constant offset of 350Ω will be added. This offset can be adjusted with the `RH-OFFSET` define in the `config.h` file.

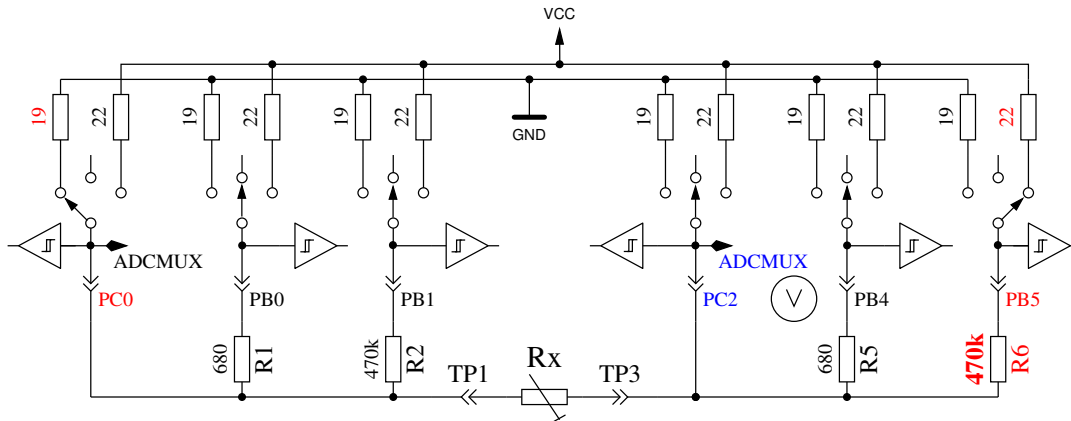


Figure 5.11: Measurement type 3 with $470k\Omega$

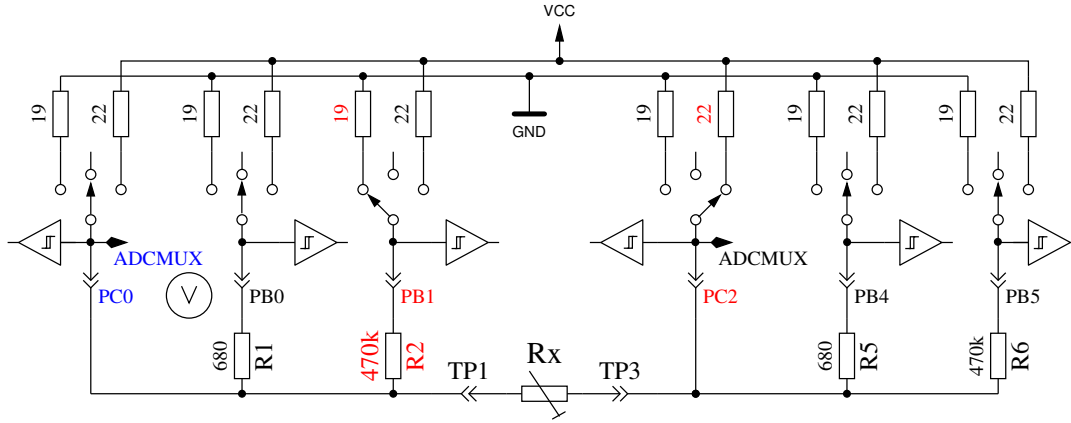


Figure 5.12: Measurement type 4 with 470kΩ

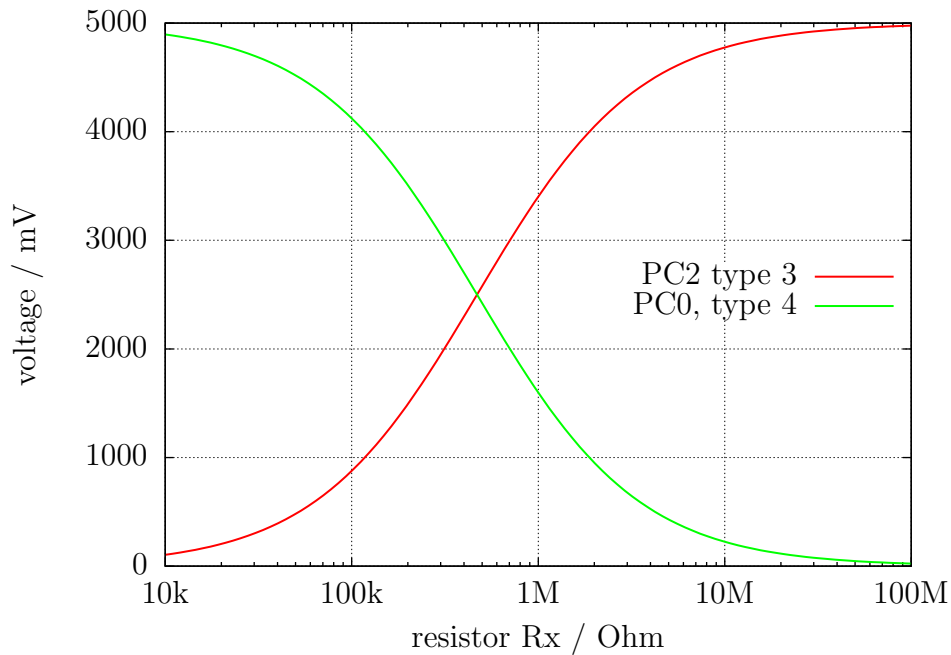


Figure 5.13: Voltages of type 3 and type 4 measurements with 470kΩ

5.2.3 Results of the resistor measurements

Figure 5.14 shows the relative errors of the resistor measurements with the ATmega8 microcontroller. Mega8 are the results without AUTOSCALE_ADC option, Mega8as are the results with the option AUTOSCALE_ADC and Mega8orig are some results of the original software of Markus F. without my improvements. You can see, that it is difficult to make the resistor measurement better with the AUTOSCALE_ADC mode of the ATmega8 processor. Figure 5.15 shows the same measurements with a ATmega168 microcontroller. Mega168 are the results without the AUTOSCALE_ADC option, Mega168as are the same measurements with the AUTOSCALE_ADC option. With the ATmega168 microcontroller it seems to be possible, that measurements of resistors in the range from 20Ω to 20MΩ can be measured with a tolerance of $\pm 1\%$. For Measurements below 100Ω you should keep in mind, that any measurement probe with wire have a resistance too. It is better to connect the resistor directly to the terminal pins. If this is not possible, subtract the resistance value of the shortened probe. For example, if your Resistor have a printed value of 30Ω, your tester shows a value of 30.6Ω and the two probes shortened have a value of 0.5Ω, then your resistor has been measured with 30.1Ω. Below a resistance value of 10Ω one resolution step results to a error of more than 1%!

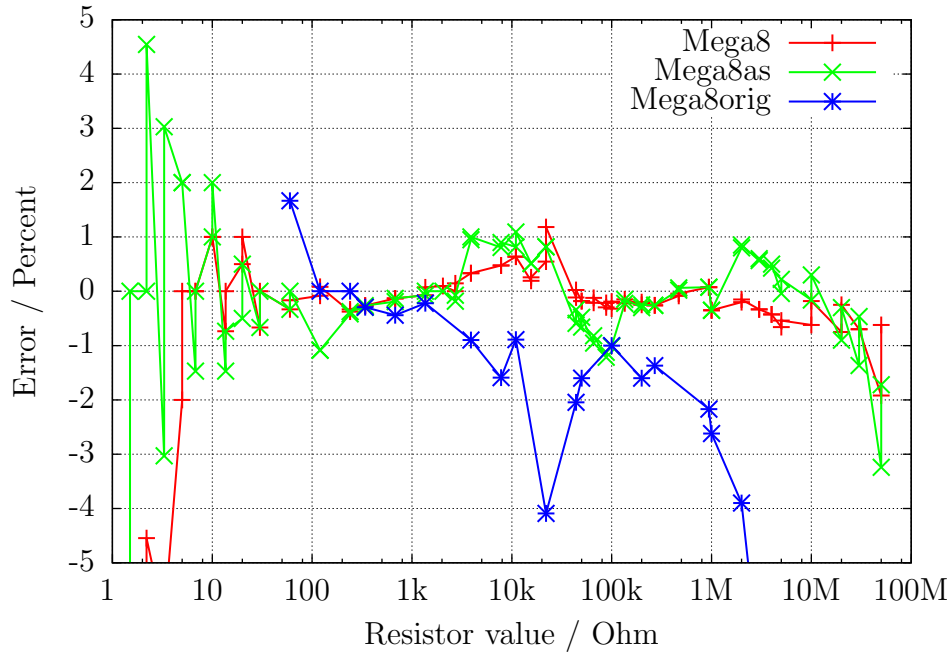


Figure 5.14: Relative error in % for resistor measurements with ATmega8

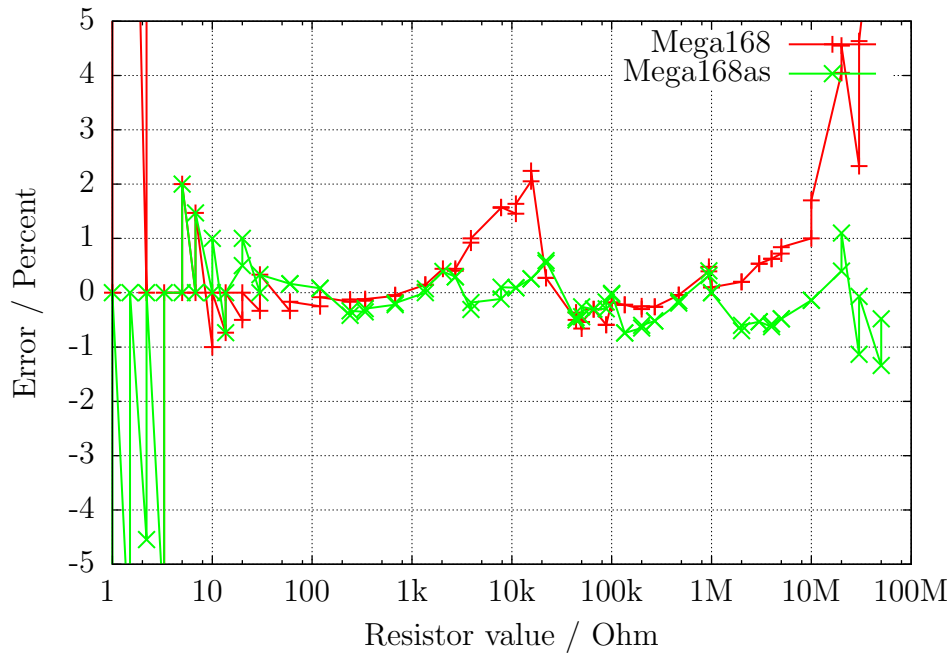


Figure 5.15: Relative error in % for resistor measurements with ATmega168

The figure 5.16a shows the measurement errors of three ATmega168 processors before calibration as points, after the calibration as line. The equivalent measurement errors of three ATmega168A processors are shown in figure 5.16b and the measurement errors of three ATmega168P processors are shown in figure 5.17. The measurement errors of three ATmega328 processors are shown in figure 5.18a and 5.18b. After the automatic calibration the relative measurement errors of resistors between $10\ \Omega - 20M\Omega$ usually are in the limit $\pm 1\%$. Only one measurement of a $22k\Omega$ resistor with the ATmega328P-13 shows a higher error. Before the calibration errors of some processors are found with $\pm 3\%$. This will be caused by the AUTOSCALE_ADC switching of the ADC reference. The direct compare of a capacitor voltage below 1 V, once measured with the VCC reference and another

once measured with the internal reference, can adjust this error. With this measurement condition the voltage is measured with the same multiplexor channel and the internal bandgap reference is connected to the AREF pin of the ATmega. Unfortunately the direct measurement of the bandgap reference with the special multiplexor channel results to this offset, which can be manually adjusted with the REF_R_KORR option or automatically with the AUTO_CAL option of the selftest. With the AUTO_CAL option the REF_R_KORR value is a additional offset to the automatic find out value!

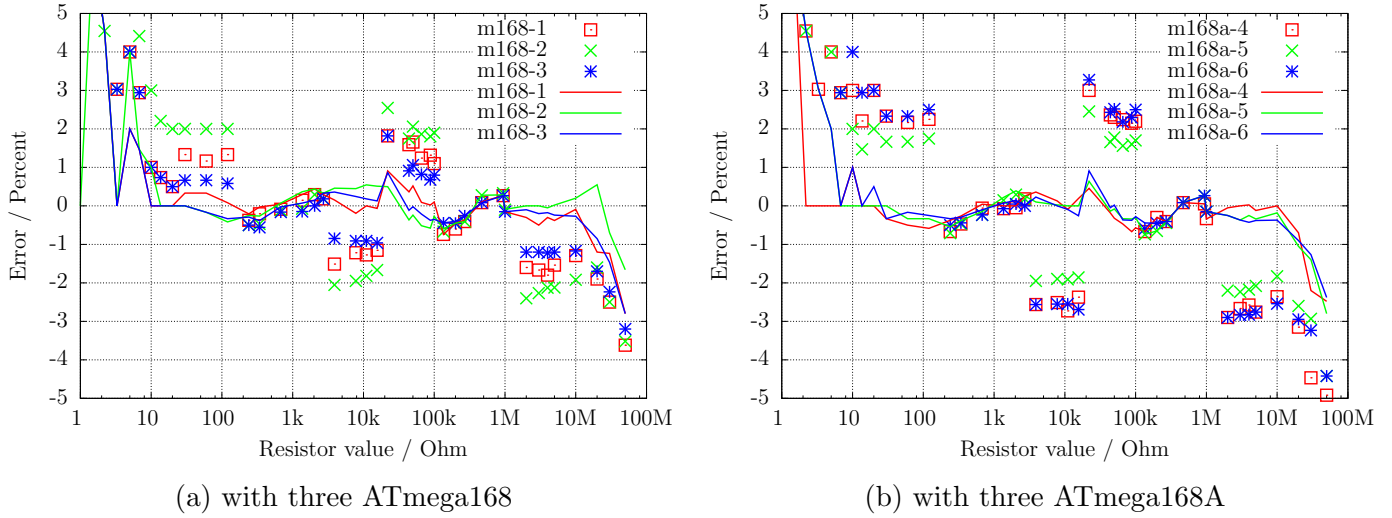


Figure 5.16: Relative error in % for resistor measurements

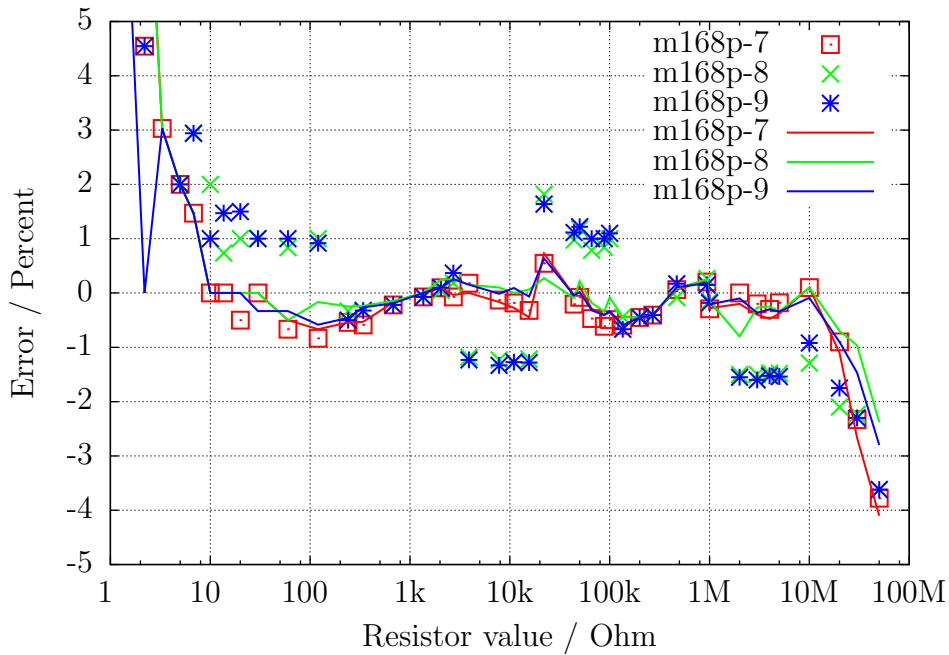


Figure 5.17: Relative error in % for resistor measurements with three ATmega168P



(a) with three ATmega328



(b) with three ATmega328P

Figure 5.18: Relative error in % for resistor measurements

5.3 Measurement of Capacitors

The measurement of capacitor values are done as separate task by measurement of load time after all other measurements. The original software of Markus F. did this with a program loop, which reads the corresponding digital input pin until a switch occurred and count the loop cycles. This has the handicap, that the resolution of time measurement is limited by the time consumption of one loop cycle. This usually was done in all six combinations for all three probe pins. The actual software uses two different ways to get the load time in only three combinations for the three probe pins. The positive side is now always the higher probe number. Only if capacity is measured parallel with a diode, the polarity can be in the other order.

5.3.1 Discharging of Capacitors

You should always discharge the capacitor before connecting it to the tester. The tester additionally discharge the capacitor before any measurement. If the voltage is below 1300mV, the capacitor is shortened by the output pins of the connected ADC port (Port C). I believe that this is legal because every output port has a built in resistance of about 20Ω . The data sheet Figure 149 (page 258) [2] shows voltage drop of output pins up to 2V. Of course I can not guaranty, that no damage can occur. I have tested the function with a 15mF Capacitor many times and I have never noticed any problem. The current should be below the specified limit of 40mA and is reduced fast by discharging. Of course damage can occur if you do not discharge a (high voltage) capacitor before connecting it to your tester.

5.3.2 Measurement of big Capacitors

One side of the capacitor is connected to GND. The other side of the capacitor is connected with the 680Ω resistor to VCC for a period of 10ms. Afterwards this probe pin is switched to Input (High Impedance). After this 10 ms current pulse the voltage of the capacitor is measured without any current. If the voltage has not reached a minimal value of 300mV, the load pulse is repeated up to 499 times. If after 127 pulses a minimum voltage of 75mV is not reached (about 2s), further load is stopped, because never the 300mV can be reached with the remaining load pulses. Figure 5.19 shows the three phases of measuring the capacity value of a capacitor. The value of the capacity is then computed with the count of load pulses and the reached load voltage from a table. The table contains the factors to get the capacity in nF units from load time and the reached voltage with a spacing of 25mV. Interim value of voltage will be interpolated.

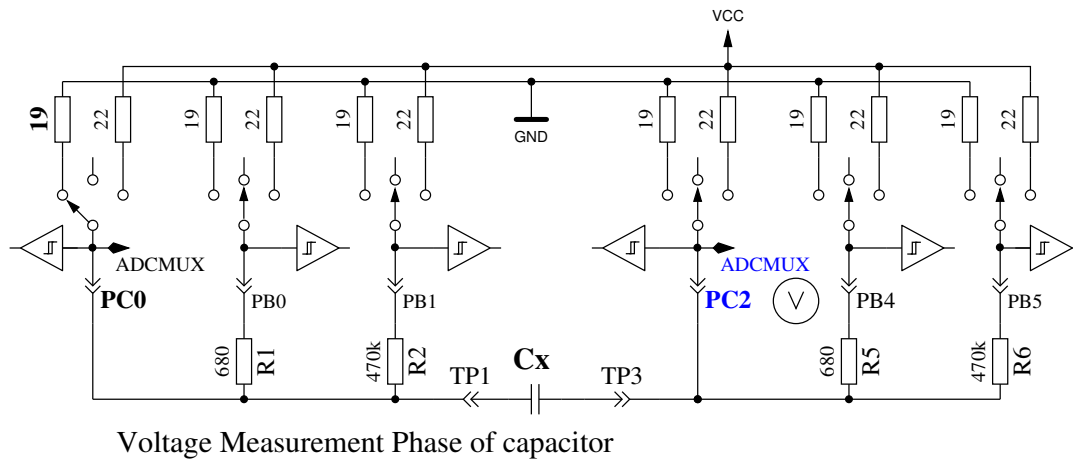
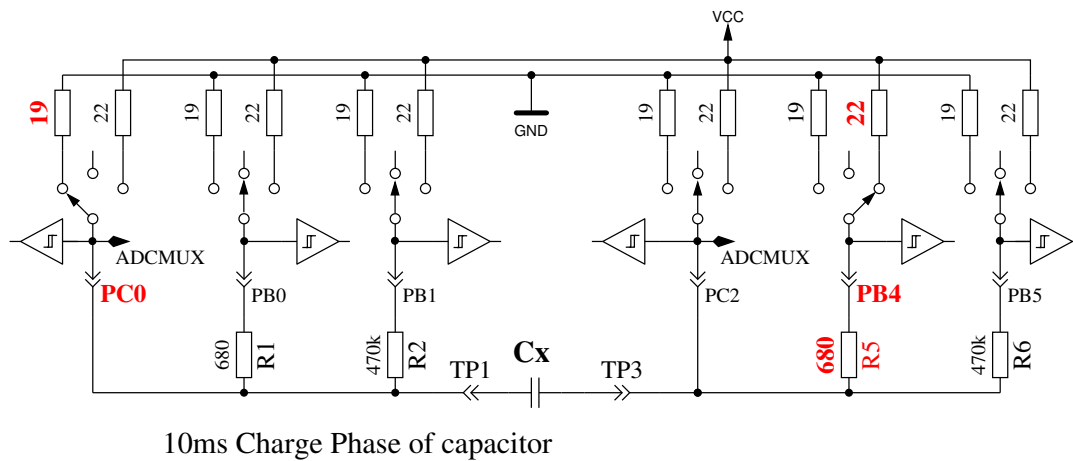
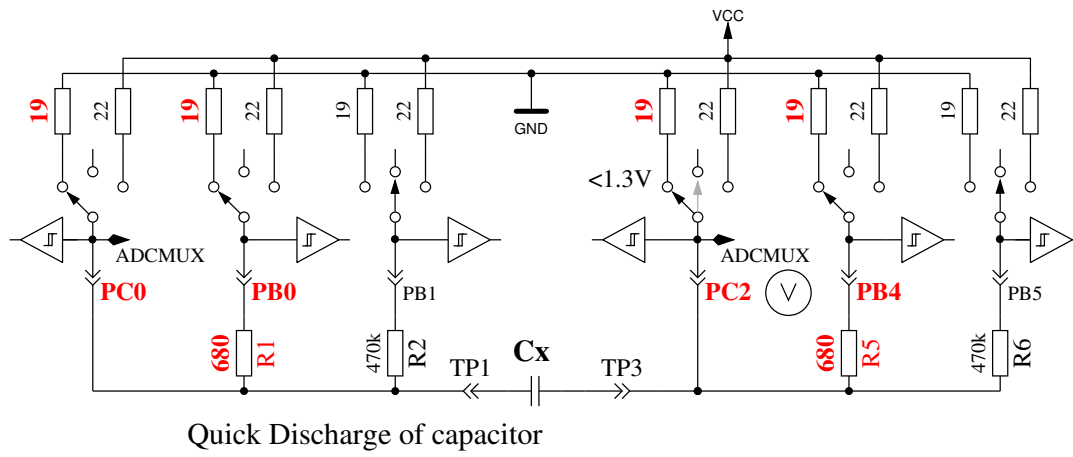
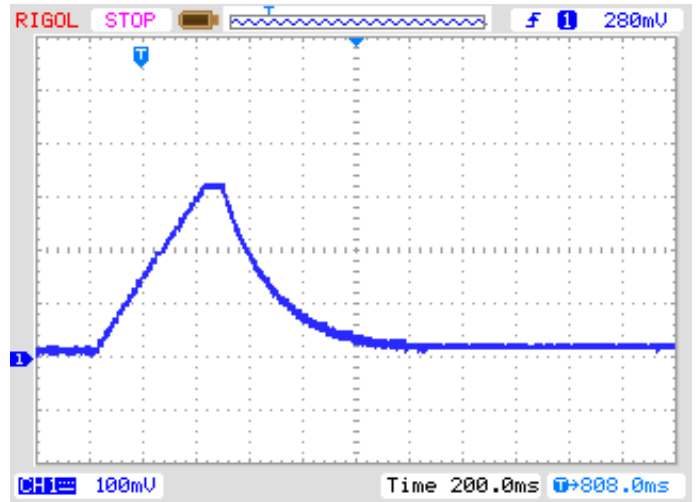


Figure 5.19: discharge a capacitor and load with 10ms load pulses until voltage reach a value of 300mV

As a result of the low load voltage, the measurement is much faster than the initial software version, because this advantage works also on discharging. So bigger capacitors can be measured. Furthermore a diode, which is parallel connected to the capacitor dont disturb the measurement in most cases, because the flux voltage of most diodes is not reached. Figure 5.20a shows the charge and discharge for a $229\mu F$ capacitor. The flat top of diagram from load end to discharge begin is caused by the measuring and computing time of the ATmega. Figure 5.20b shows the same measurement for a $5mF$ capacitor, notice how the time for measurement is grown to about 1.5 seconds inclusive the discharge. The last example shows the capacity measuring of a $15mF$ capacitor in Figure 5.21



(a) $229\mu F$ Capacitor



(b) $5mF$ Capacitor

Figure 5.20: Charge and discharge of big Capacitors for measuring

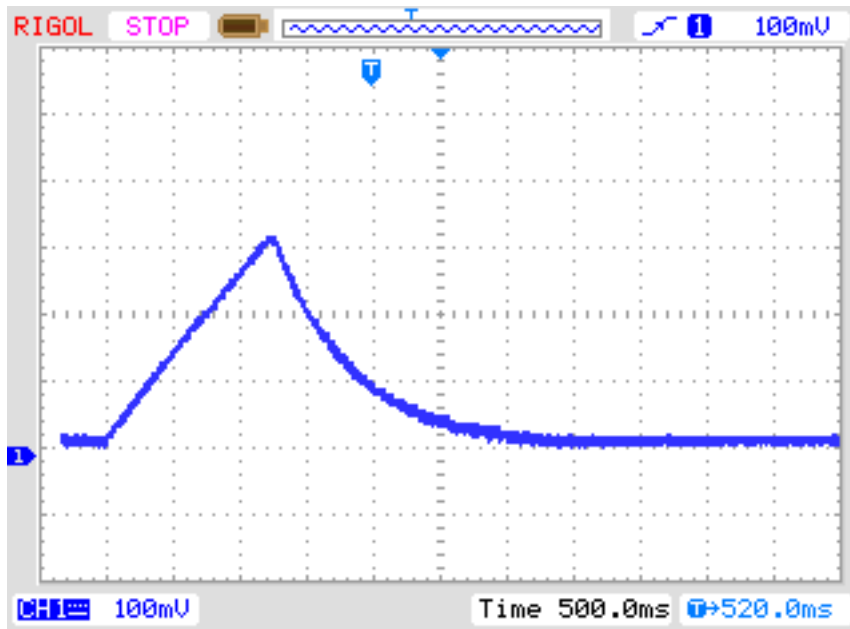


Figure 5.21: Charge and discharge of a $15mF$ Capacitor for measuring

5.3.3 Measurement of small Capacitors

If the first 10 ms load pulse has overloaded the capacitor, another technique of measurement is used. The ATmega processor has a built in 16-Bit counter, which can operate at the full clock rate (1MHz or 8MHz). This counter has also the feature to save his counter value by a external event. This event can be built by the output of the comparator. The comparator can operate with any ADC input pin and the band gap reference. Figure 5.22 shows a simplified diagram of the measurement situation. So I discharge the capacitor, prepare the comparator to the proper pin input, start the counter at 0 and start immediately the charging of the capacitor with one side connected to GND and the other side connected with the $470k\Omega$ resistor to VCC. Now I check within a program loop, if the counter flags signals a overflow event or a input capture (external) event. I count the overflow events until I detect the input capture event. In this case I stop the counter and check if I must count a additional overflow, because the counter can't be stopped by the input capture event.



Figure 5.23: Charge and discharge of a $22\mu F$ Capacitor for measuring

In principle this technique of measurement can also be done with the 680Ω resistor, but because the ADC can't be used if the comparator is working, I have no chance to monitor the load voltage until the comparator is stopped. If a undetected diode is parallel connected with the capacitor, the load current of the capacitor can be absorbed by the diode (threshold voltage) and the band-gap voltage will never be reached. The method taken in actual software for big capacitors in section 5.3.2 avoids this conceptual bug.

5.3.4 Results of Capacitor measurement

The results of my measurements is shown in figure 5.24 for ATmega8 processor without and with the AUTOSCALE_ADC option. Additionally some values of original software are shown with a correction factor of 0.88 (-12%). The results of the measurement of the same capacitors for a ATmega168 is shown in figure 5.25. The base for the error computing are the measurement results of a PeakTech 2170 RCL-meter, not the printed value of the parts. A part of the relative high measurement difference is caused by the too high measurement frequency of the RCL-meter for big electrolytical capacitors. On the other side the bad quality factor of the electrolytical capacitors may cause another percentage.

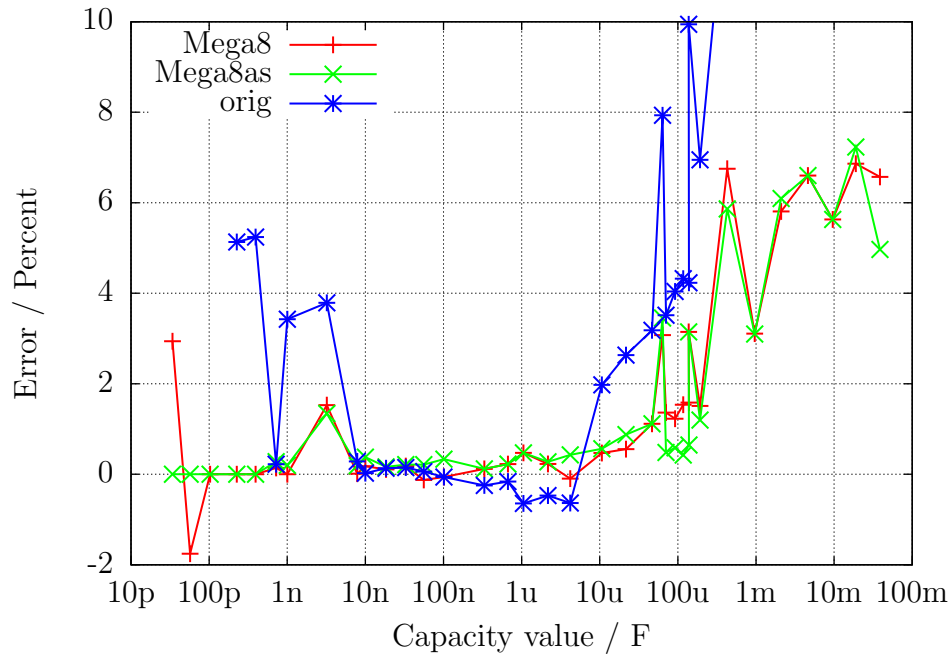


Figure 5.24: Error in % for capacitor measurements with ATmega8

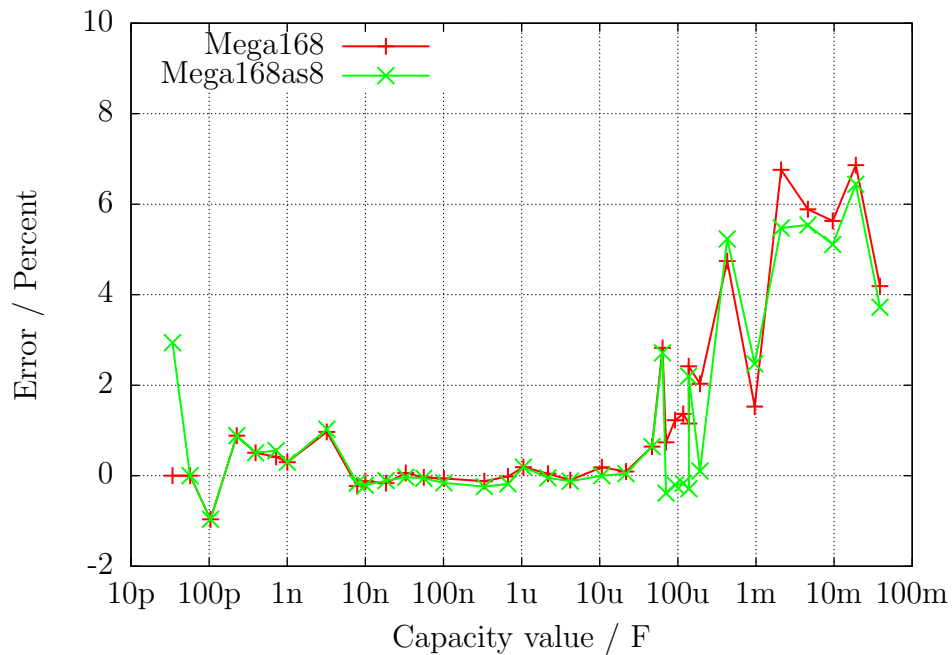


Figure 5.25: Error in % for capacitor measurements with ATmega168

Figure 5.26 illustrates, how difficult is it to choose the right base for the capacity measurement. All measurement results are compared with the best estimated value of the capacitors. The gradient „Multimeter” shows the differences of the Peaktech 3315 Multimeter results. The next gradient „LCR” shows the differences of the Peaktech 2170 LCR-Meter results, which is taken from best frequency approach. To compare this results to the results of a ATmega168 equipped Transistor-Tester the gradient „ATmega168as” is also shown. I beleave, that this errors are not real measurement errors of the particular equipment, because my best estimated value are also not the real capacity value of the capacitors.

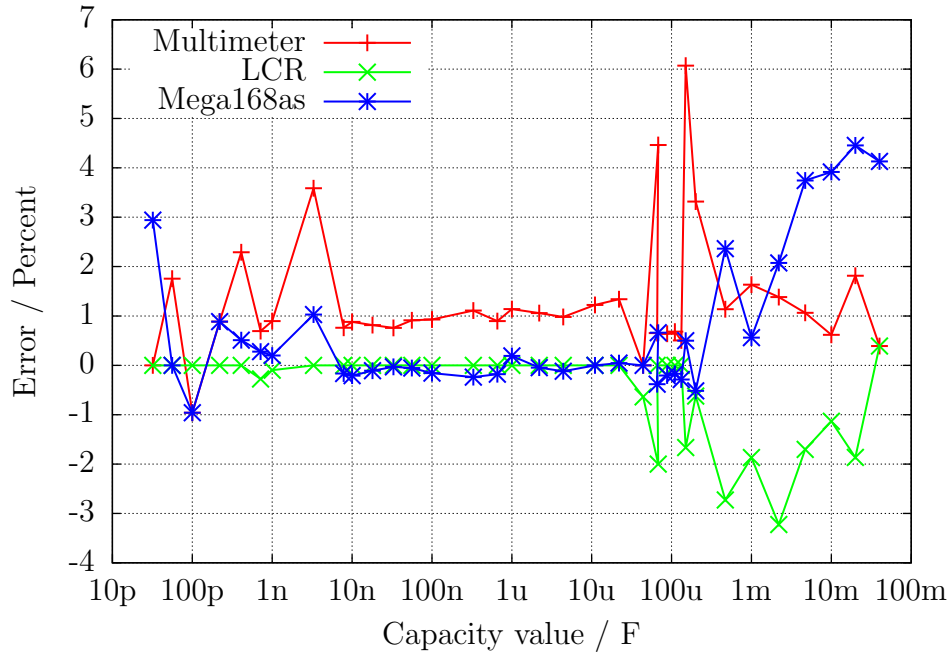
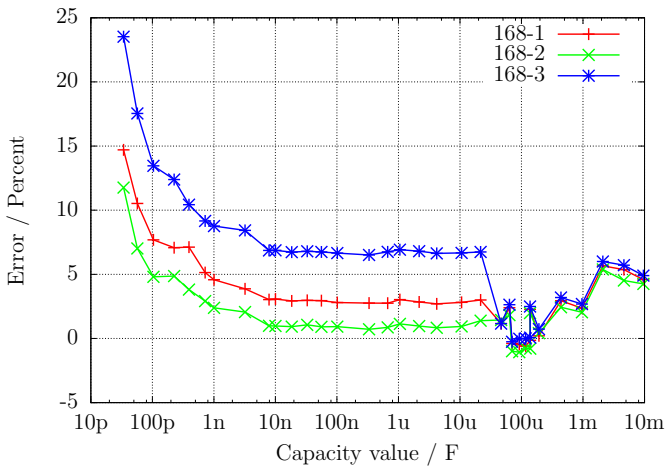
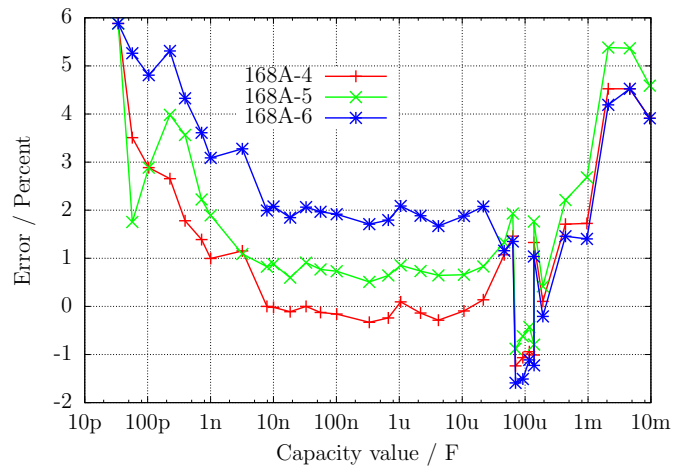


Figure 5.26: Comparison of capacity measurement results of Multimeter, LCR-meter and ATmega168

The differences of measurements of three different ATmega168 processors are shown in figure 5.27a . In this case the results of the LCR meter is taken as base of comparison. The same results of three different ATmega168A processors are shown in figure 5.27b and three different ATmega168PA processors are shown in figure 5.28. The results of three ATmega328 are additionally shown in figure 5.29a and the results from three ATmega328P are shown in figure 5.29b. At this only the zero value of the capacity measurement of 39pF is respected, all other facility to correct the results are not used. This zero value includes the 2-3pF, which are caused by the 12 cm long cable with the clips. The board layout can cause a different zero value, I have fixed this zero value with the board "DG2BRS V 5.2.1".



(a) three ATmega168



(b) three ATmega168A

Figure 5.27: capacity measurement error, not calibrated

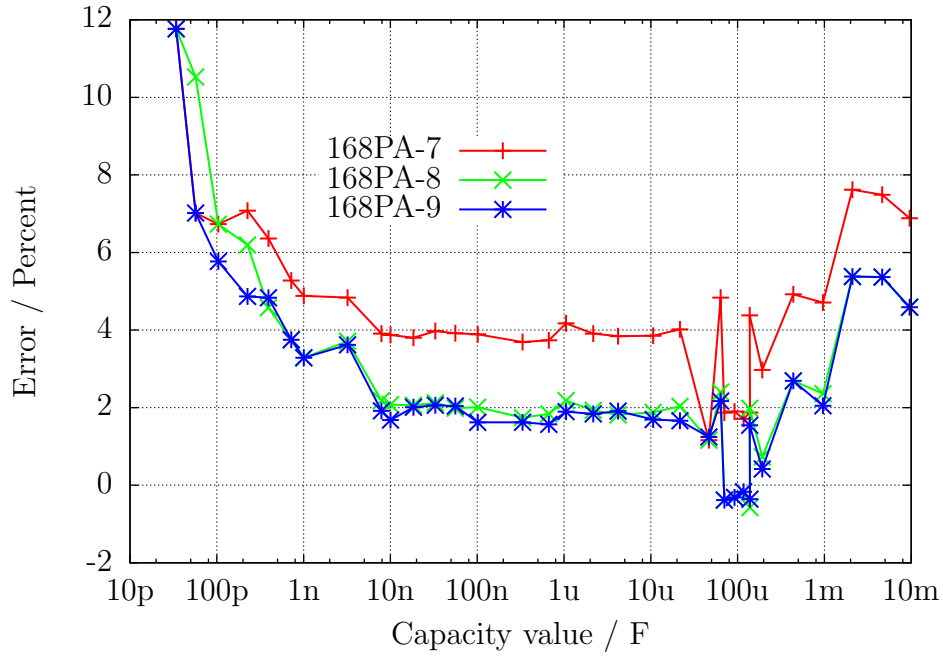
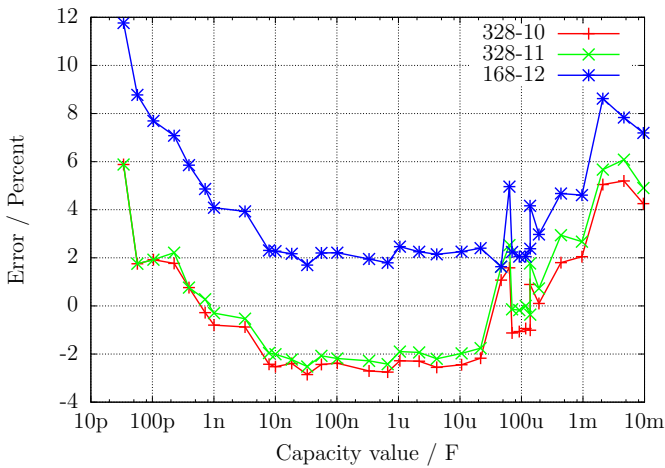
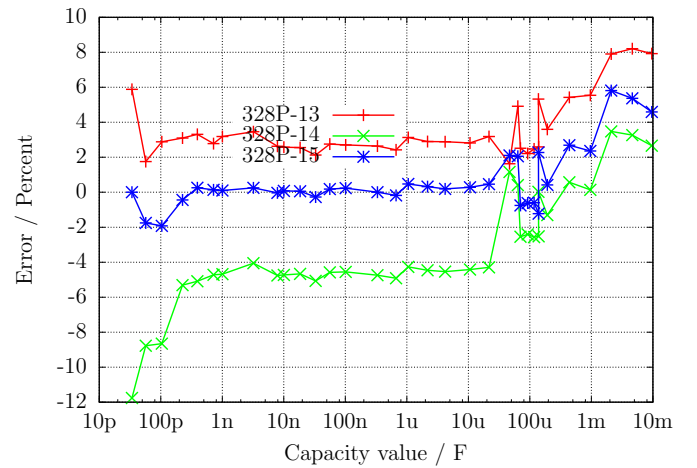


Figure 5.28: capacity measurement error of three ATmega168PA, not calibrated



(a) three ATmega328



(b) three ATmega328P

Figure 5.29: capacity measurement error, not calibrated

To get the best accuracy you must adapt the software to the individual characteristic of your ATmega exemplar. For this you can set a correction voltage `REF_C_KORR` for the comparator, which will be used for measurement of little capacity values. A correction of 1 mV will reduce the measurement results to 0.11 % . For big capacity values you can specify with the per mill value `C_H_KORR`, how much your capacity values are measured too big. Because the capacitors with big values are most electrolytic capacitors with worse quality factor, the measurement of the capacity value is difficult. So it is also extra difficult to get the difference to the real value of a capacitor.

Especially with the ATmega168 processors I have noticed a anomaly of measurement results of little capacity values, which depend on the slew rate of the voltage during loading of the capacitor. Figure 5.30 shows the error of the capacity measurement when only the zero value is respected (168-3-A), with correction factor for little capacitors `REF_C_KORR`=66 as well as the correction factor for big capacitors `C_H_KORR`=5 (168-3-B), plus additional as gradient 168-3-C with a model of the slew rate dependency of little capacitor measurements (`COMP_SLEW1`=4000 und `COMP_SLEW2`=220).

Also the self-discharge of big capacitors is respected with gradient 168-3-C. The component with the slow rate dependent value is computed with $\frac{COMP_SLEW1}{cval+COMP_SLEW2} - \frac{COMP_SLEW1}{COMP_SLEW2}$, where cval is the measured capacity value with pF units.

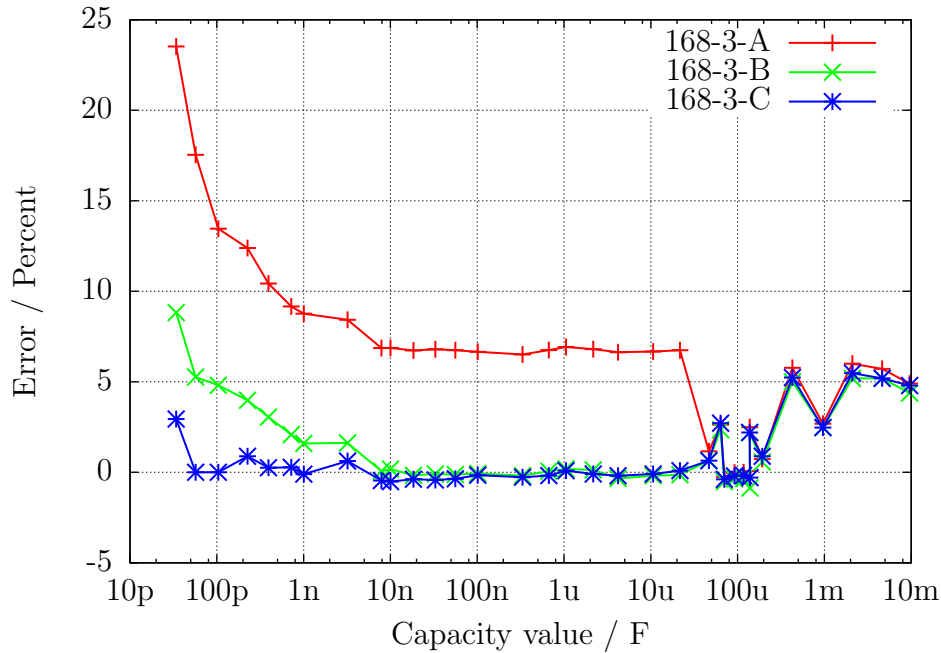


Figure 5.30: Improvement of the capacitor measurement of one ATmega168

5.3.5 Automatic calibration of the capacitor measurement

The automatic calibration is build in two parts. The first part find out the zero offset of the capacity measurement. For that the mean value of the capacity measured without connected capacitor is build. A mean value for all 6 measurement combinations is build with 8 repetitions. After successfull determination the zero offsets are written to the EEprom and will be used for further measurements. More difficult was the clearance of the variance of the different ATmega processors for little capacitors ($40\mu F$), which is shown in Figure 5.27a, 5.27b and 5.28. As a significant reason for this is found the different characteristic (Offset voltage) of the analog comparator.

The date of measurement of nine different processors is shown in figure 5.31 . The "diff2ref" points show the difference of the voltage of a loaded capacitor of $660nF$ to the individual internal reference voltages (band gap). Ideally this difference Voltage should be zero, if the analog comparator has stopped the loading by the signal to the processor. The short handling time of the processor should not result to a measurably rising of the capacitor voltage of this relative big capacitor. The "CapErr" points show the estimated measurement errors of each processor out of figure 5.27a, 5.27b and 5.28 with per mill units. It is noticeable, how the "CapErr" points will follow the "diff2ref" points. Therefore the "diff" points show the difference between the particular "CapErr" and "diff2ref" points. With a mean value of the "diff" points we can get a good estimation for the correction of the capacitor measurements together with the difference voltage of the loaded capacitor and the internal reference. For the second part of adjustment you must connect a capacitor to pin 1 and pin 3. This capacitor should have a good quality factor and should have a capacity of at least $100nF$. It should be a film capacitor, as far as possible not a ceramic capacitor und in no case a electrolytic capacitor. The value of this capacitor can be up to $22\mu F$, but you don't need to know the exact value.

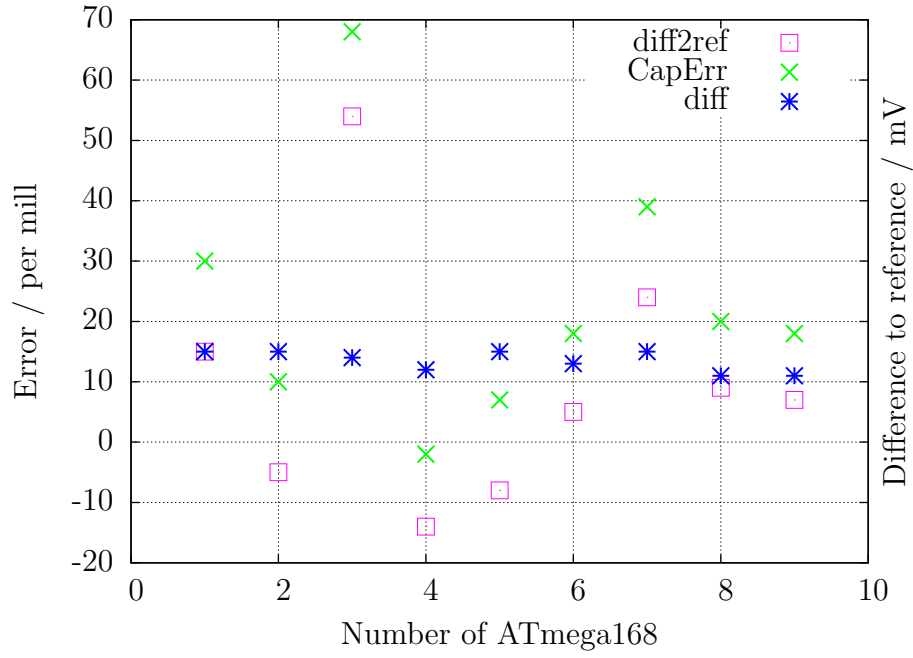
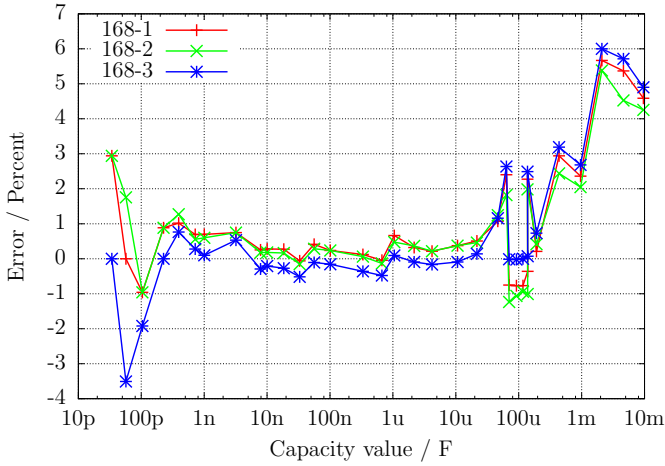
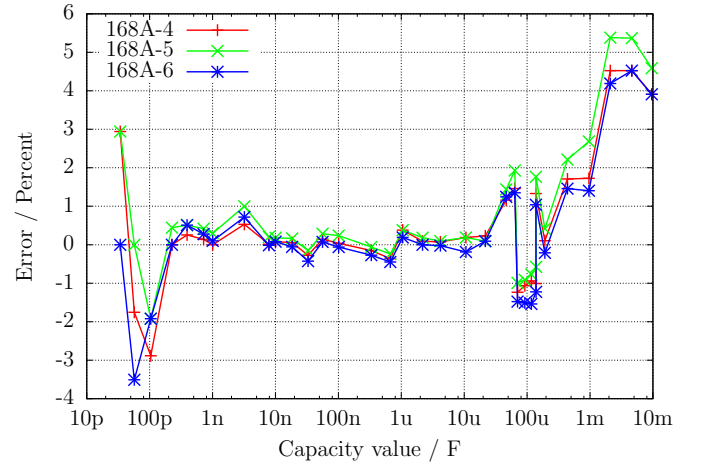


Figure 5.31: Date of nine ATmega168 processors

The figures 5.32a, 5.32b, 5.33, 5.34a and 5.34b shows the measurement results of the different processors with a standard software after the auto calibration. The flash of the processors was loaded with the same software, only for the ATmega168PA the Makefile must be adapted with "PARTNO = m168p". After loading the data the selftest was started for each ATmega and a capacitor with $330nF$ was connected during test No. 10 to pin 1 and pin 3.



(a) three ATmega168



(b) three ATmega168A

Figure 5.32: capacity measurement error, calibrated

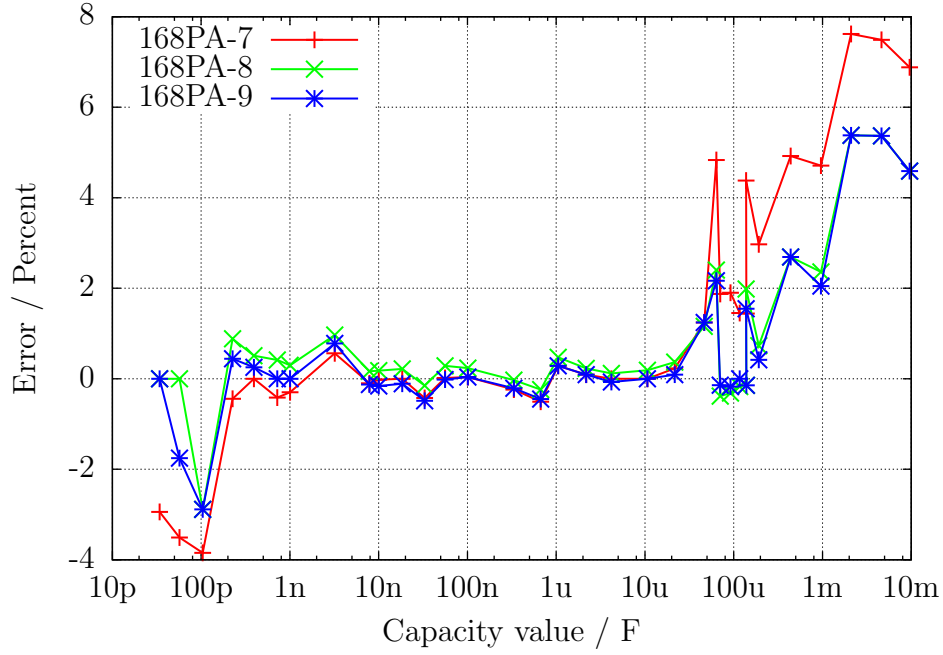
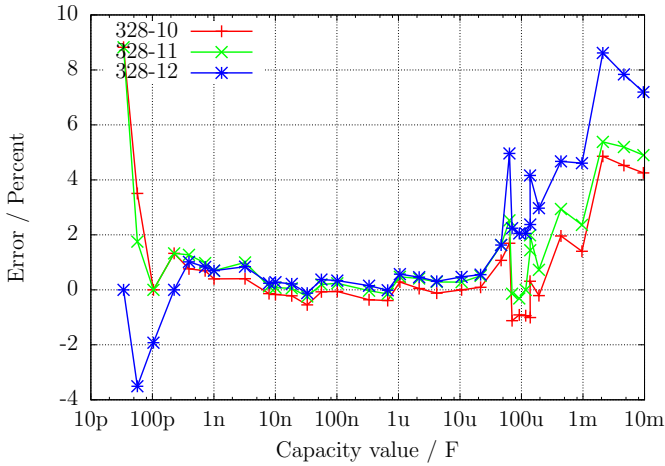
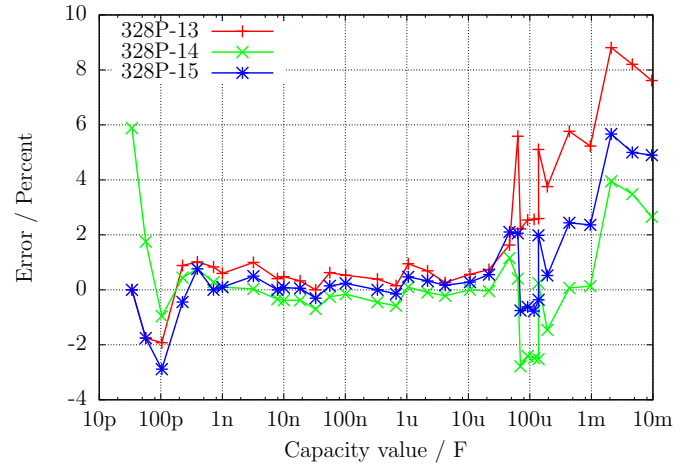


Figure 5.33: capacity measurement error of three ATmega168PA, calibrated



(a) three ATmega328



(b) three ATmega328P

Figure 5.34: capacity measurement error, calibrated

At last I will make more clear the effect of the AUTO_CAL option in the selftest program. The following figure 5.35 shows the results from the three ATmega processors with the biggest error of measurement, one measurement before the calibration and another measurement after the calibration. The points marked with the ending "unc" shows the the errors without calibration. The lines with the ending "cal" shows the error results of the **same processors** with the **same software** after the calibration in the selftest section. The reason for the measurement errors for big capacitors ($>40\mu F$) is not yet known. All used capacitors for this series of measurements are film capacitors or ceramic capacitors ($56pF$, $100pF$ and $3.3nF$), no electrolytical capacitors are used.

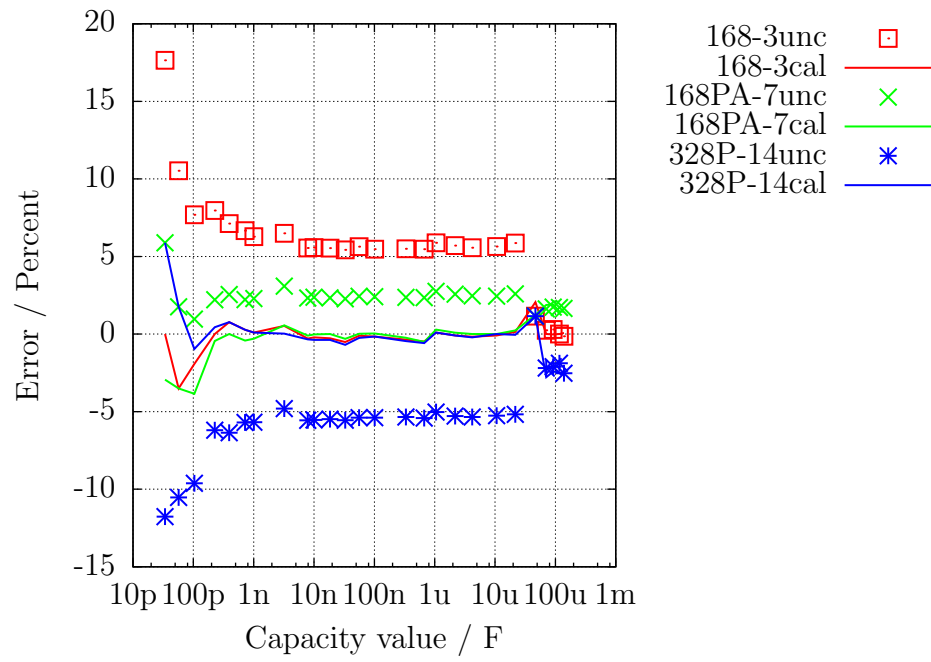


Figure 5.35: Error of capacitor measurement of three ATmega, before and after the calibration

5.4 Selftest Function

Beginning with release 0.9k I have implemented a self test function. Usage is very simple. If you have installed test terminal with clamps, put all clamps together to a piece of uninsulated wire and press the start button. The program notice the shorten probes and start the self test function. After finishing the self test the transistor tester will continue with normal measurement. If no equipment is connected, the program will end with “part unknown or damaged”. The unhappy side of the self test function is that the 8K flash is used near the limit. The length of the ATmega8 version 0.9k is about 8000 bytes. You can configure self test only together with all other options for a ATmega168 or ATmega328. For the ATmega8 you must at least omit the option AUTOSCALE_ADC because of the limited flash memory. The separate steps of the self test function is generally displayed on row 1 of the LCD display with the letter T followed by the step number. Every step is repeated 8 times, before the program continues with the next step. But if you hold the start key pressed, when the test is finished, this test is not repeated any more. If you leave the key pressed the total time, every test is executed only once.

Without the AUTO_CAL option only measurement results are displayed in evvery step, no error analysis are done, you must interpret the results yourself. At this place I will give you an additional important hint. Never do a measurement with connected ISP plug! The ISP interface influences the measurement. Here is the list of currently implemented tests:

1. **Measurement of the 1.3V (or 1.1V) reference Voltage (band gap Reference).** In row 1 the text “Ref=” and the measured Voltage in mV is displayed. For the ATmega8 the voltage should be near to 1.3V. For the other processors the voltage should be near to 1.1V. The second row shows the resulting factor for the capacity measurement with the 470kΩ resistor.
2. **Comparing of the 680Ω resistors.** In row 1 the cryptic text “+RL- 12 13 23” is shown. Meaning of this is as follows: The RL is the short form of Resistor Low meaning the 680Ω resistors. The 12 stand for: resistor at pin 1 is connected to VCC (+) and resistor at pin 2 is connected to GND (-). The result of this measurement is displayed in row 2 at the first place as difference to the theoretical value. In row 1 follows now a “13” which means, that the first connection of measurement 1 is still connected with 680Ω to VCC but that the resistor of pin 3 is connected to GND. The result is displayed in the middle place of row 2 as difference to the theoretical value. The last measurement of this test “23.” means that now the resistor at pin 2 is connected to VCC (+) and the resistor of pin 3 is connected to GND. The result of measurement is displayed at the last place of LCR row 2 as difference to the theoretical value. Please remember, that the resolution of the ADC is about 4.88mV! The measurement situation is also shown in figure 5.36. The theoretical value with respect to the internal resistance of the pins should be: $\frac{5001 \cdot (19+680)}{(19+680+680+22)} = 2493$.

measurements as those in test 7 to the other side as you can see in Figure 5.38. With the following steps the resistance is computed: To get the current, the following is computed: $(VCC - (resultoftest7) - (resultoftest8))/680$. To get both resistor values, the voltage (result of test 7 or 8) is divided by this current. The result for this test will then be notified in row 1, the resistance value to the GND side is displayed in row 2. Both results will be written to the EEprom memory and will be used for further measurement.

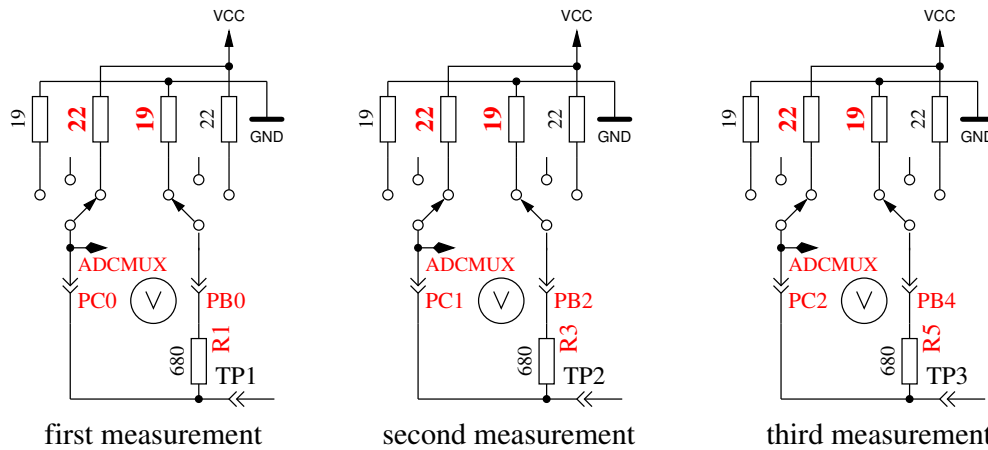


Figure 5.38: Measurement of internal resistance of Port C switched to VCC

9. **Measurement of the zero offset of the capacitor measurement.** The zero offset for the capacity measurement with pin combinations 1:3, 2:3 and 1:2 is shown in that order. All three values are shown in pF units. For this measurements no predefined zero offset is respected. The results will be written to the EEprom, if all values are less than $70pF$. This will be notified by the output of "C0" in row 1, followed by the results of the three zero offsets for pin combinations 1:3, 2:3 and 1:2. Please notice, that changes of the test equipment can cause a new adjustment of the zero offset. If you use wire with clips, the zero offset may be 3 pF greater compared to a empty socket.
10. **Wait for the connection of a capacitor to pin 1 and pin 3.** The message "1-C-3 >100nF" is shown in row 1 of LCD. To prepare the measurement of the offset voltage of the analog comparator you must connect a sufficient big capacitor to pin 1 and pin 3. It should be a capacitor with a high quality factor and a capacity of at least $100nF$. Capacitors with up to $22\mu F$ can be used without problems. You should never use electrolytical capacitors, use film capacitors instead.
11. **Measurement of the comparator offset for capacitor measurement adjustment.** To get the offset of the analog comparator, a capacitor must already be connected to pin 1 and pin 3. The capacitor will be needed for buffering the load voltage of a capacitor, in order to get the voltage difference of load voltage to the internal reference voltage (band gap). If measurement is successfull, the correction value is short shown with the text "REF_C=" in row 2 of the LCD and written to the EEprom. You can give a additional offset to the automatic measured value with the REF_C_KORR option. If you have selected the AUTOSCALE_ADC option, the gain of the ADC readings with the internal reference will be adjusted by comparing a capacitor voltage below 1 V once readed with VCC reference and once readed with the internal reference. Your REF_R_KORR value is a additional offset to the automatic find out difference value.

At the end of test function the text "Auto Test End" is shown in row 1 and the version number of software is shown in row 2. If selected with the Makefile option FREQUENCY_50HZ, a **50Hz rectangle signal** is generated on pin 2 and the same signal in opposite direction on pin 3. Pin 1 is

switched to GND . The current is limited with 680Ω resistors. This will be notified by the Output of “50Hz” at the end of row 1 of the LCD display. The 50Hz signal will be generated 30 times for 2 seconds each. You can check the time of the wait calls, if you have an oscilloscope or frequency counter. If you don’t use the crystal clock version, the result may be inexact. A exactly clock frequency and wait time are important for measurement of capacity values. You can abort the generation of the 50Hz signal by long time pressing of the start button. Then the program continues with the normal measurement task.

Test No.	1. Result	2. Result	3. Result
Test 1 Reference	band gap Ref 1228 RHfakt 756		
Test 2 comparison 680Ω best value: 0	RL1+ RL2- -4	RL1+ RL3- -5	RL2+ RL3- -5
Test 3 comparison $470k\Omega$ best value: 0	RH1+ RH2- -7	RH1+ RH3- -7	RH2+ RH3- -7
test 4	isolate probe		
test 5 $470k\Omega$ isolation best value: 0	RH1- 0	RH2- 0	RH3- 0
Test 6 $470k\Omega$ isolation best value: 0	RH1+ -5	RH2+ -6	RH3+ -6
Test 7 Pin resistance Low	TP1- RL1+ 132	TP2- RL2+ 132	TP3- RL3+ 137
Test 8 Pin resistance High	TP1+ RL1- 151	TP2+ RL2- 151	TP3+ RL3- 151
Test 9 Capacitor offset	C0 1:3 34	C0 2:3 35	C0 1:2 33

Table 5.5: Selftest Mega8 @ 8MHz, Signature 1E9307, WITH_AUTOREF

The results of the selftests of nine different ATmega168 processors and of six ATmega328 processors will be shown in the following figures.

Test No.	measurement typ	theoretical	figure
Test 1	band gap Ref	1100	5.39
Test 2	RL-Mean	0	5.40a
Test 3	RH-Mean	0	5.40b
Test 5	RH-Low	0	5.41a
Test 6	RH-High	0	5.41b
Test 7	R out Lo	131	5.42a
Test 8	R out Hi	151	5.42b
Test 9	Cap zero offset	30	5.43
Test 11	Reference correction	0	5.44

Table 5.6: Table of the selftest figures

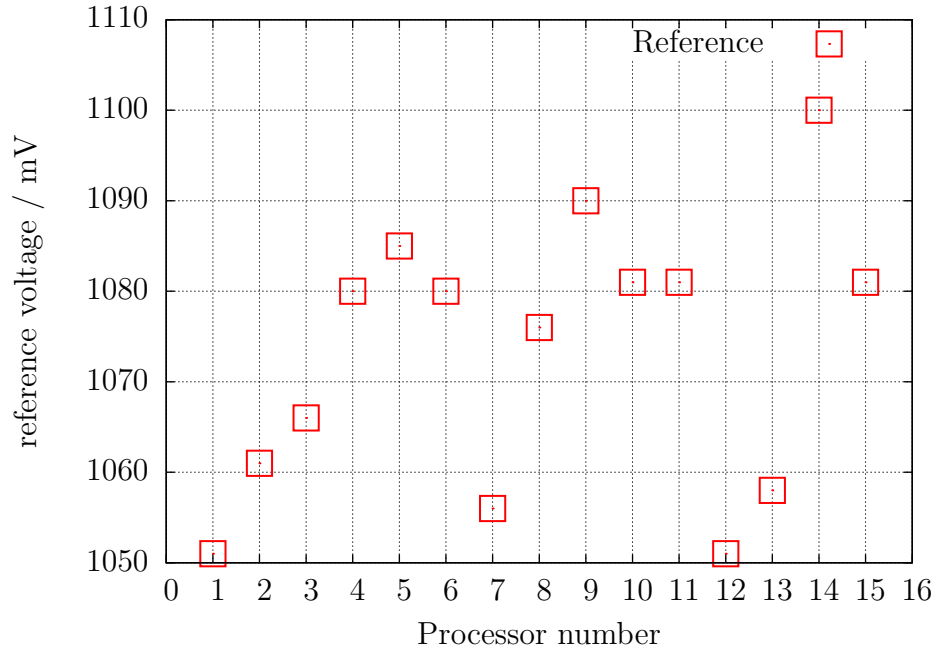
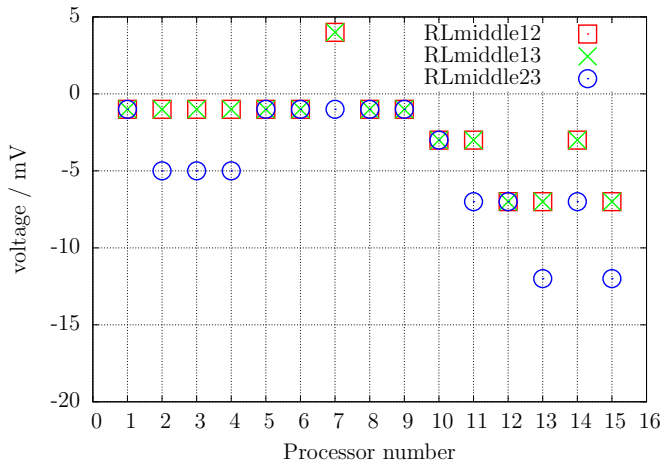
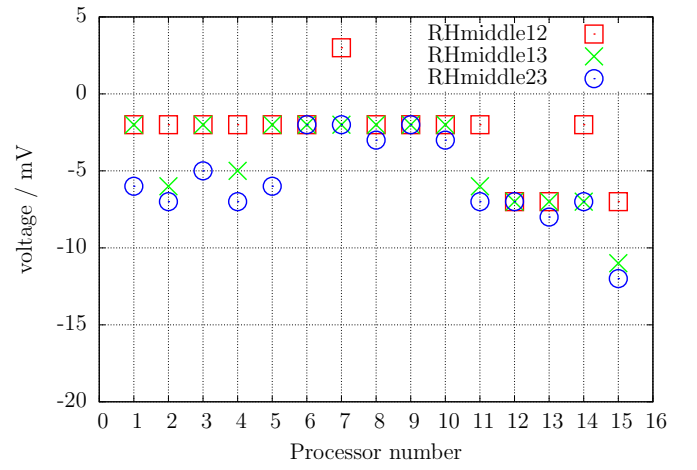


Figure 5.39: Selftest: Reference-Voltages

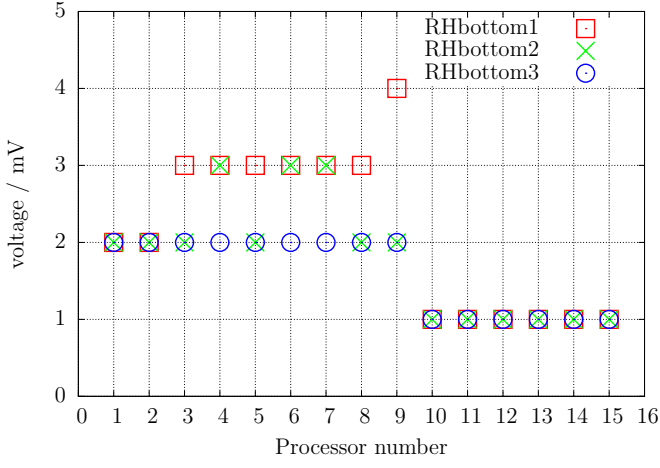


(a) with 680Ω

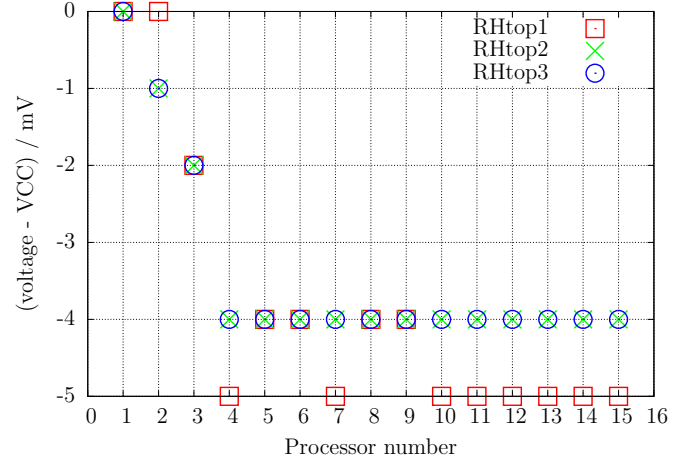


(b) with 470kΩ

Figure 5.40: Selftest: difference to ideal mean voltage

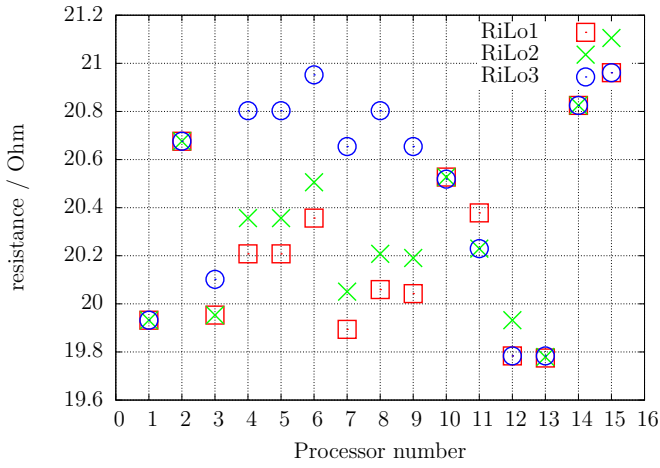


(a) with $470k\Omega$ to 0V

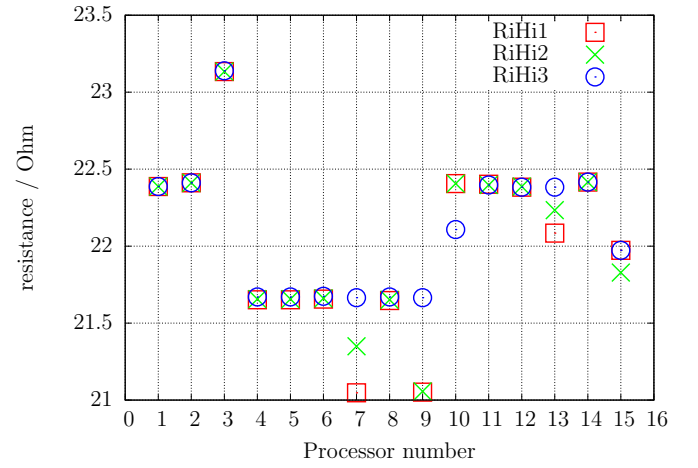


(b) with $470k\Omega$ to 5V

Figure 5.41: Selftest: Input voltage



(a) with 680Ω to 5V



(b) with 680Ω to 0V

Figure 5.42: Selftest: Output resistance

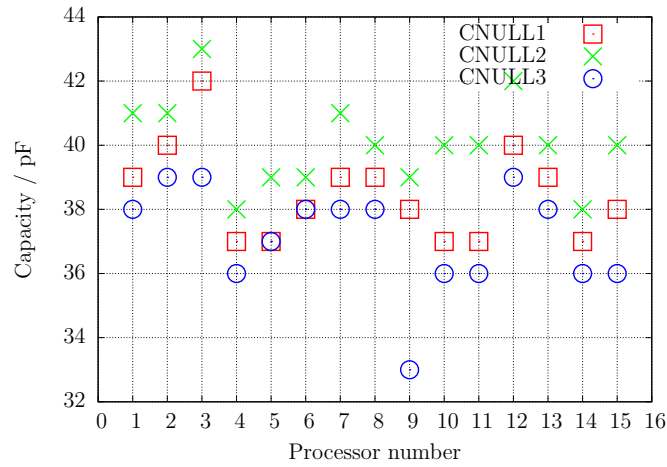


Figure 5.43: Selftest: zero offset of the capacity measurement

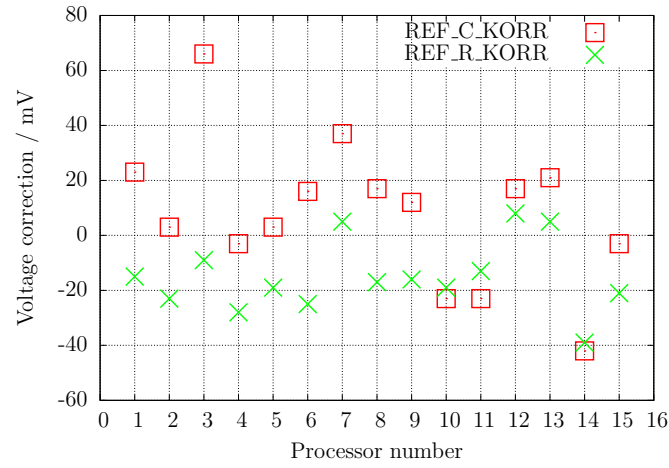


Figure 5.44: Selftest: correction values after automatic calibration

At last I would like to show you the difference voltages of the external at the AREF pin with a multimeter measured voltages and the internal with the ADC measured voltages of the reference voltages of 15 different ATmega processors and the found correction voltages (REF_R_KORR) after the automatic calibration in figure 5.45. You can see, that the automatic calibration values nearly follow the external measured values.

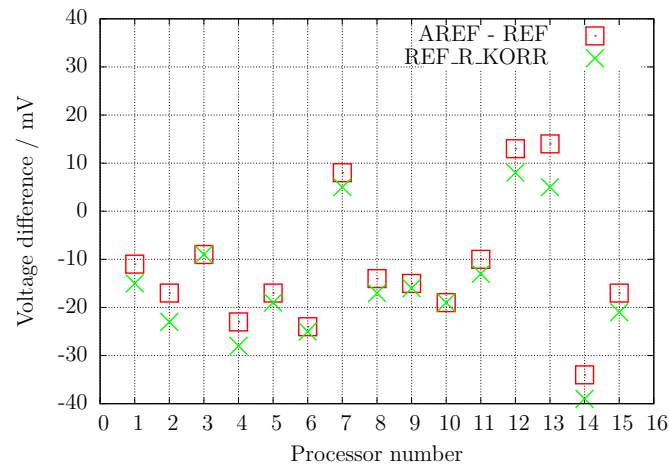


Figure 5.45: Selftest: Voltage difference of the internal reference

Chapter 6

Known errors and unsolved problems

Software Version 0.99k

1. Germanium Diodes (AC128) are not detected in all cases.
2. Capacitors with values of more than 40mF are detected as resistors with a value of 2Ω . Because resistor is detected first, capacity measurement is not done. More practical capacitors normally have values of 10mF or below. So if this problem can not be solved, capacity measurement can also be limited to 40mF instead of the aspired 100mF top value.
3. Discharging of capacitors does not end regular. One user had reported to me, that a ATmega8 processor could not find the end of discharge in the AUTOSCALE_ADC mode. The voltage limit for discharged capacitor ist set very low to below 3mV. My own tests with different processors are all well. The appearance of this error is, that the tester wrote after a long time of about 12s the text “Cell!” to the LCD. You can adjust the CAP_EMPTY_LEVEL in the Makefile to a higher level of 3mV, if you can not find the reason!
4. Power Schottky Diodes such as MBR20100CT are not detected correctly as dual diodes.

Chapter 7

Special Software Parts

Several modifications are done to save flash memory. The LCD-Output of probe-pin numbers was done in the form `"lcd_data('1'+pin)"`. To save the add operation for every call, the entry `"lcd_ziff1(uint8_t pin)"` was added to the `lcd_routines.c`.

The pseudo calls in the form `"_delay_ms(200)"` are not implemented as library calls, but wait loops are implemented for every call. This will consume much memory, if you have many calls at different location in your program. All of this pseudo calls are replaced with calls to my special assembly written library, which uses only 74 bytes of flash memory (@8MHz), but enables calls from `wait1us()` to `wait5s()` in steps of 1,2,3,4,5,10,20... The routines include the Watch Dog Reset for all calls above 50ms. Every wait call usually only need one instruction (2 Byte). Wait calls with interim value such as 8ms need two calls (5ms and 3ms or two times a 4ms call). I don't know any implementation, which is more economical if you use many wait calls in your program. The calls uses no registers, only the Stack Pointers for the return addresses in the RAM (at most 28 Byte stack space in current release) is used. The total list of functions is:

`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()` and `wait5s()`;

That are 36 functions with only 37 instructions inclusive Watch Dog Reset! There is really no way to shorten this library. Last not least matches the wait calls the exactly delay time, if the lowest wait call does. Only the wait calls above 50ms are one cycle per 100ms to long because of the additionally integrated watch dog reset.

Additionally the often used calling sequence `"wait5ms(); ReadADC...();"` is replaced by the call `"W5msReadADC(...);"`. The same is done for the sequence `"wait20ms(); ReadADC(...);"` which is replaced by one `"W20msReadADC(...);"` call. The function `ReadADC` is additionally written in assembly language, so that this add-on could be implemented very effective. The functional identical C-version of the `ReadADC` function is also available as source.

Chapter 8

To Do List and new ideas

1. Add more and better documentation.
2. By my tests I have noticed that the measured voltages of the internal band gap reference is lower than the data sheets let me expect. The reason is unknown. VCC?, ADC-error?
3. Check if transistor tester could get better interpolated ADC values if additional noise is added to the signal or to the ADC reference (see ATMEL document AVR121: Enhancing ADC resolution by oversampling). If all items are identical, there can't be any enhancement of resolution by oversampling. Can enough noise be generated with the ATmega counter? How additional noise affects the upper and lower limit values? Of course this method can not eliminate all of the ADC errors.
4. This method can be tested by building a ramp input signal and monitoring this signal. The ramp signal can be build by slowly charging a big capacitor with the 470k resistor. The growing of the voltage can then be monitored with the LCD display in a special part of self test.
5. The ADC operates with a clock frequency of 125kHz. The specification allows up to 200kHz with full accuracy. But the 200kHz clock is impossible to set with prescaler, if CPU-Clock is 1MHz or 8MHz. How much accuracy is lost, if the ADC Clock is set to 250kHz? Measurements could be done in nearly half the time, if 250kHz operation is tolerable.
6. Think about how we can get the real internal resistance of port B output (resistor switching port) instead of assuming, that ports are equal.
7. Can discharging of capacitors be made more quickly, if the minus pin is additionally raised with the 680 Ω resistor to VCC (+)?
8. Who is using the serial port? I did not test this function and even I don't know how.
9. Can inductance be tested?
10. How measurement results changes by variation of the supply voltage between 4,5V and 5V?
11. Check if the tester can use floating-point representation of values. The risk of overflow is lower. There is no need to use multiplication and division together to build a multiplication with a non integer factor. But I don't know how much flash memory must be spend for the library.
12. Write User's guide for configuring the tester with the Makefile options and description of the build chain.

13. If the holding current of a thyristor can not be reached with the 680Ω resistor, is it harmless to switch the cathode directly to GND and the anode directly to VCC for a very short time? The current could reach more than 100mA. Will the port be damaged? What is with the power supply (voltage regulator)?
14. Check the Port afterwards with self test function!
15. Can voltage regulators be checked? (Input, Output, GND)
16. Can optoelectronic couplers be checked?
17. Is the ESR measurement of electrolytical capacitors possible.
18. Warning message, if the found reference voltage is not plausible in relation to ATmega model and VCC.
19. Can we connect precision voltage reference to PC4 to calibrate VCC and internal reference?
20. If a battery cell is connected to the tester, the tester tries to discharge, but fails without message, better is it to recognise the part as cell (with voltage?).
21. What is about a second generation tester with a bigger ATmega which includes differential ADC-port, more flash memory . ? There is no ATxmega which have supply voltage of 5V, only the ATmega line is possible.
22. Idea for a New Projekt: USB version without LCD-Display, Power from USB, Communication to PC over a USB-Serial bridge.

Bibliography

- [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 documentation of the Transistortester*, Online Article, 2009-2011
- [11] <http://www.mikrocontroller.net/articles/AVRDUDE> *Online documentation of avrdude programmer interface*, Online Article, 2004-2011
- [12] <http://www.mikrocontroller.net/topic/131804> *Thread from Markus*, Forum thread, 2009
- [13] http://www.mikrocontroller.net/articles/AVR_Transistortester *Short description of new features of the TransistorTester von Karl-Heinz K.*, Online Article, 2012
- [14] <http://www.mikrocontroller.net/topic/248078> *Thread from Karl-Heinz*, Thread and new software versions, 2012