TransistorTester with AVR microcontroller and a little more
Version 0.97k

Karl-Heinz Kübbeler
kh_kuebbeler@web.de

July 20, 2012

Contents

Preface

Basically Motive

Every hobbyist knows the following problem: You disassemle 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 7. 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.

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,

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 a 100nF 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 $27k\Omega$ resistor is added to the PD7 (pin 12) 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 3. 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.

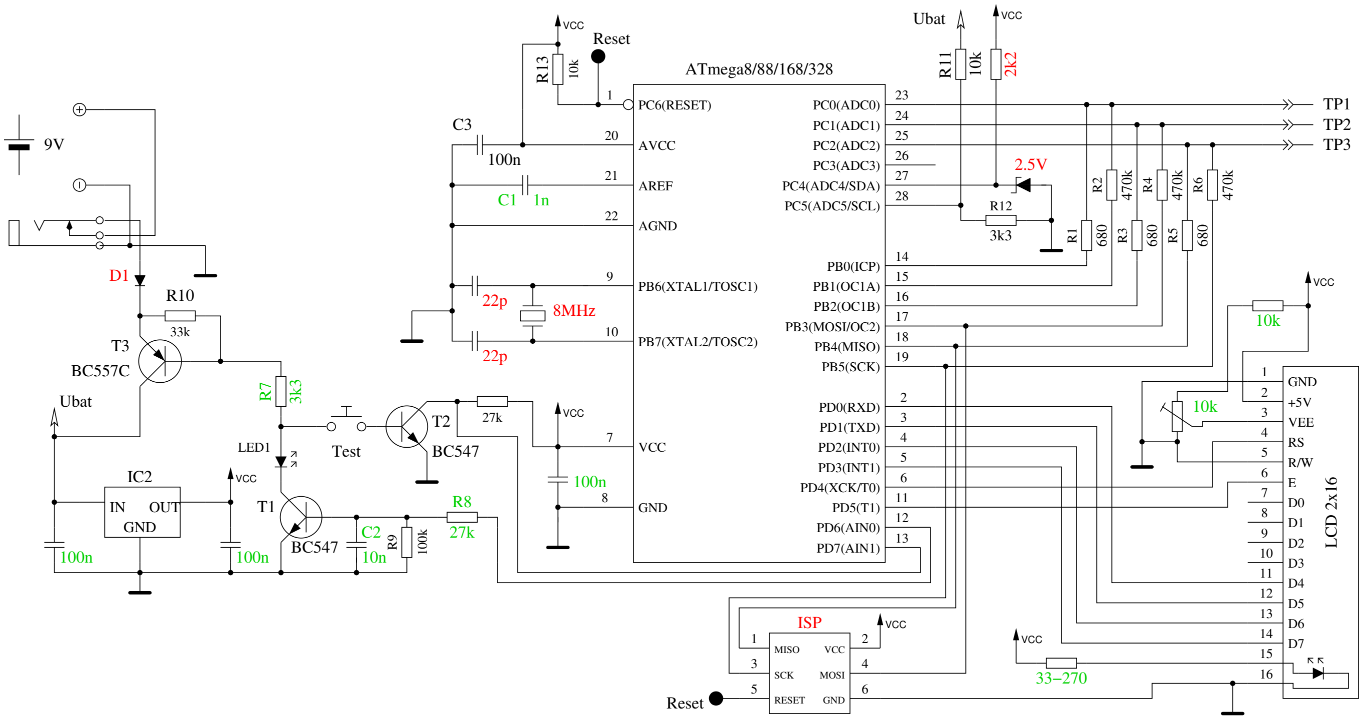


Figure 2.1: New circuit of TransistorTester

2.2 Hints for building the TransistorTester

Every LCD-display with a 2 x 16 character display and a HD44780 compatible controller can be used with the TransistorTester. You should respect the current needed for illumination, some LCD need lower current than others. I had tried OLED type displays, but this type cause interference with measurements of the ATmega and are **not** recommended. The resistors R1 to R6 are critical for measurements and this 680Ω and $470k\Omega$ resistors should be measurement type resistors (tolerance of 0.1%) to get the full accuracy. You should use a precision socket for the ATmega microcontroller to enable the replacement of the microcontroller. The microcontroller ATmega8, ATmega88, ATmega168 and ATmega328 can be used. Recommended is a ATmega168 or ATmega328, if you wish to use all features. Anyway you should assemble all parts to printed board without the microcontroller. A up-to-date low voltage drop regulator like MCP1702-5002 is recommended as IC2, because it need only $2\mu A$ of standby current and can still deliver 5V, if your input voltage is only 5.4V. But this part is not pin compatible to well known 78L05 with TO92 body! After checking, that all needed parts are at the correct place, you should first connect the battery or power supply to the printed board without LCD-display and microcontroller. You should check the voltage at the power pins of the microcontroller and LCD-display terminal during the Test key is pressed. The voltage should disappear, if you release the Test key. If the voltage had correct polarity and value, you should disconnect the power and assemble the microcontroller with correct alignment. Be careful and make shure, that all pins of the microcontroller are in the socket holes. Now you can also connect the LCD. Check if power pins of the LCD has the right connection to GND and VCC of your board. If you are shure that everything is all right, reconnect the power. If you have already programmed the ATmega, you can press the Test button. By pressing the Test key, the background light of the LCD should switch on. If you release the Test button, the LED should illuminate weak. Notice, that the software for the microcontroller must be compiled for the correct processor type. A program for the ATmega8 does not run on a ATmega168!

2.3 Programming of the microcontroller

I release the software for the microcontroller with source code. The developement is done with Linux operationg 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 3. 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 and 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 all parameter data 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 4.4. 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

Configuring the TransistorTester

The complete software for the TransistorTester is available in source code. The compilation of modules is controlled with a Makefile. The developement 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.

The following options in the Makefile are avaiable 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 avaiable
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 avaiable 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

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 in the WITH_AUTO_REF mode.
example: CFLAGS += -DREF_C_KORR=14

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 3.1a and 3.1b 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.
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. Possible values for 5V Operation are 44, 22 and 11. The 44 gives best accuracy, but the longest measurement time. One ADC measurement with 44 values takes about 4.7ms.
example: CFLAGS += -DANZ_MESS=44

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 time-out. 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). The maximum value is 255 (CFLAGS += -DPOWER_OFF=255).

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

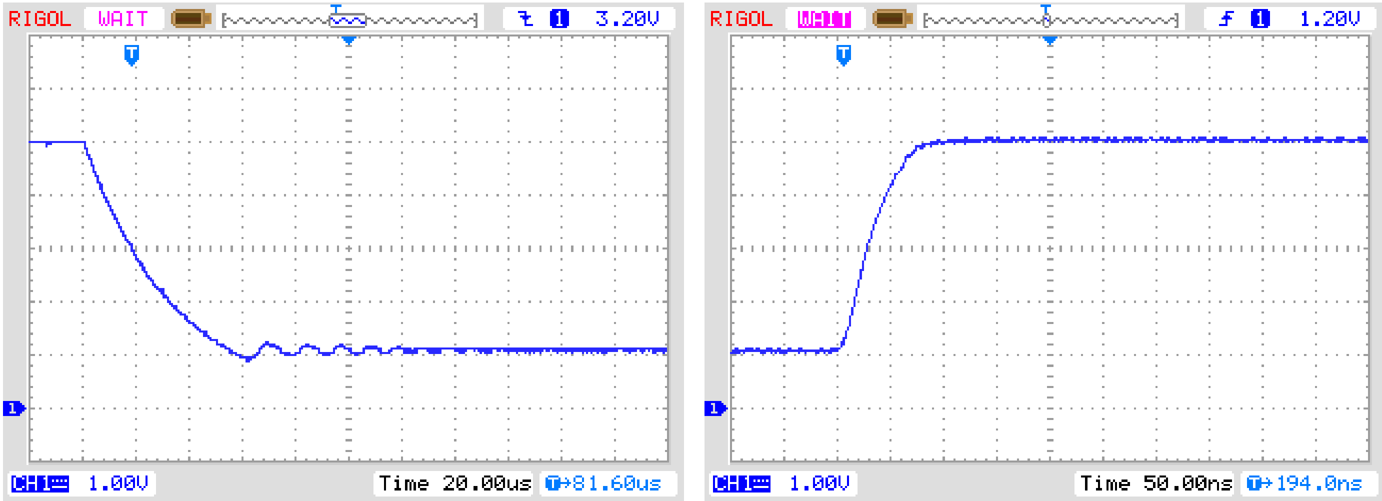
BAT_OUT enables Battery Voltage Output on LCD (if BAT_CHECK is selected). If your 9V supply has a diode installed, use the BAT_OUT=600 form to specify the threshold voltage (mV) of your diode to adjust the output value. Also the voltage loss of transistor T3 can be respected with this option. threshold level does not affect the voltage checking levels (BAT_POOR).
examples: CFLAGS += -DBAT_OUT=300 or 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

UF_OUT_MV set the output format of the flux voltage to mV units. Otherwise the flux voltage is shown in Volt units with two digit decimal place. The output format with Volt units save one digit at the LCD, the mV output format needs less program memory. The 16 digits of the LCD are sometimes insufficient for mV output together with capacity value, therefore you should prefere the Volt output.
exmple: CFLAGS += -DUF_OUT_MV

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 3.1: 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 4

Description of the measurement procedures

The simplified schematic of a Input/Output-Port Pin of the ATmega is shown in figure 4.1. 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 (PIN) can operate regardless to the state of the switch DD. The PORT switch usually defined the output level, but also switched 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 prefere 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.

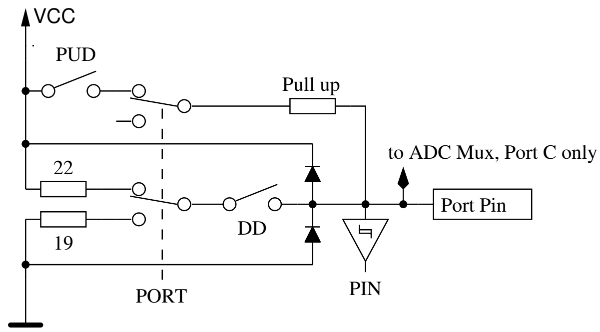


Figure 4.1: simplified diagram of each ATmega port pin

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 4.2.

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 4.1 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.

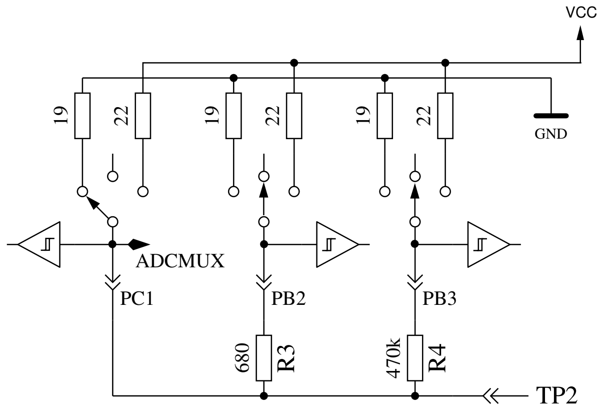


Figure 4.2: simplified circuit of each measurement terminal probe TP

	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 4.1: all combinations of measurement

4.1 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 4.1.1. If current was detected, the next test is described in the diode section 4.1.3.

4.1.1 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 4.3. 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 $470k\Omega$ resistor at the base. In this case the current amplification factor is build as $hFE = \frac{UE \cdot 470000}{UB \cdot (680+22)}$. Because the current amplification factor of Darlington Transistors can be very high, the factor is limitted to 65535 ($2^{16} - 1$).

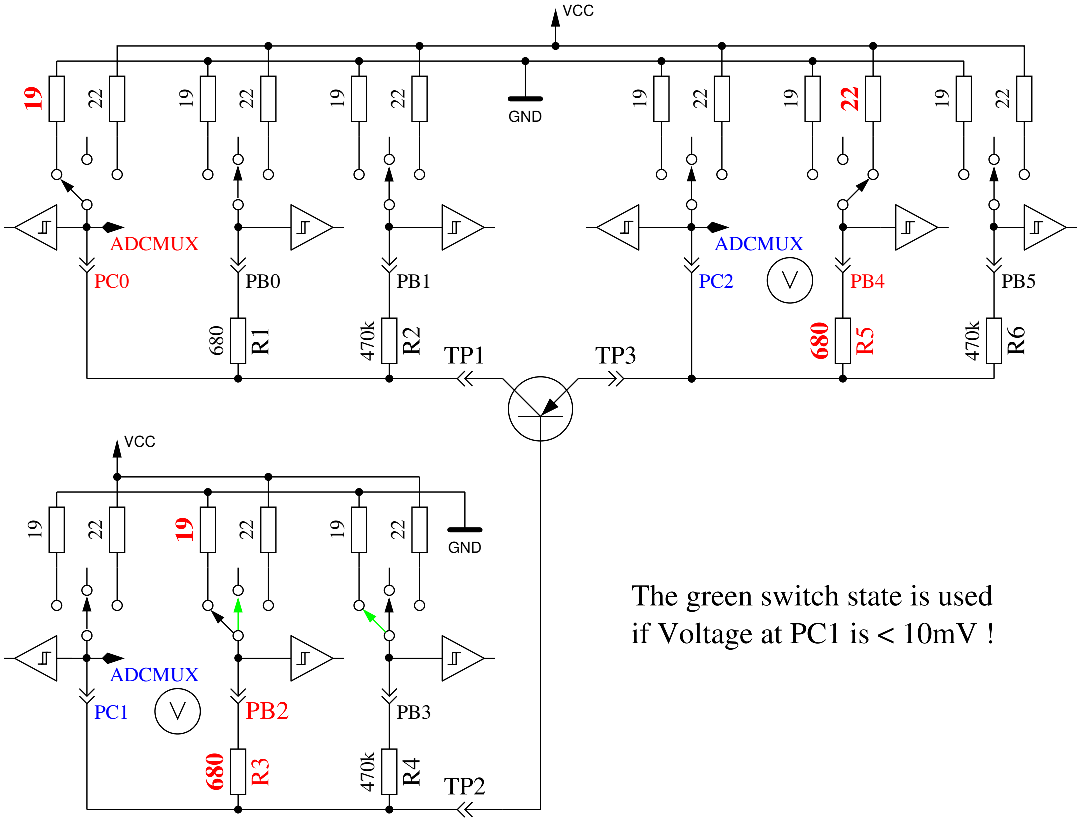


Figure 4.3: hFE measurement of PNP transistor with common collector circuit

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 4.4. 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 470$k\Omega$ 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 470$k\Omega$ resistor up and down, waiting for a digital input signal change of the Drain side and then read the voltage of the gate pin.

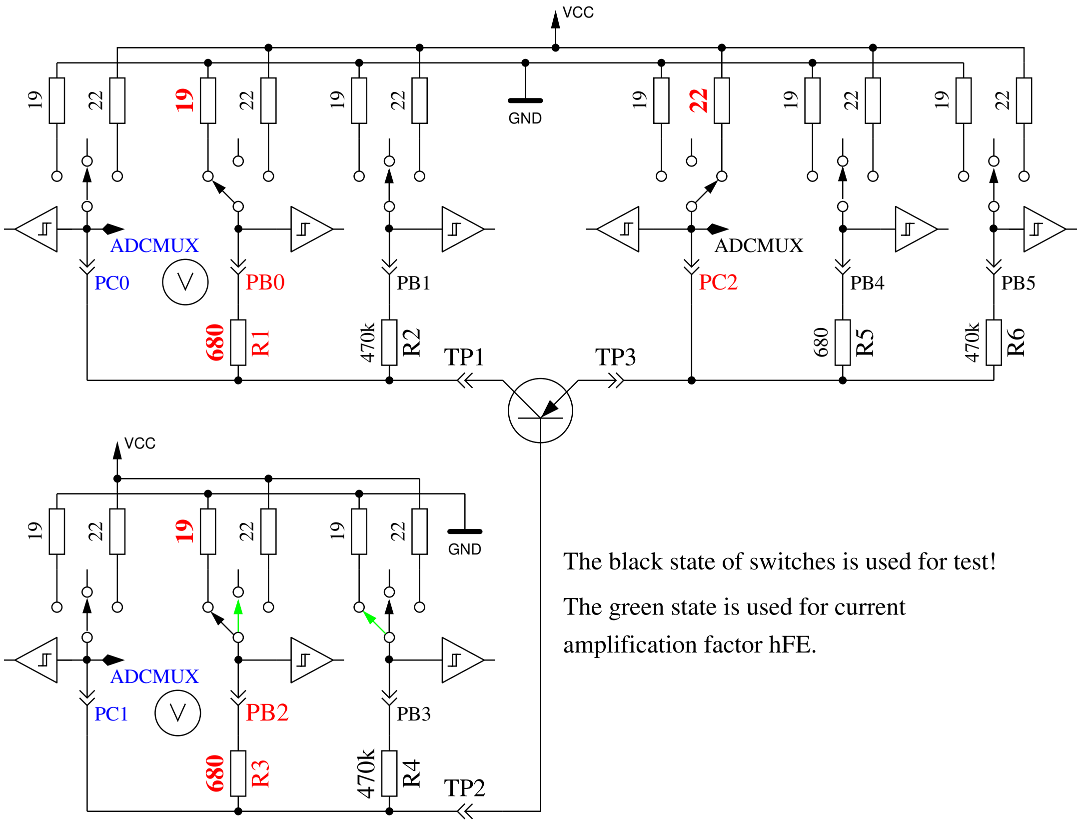


Figure 4.4: test and hFE measurement of PNP transistor with common emitter circuit

4.1.2 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 $470k\Omega$ resistor is taken instead. The amplification factor is limited to 65535 (16Bit). Measurement then continues with the common emitter circuit as shown in figure 4.5.

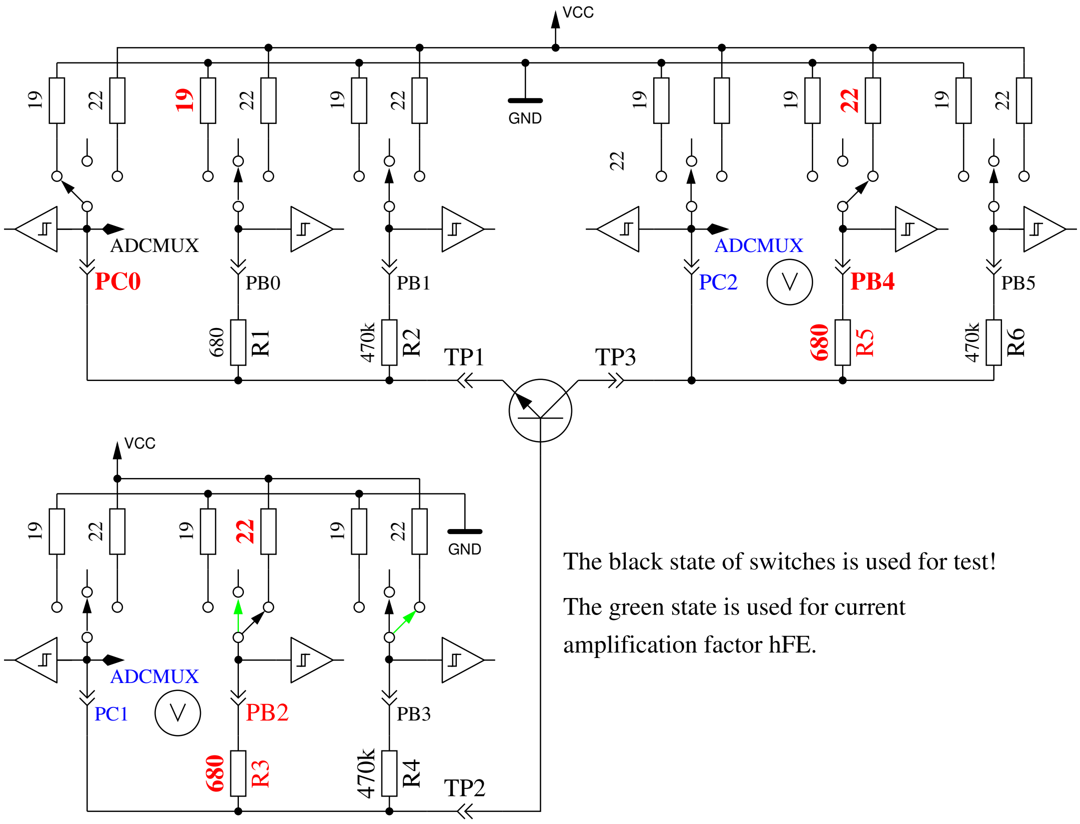


Figure 4.5: test and hFE measurement of NPN transistor with common emitter circuit

If the voltage of collector sinks below 1.6V, when the 680Ω base resistor is connected to VCC, ist must be a NPN, N-Channel MOSFET or Thyristor/Triac. With two simple tests a Thyristor or Triac can be identified. If the gate pin resistor is connected for 10ms to GND and than made currentless, the current at the anode should stay. If then the anode resistor is short connected to GND and reconnected to VCC, the Thyristor should not trigger again (no current). Please keep in mind, that only low power Thyristors can be tested, because the holding current of the tester can reach only 6mA. If both tests attest a Thyristor, further tests with reverse polarity are done to exclude or confirm a Triac.

If neither Thyristor nor Triac could be confirmed, it can be a NPN or N-Channel E-MOSFET. The Base voltage of a NPN Transistor will be near the Emitter voltage, so this type can be identified definitely. The current amplification factor in the common emitter circuit is build by $hFE = \frac{(VCC-UC)\cdot 470000}{(VCC-UB)\cdot(680+22)}$. If the voltage of the Base or better Gate shows, that there is no or little current, part will be a N-Channel E-MOS (Enhancement MOSFET). In this case the threshold voltage is measured by switching the Gate slowly with the $470k\Omega$ resistor to VCC and GND, waiting for a digital input signal change of the Drain side and then read the voltage of the Gate pin. This measurement is done eleven times with ADC results accumulated as shown in Figure 4.6. The result is multiplied by four and divided by 9 to get the voltage in mV resolution.

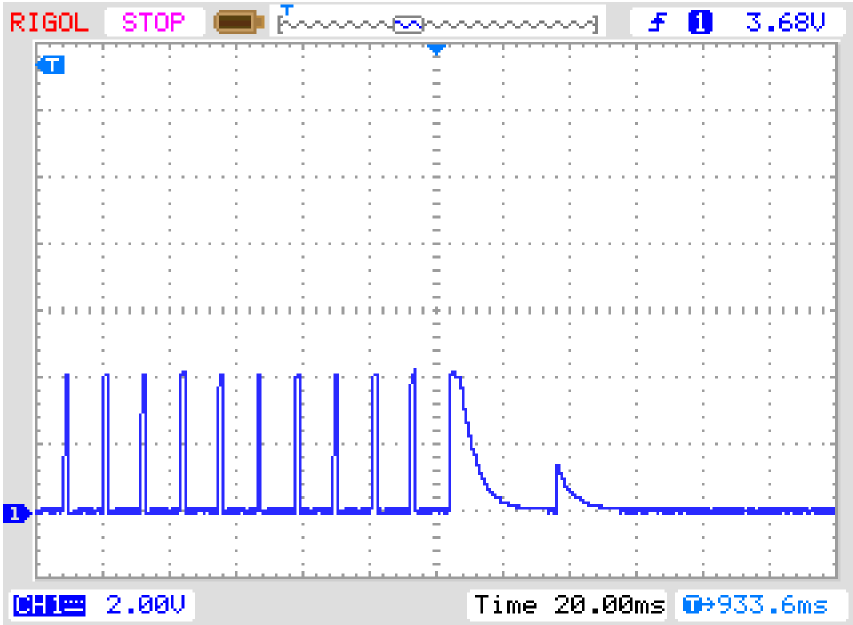


Figure 4.6: measuring of threshold voltage of N-Channel-MOSFET

4.1.3 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 470$k\Omega$ resistor and eight times the flux voltage with the 470$k\Omega$ must be greater than the flux voltage with the 680Ω resistor. I hope, that this behavior is always a diode.

4.1.4 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, 721mV, 0pF	Diode, 729mV, 0pF	Diode, 725mV, 0pF
1N4150	Diode, 678mV, 0pF	Diode, 681mV, 0pF	Diode, 682mV, 0pF
BA157	Diode,623mV, 17pF	Diode, 631mV, 16pF	Diode, 620mV, 15pF
BY398	Diode, 541mV, 0pF	Diode, 553mV, 0pF	Diode, 542mV, 0pF
1N4007	Diode, 654mV, 13pF	Diode, 665mV, 9pF	Diode, 658mV, 11pF
LED green	Diode, 1954mV, 6pF	Diode, 1970mV, 6pF	Diode, 1951mV, 4pF
ZPD2,7	2xDi, 729mV, 2659mV	2xDi, 738mV, 2674mV	2xDi, 730mV, 2656mV
BU508A B+E	Diode, 613mV, 5201pF	Diode, 621mV, 5285pF	Diode, 611mV, 5344pF
BU508A B+C	Diode, 595mV, 261pF	Diode, 597mV, 267pF	Diode, 591mV, 272pF

Table 4.2: 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=9, 613mV	NPN, B=9, 621mV	NPN, B=9, 615mV
2N3055	NPN, B=21, 617mV	NPN, B=21, 626mV	NPN, B=21, 625mV
BC546B	NPN, B=381, 780mV	NPN, B=376, 777mV	NPN, B=387, 771mV
BC556B	PNP, B=266, 790mV	PNP, B=429, 787mV	PNP, B=266, 790mV
BC639	NPN, B=180, 722mV	NPN, B=180, 733mV	NPN, B=188, 724mV
BC640	PNP, B=185, 716mV	PNP, B=227, 725mV	PNP, B=187, 719mV
AC128 (Ge.)	PNP, B=68, 270mV	PNP, B=64, 269mV	PNP, B=66, 271mV
BC517	NPN, B=26996, 1419mV	NPN, B=28220, 1413mV	NPN, B=28250, 1404mV
BC516	PNP, B=65535, 1430mV	PNP, B=65535, 1420mV	PNP, B=65535, 1417mV
BRY55/200	Thyristor	Thyristor	Thyristor

Table 4.3: 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, 2616mV 66pF	N-E-MOS,D, 2562mV 67pF	N-E-MOS,D, 2564mV 68pF
J310	N-JFET	N-JFET	N-JFET
IRFU120N	N-E-MOS,D, 4151mV 922pF	N-E-MOS,D, 4156mV 894pF	N-E-MOS,D, 4153mV 933pF
IRFU9024	P-E-MOS,D, 3525mV 960pF	P-E-MOS,D, 3525mV 926pF	P-E-MOS,D, 3534mV 965pF
ZVP2106A	P-E-MOS,D, 3217mV 115pF	P-E-MOS,D, 3220mV 114pF	P-E-MOS,D, 3217mV 113pF
ZVNL120A	N-E-MOS,D, 1560mV 140pF	N-E-MOS,D, 1535mV 138pF	N-E-MOS,D, 1535mV 138pF

Table 4.4: measurement results of MOS transistor testing

4.2 Resistor Measurement

Each resistor is measured with four different types of measurement in one current direction. The same resistor ist 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.

4.2.1 Resistor Measurement with 680 Ohm Resistors

The measurement of a unknown resistor Rx is done in two ways with the build in precision 680Ω resistors. The diagram of this measurements for test pin 1 (TP1) and test pin 3 (TP3) are simplified shown in figure 4.7 and figure 4.8 as a example of the six choises of probe combinations.

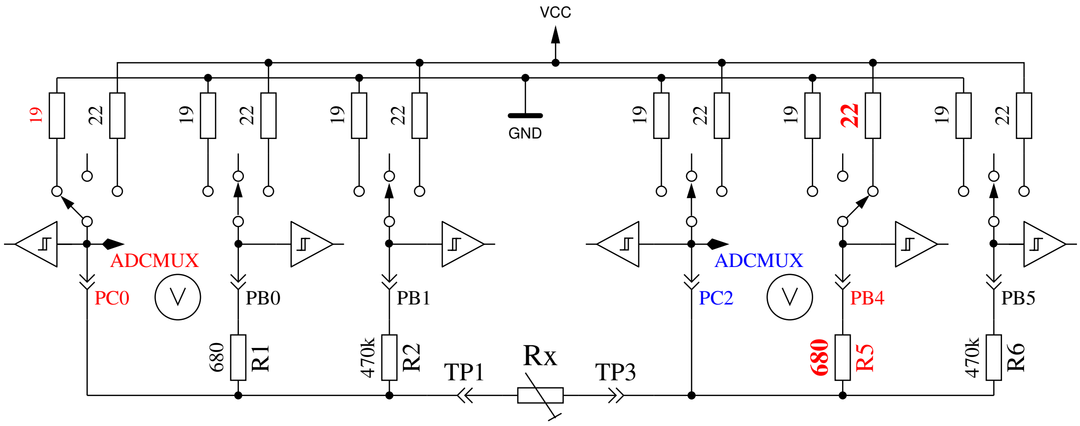


Figure 4.7: Measurement type 1 with 680Ω

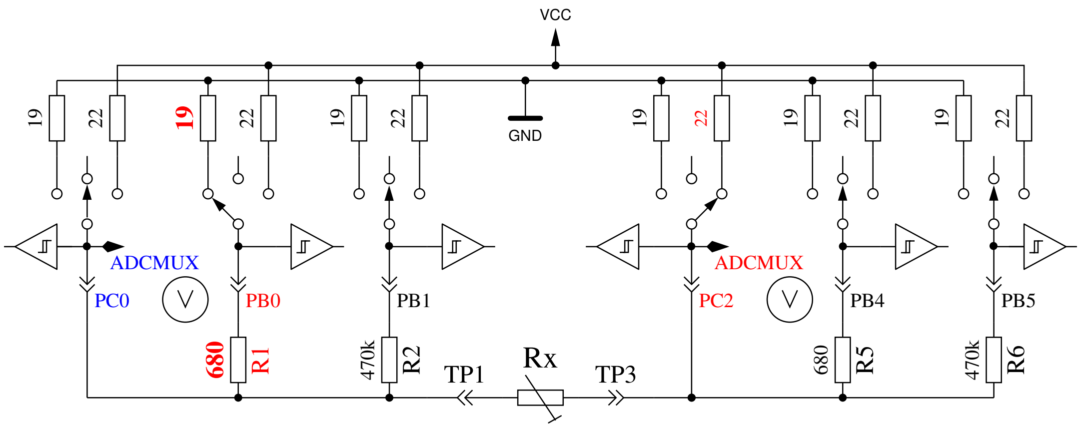


Figure 4.8: Measurement type 2 with 680Ω

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 Rx is allways 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 4.9, where resistor values are shown in logarithmic scale.

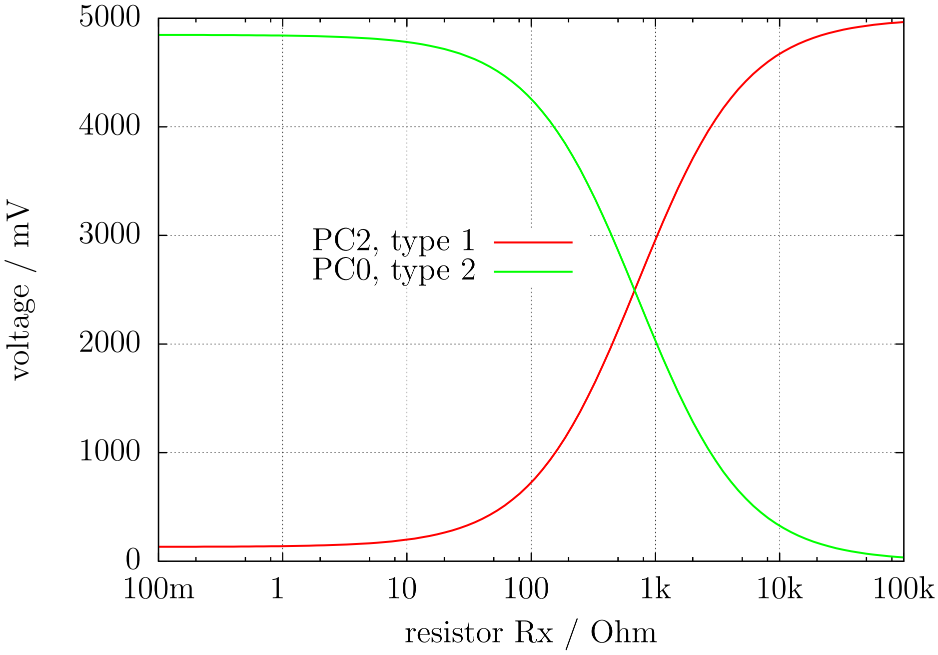
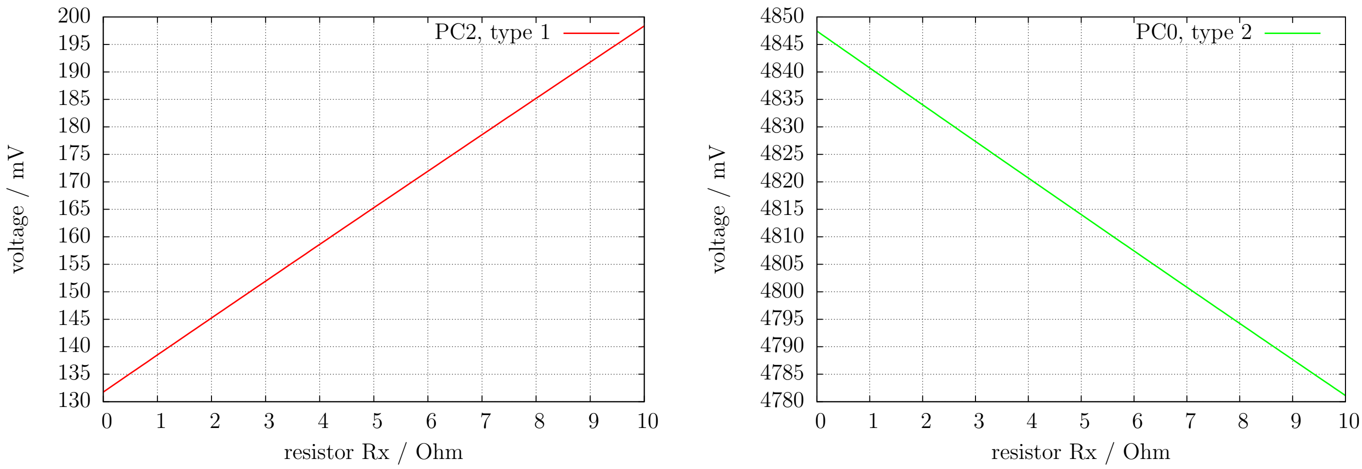


Figure 4.9: Voltages of type 1 and type 2 measurements with 680Ω

The graph of measurement type 1 is shown in figure 4.10a 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 4.10b. 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 $20k\Omega$ (Voltage will be below 169mV). For higher resistor values the measurements with the $470k\Omega$ 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 4.10: Cut-out of theoretical Voltage from 0Ω to 10Ω

4.2.2 Resistor Measurement with 470 kOhm resistors

The next figures 4.11 and 4.12 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 Rx.

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 4.13 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 700Ω will be added. This offset can be adjusted with the RH_OFFSET define in the config.h file.

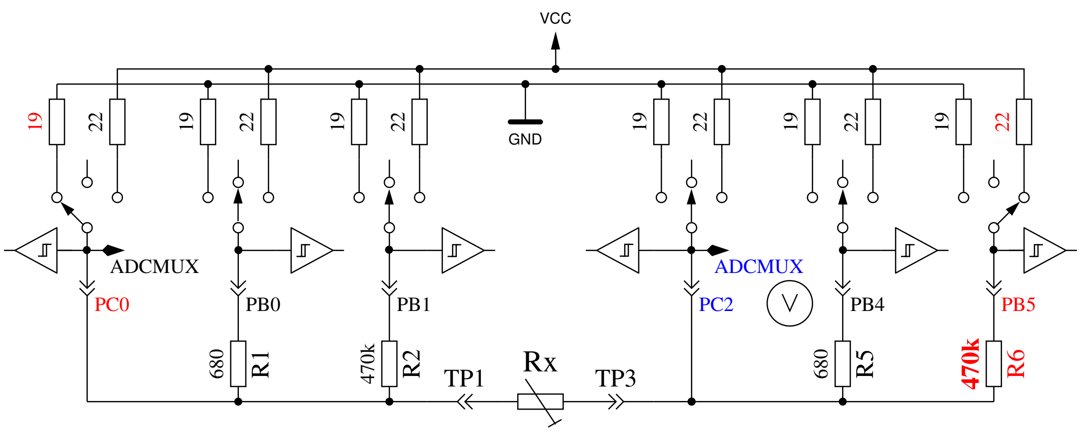


Figure 4.11: Measurement type 3 with $470k\Omega$

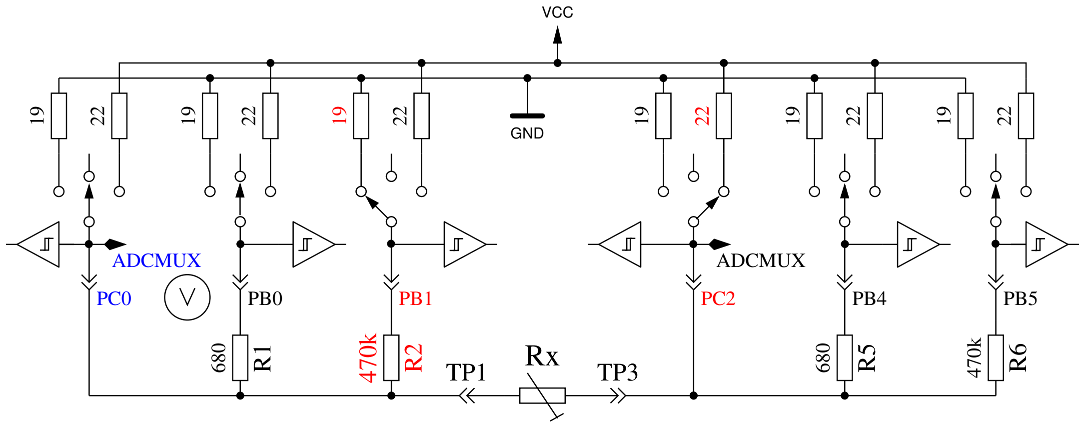


Figure 4.12: Measurement type 4 with $470k\Omega$

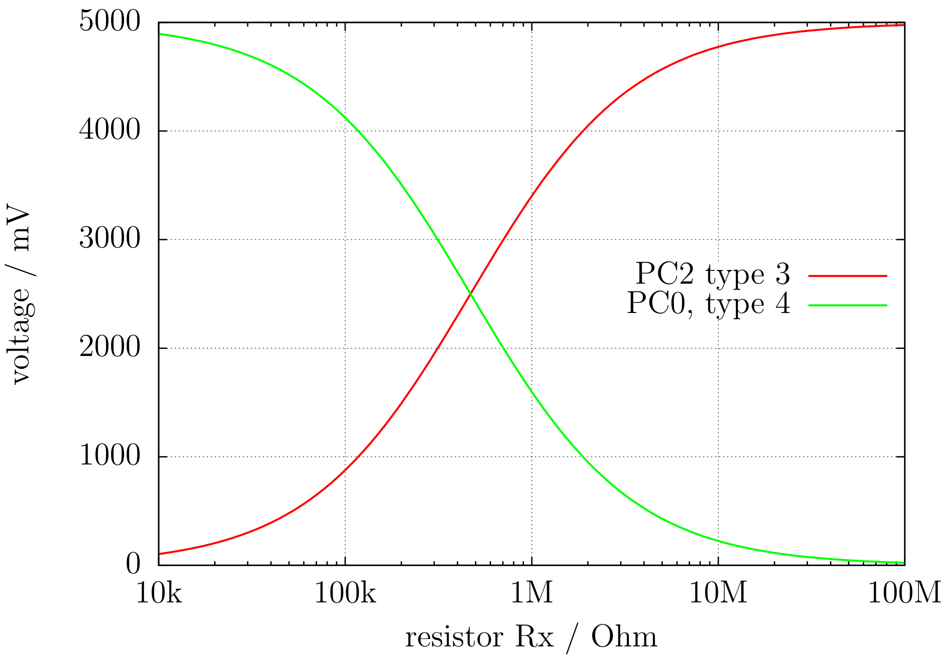


Figure 4.13: Voltages of type 3 and type 4 measurements with $470k\Omega$

4.2.3 Results of the resistor measurements

Figure 4.14 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 4.15 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 $20M\Omega$ 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%!

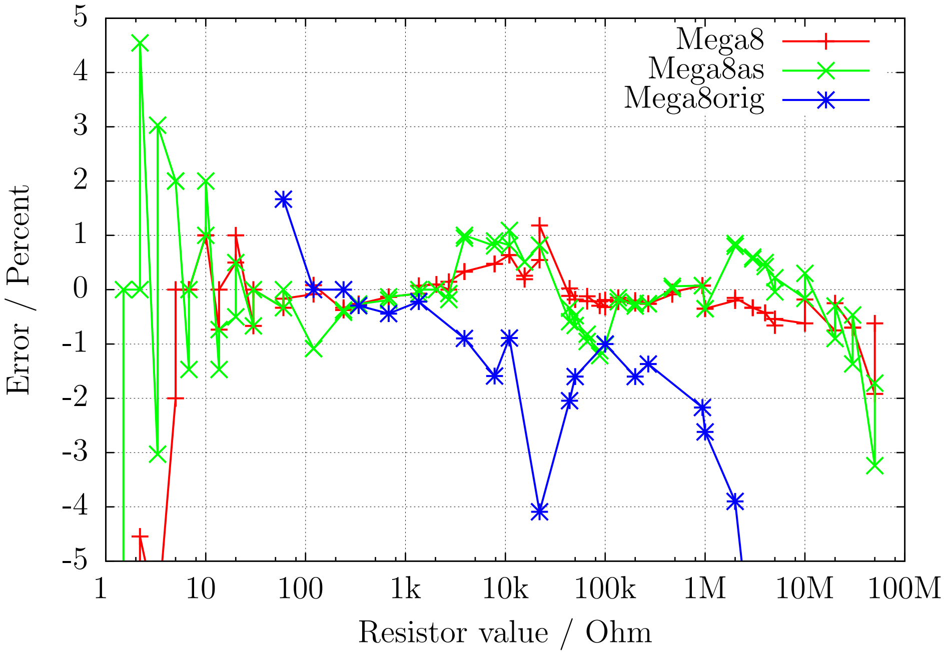


Figure 4.14: Relative error in % for resistor measurements with ATmega8

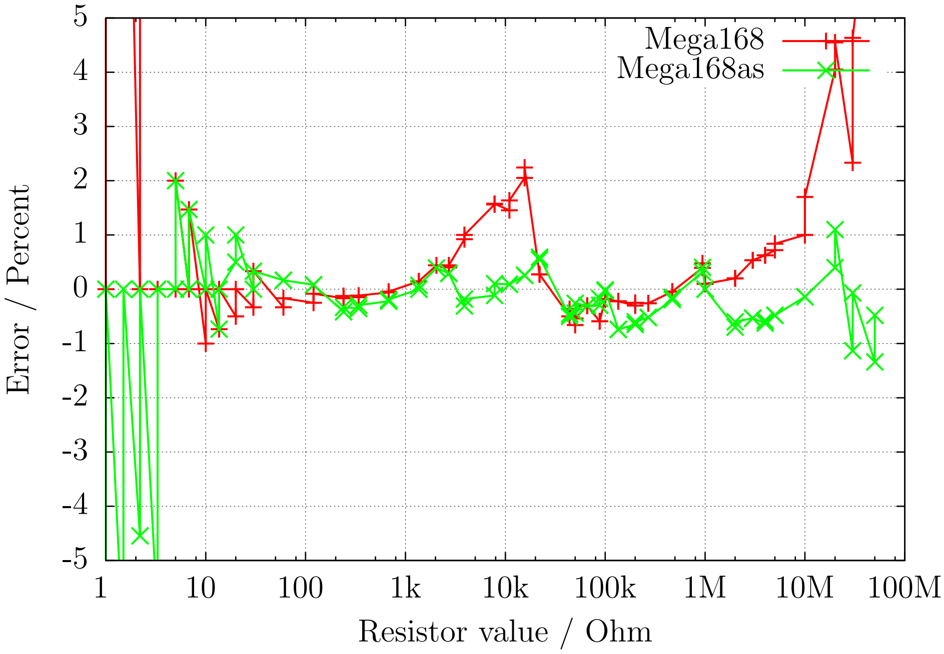


Figure 4.15: Relative error in % for resistor measurements with ATmega168

4.3 Measurement of Capacitors

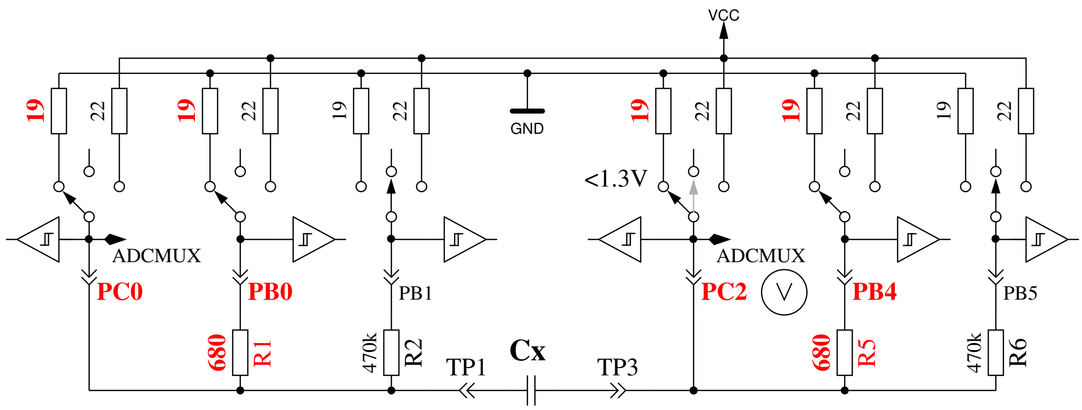
The measurement of capacitor values are done as separate task by measurement of load time after all other measurements. The original software of Markus F. did this with a program loop, which reads the corresponding digital input pin until a switch occured and count the loop cycles. This has the handicap, 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.

4.3.1 Discharging of Capacitors

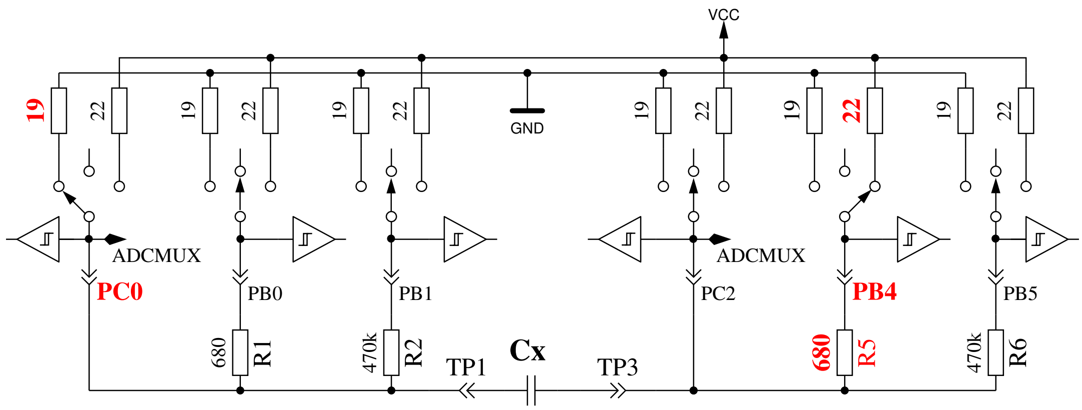
You should always discharge the capacitor before connecting it to the tester. The tester additionally discharge the capacitor before any measurement. If the voltage is below 1300mV, the capacitor is shortened by the output pins of the connected ADC port (Port C). I believe that this is legal because 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. Off course damage can occur if you do not discharge a (high voltage) capacitor before connecting it to your tester.

4.3.2 Measurement of big Capacitors

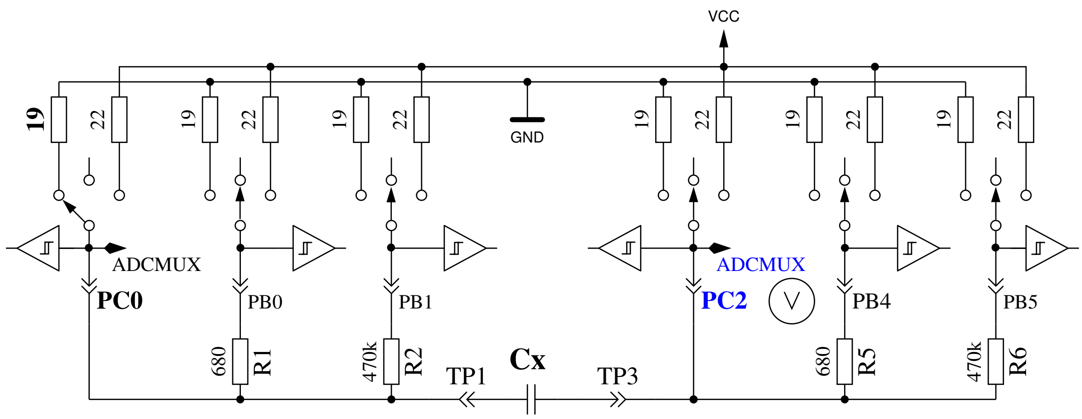
One side of the capacitor is connected to GND. The other side of the capacitor is connected with the 680Ω resistor to VCC for a period of 10ms. Afterwards this probe pin is switched to Input (High Impedance). After this 10 ms current pulse the voltage of the capacitor is measured without any current. 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 4.16 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.



Quick Discharge of capacitor



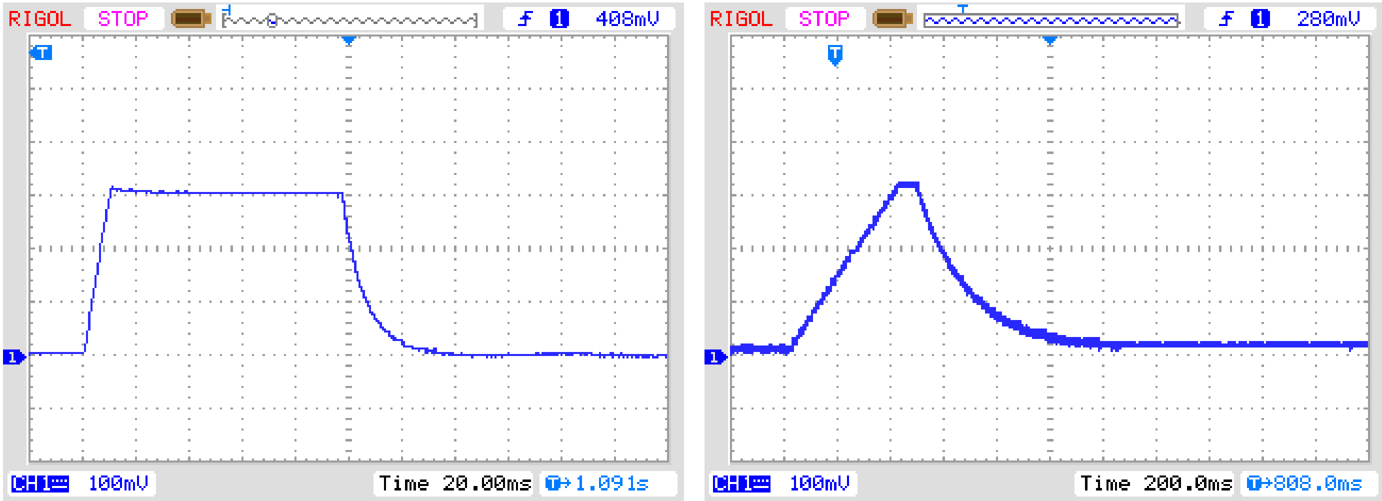
10ms Charge Phase of capacitor



Voltage Measurement Phase of capacitor

Figure 4.16: 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 4.17a 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 4.17b 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 4.18



(a) $229\mu F$ Capacitor (b) $5mF$ Capacitor

Figure 4.17: Charge and discharge of big Capacitors for measuring

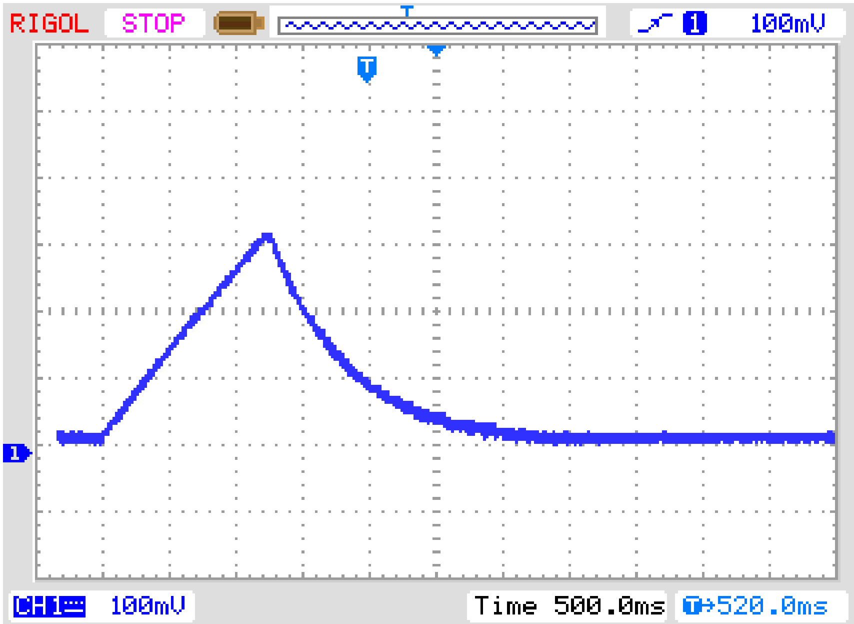


Figure 4.18: Charge and discharge of a $15mF$ Capacitor for measuring

4.3.3 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 build 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 4.19 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.

The input capture counter and the overflow counter built together the total time, from which I subtract a experimental find out constant to eliminate the measurement offset. I don't know, if this constant must be adapted to other printed circuit boards. The actual software can use a table with the theoretical dependency of the load time in respect to the comparator voltage. The table is spaced in 50mV steps and will be interpolated according to the actual reference voltage. This table will only be acticated with the Makefile option WITH_AUTO_REF. I noticed that the reference voltage is permanently somewhat to low, so that you can choose an offset with the Makefile option REF_KORR. The measured reference voltage will then be corrected (added) by your value (mV units). If option WITH_AUTO_REF is not used, the reference voltages of ATmega8, ATmega88, ATmega168 and ATmega328 are applied as noted in the data sheets [2] [3]. A sample measurement of this type is shown in figure 4.20. The measurement time is above 2.6s because the 470$k\Omega$ is used for charging, but discharge is in this case much faster than charging.

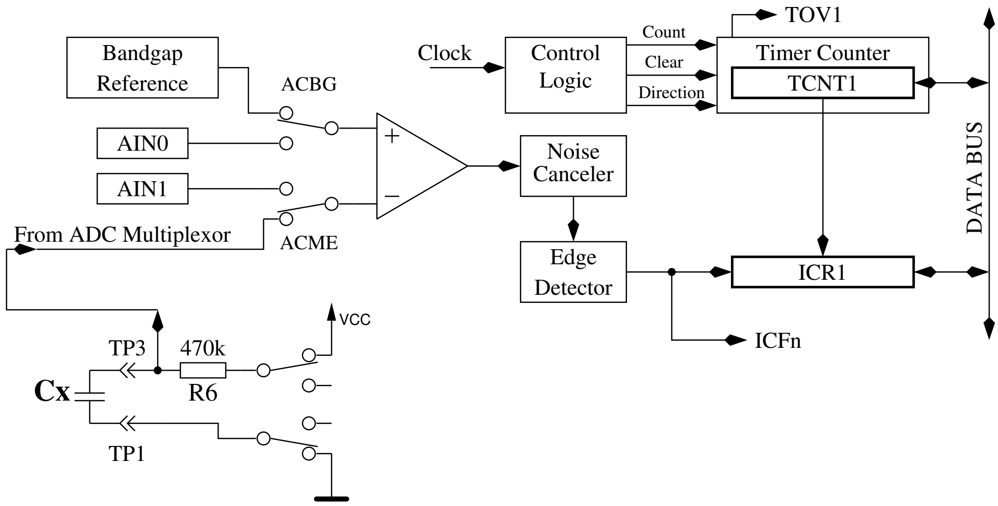


Figure 4.19: measurement little capacity values with comparator

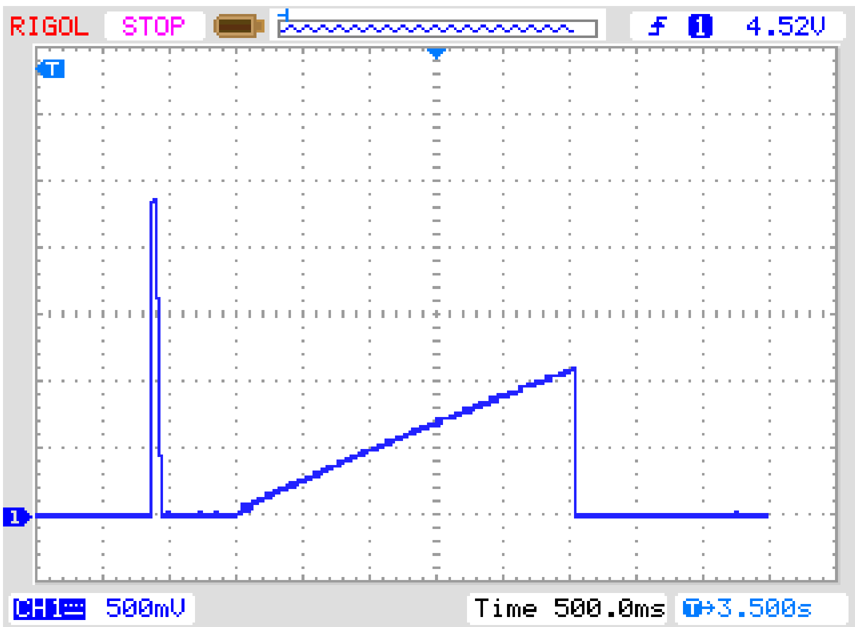


Figure 4.20: Charge and discharge of a 22μF Capacitor for measuring

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 4.3.2 avoids this conceptual bug.

4.3.4 Results of Capacitor measurement

The results of my measurements is shown if figure 4.21 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 ATmega168 is shown in figure 4.22. The reference for the error computing is the measurement of a PeakTech 3315 multimeter, not the printed value of the parts. In future I hope to take the measurements of a PeakTech 2170 RCL-meter as more exactly base values. There is a trend in relative errors with the big capacitor measurement, which I can't explain. I hope, that I will find the reason for this.

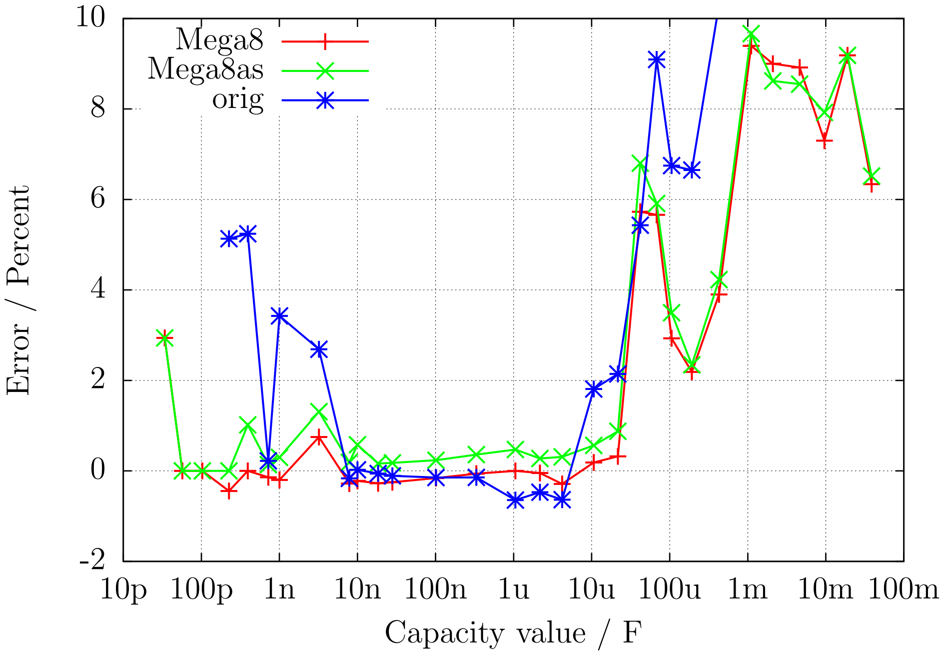


Figure 4.21: Error in % for capacitor measurements with ATmega8

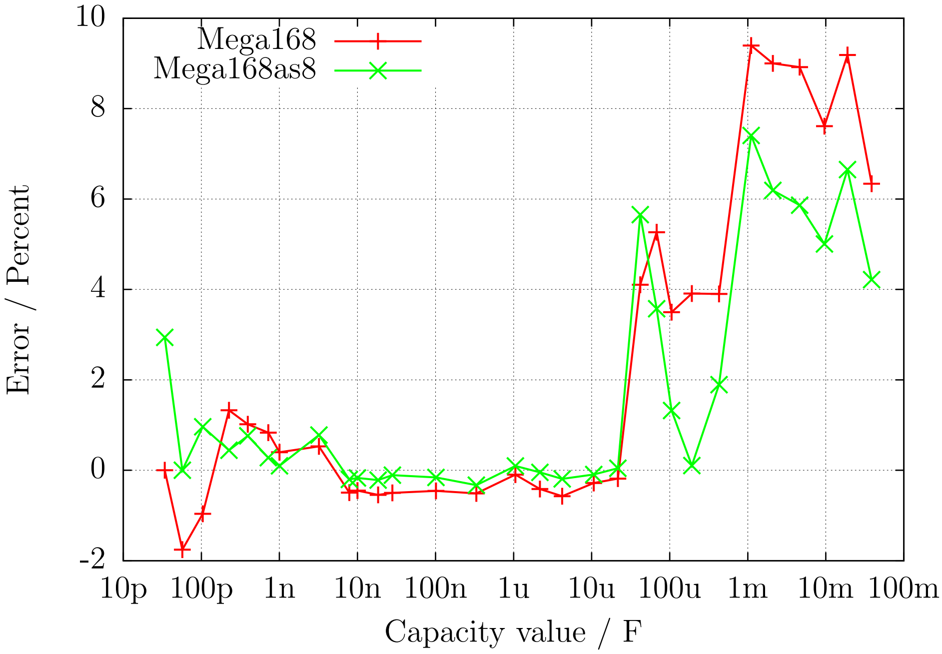


Figure 4.22: Error in % for capacitor measurements with ATmega168

The differences of measurements of three different ATmega168 processors and one ATmega168A processor are shown in figure 4.23 . 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".

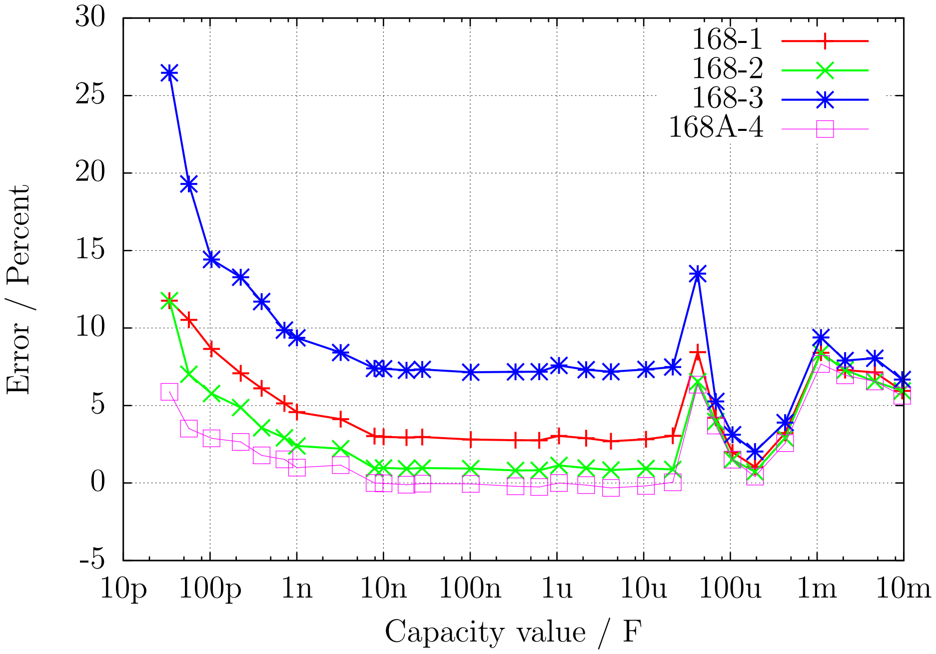


Figure 4.23: Error of capacitor measurements of three ATmega168 processors

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 0.1% . 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 measurement.

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 4.24 shows the error of the capacity measurement when only the zero value is respected (168-A), with correction factor for little capacitors REF_C_KORR=66 as well as the correction factor for big capacitors C_H_KORR=50 (168-B), plus additional as gradient 168-C with a model of the slew rate dependency of little capacitor measurements (COMP_SLEW1=4000 und COMP_SLEW2=220). The component with the slew 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.

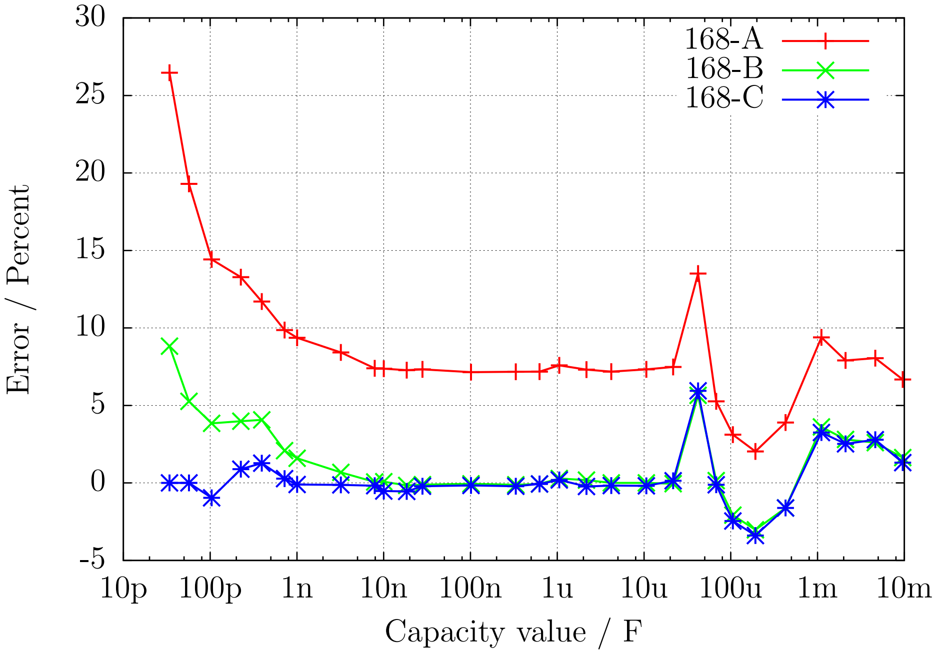


Figure 4.24: Improvement of the capacitor measurement of one ATmega168

4.4 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.

In every step only measurement results are displayed, 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 $470k\Omega$ 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. 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. 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. Please remember, that the resolution of the ADC is about 4.88mV! The measurement situation is also shown in figure 4.25. All these combinations with respect to the internal resistance of the pins should result to: $\frac{5001\cdot(19+680)}{(19+680+680+22)} = 2493$.

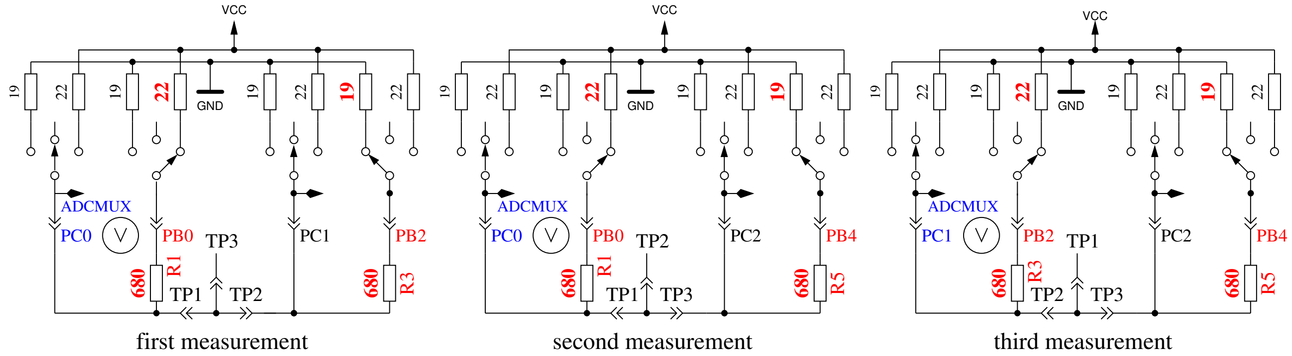


Figure 4.25: Comparison of 680Ω resistors

3. **Comparing of the** $470k\Omega$ **resistors.** Now the display shows in row 1 “+RH- 12 13 23”. The same procedure as done in step 2 is repeated with the $470k\Omega$ resistors (symbols RH). Result should be nearly $\frac{5001\cdot(19+470000]}{(19+470000+470000+22)} = 2500$ for all combinations.

4. In this step nothing is measured, but the **order is displayed isolate Probe**, which means that it is time to separate the probes (release from wire).

5. **This step tests the capability of GND (-) connected** $470k\Omega$ **resistors (H) to pull the test pins to GND.** Row 1 shows the text “RH-”. Row 2 should display zero for all three pins.

6. **This step tests the capability of VCC (+) connected** $470k\Omega$ **resistors (H) to pull the test pins to VCC (+).** Row 1 shows the text “RH+”. The best value for this three measurements is 5001. Great differences from the best value for test 5 and 6 are errors such as isolation problem, flux material or damaged port.

7. **Measuring of internal resistance of pin output switched to the GND signal.** The text in the 1st LCD row is “Ri_Lo = (mV)”. In the second row of the LCD three voltages were displayed. The internal resistance of the port C outputs switched to GND (-) are measured with the current of to VCC (+) switched 680Ω resistors, see Figure 4.26. Only the three pins of the ADC port are measured, the resistor port B (PB0,PB2 and PB4) can not be measured without hardware modification. Is is assumed that the port resistance of the different ports are nearly identical. To get the resistor values, you must divide the displayed mV values by about 7 (see test 8).

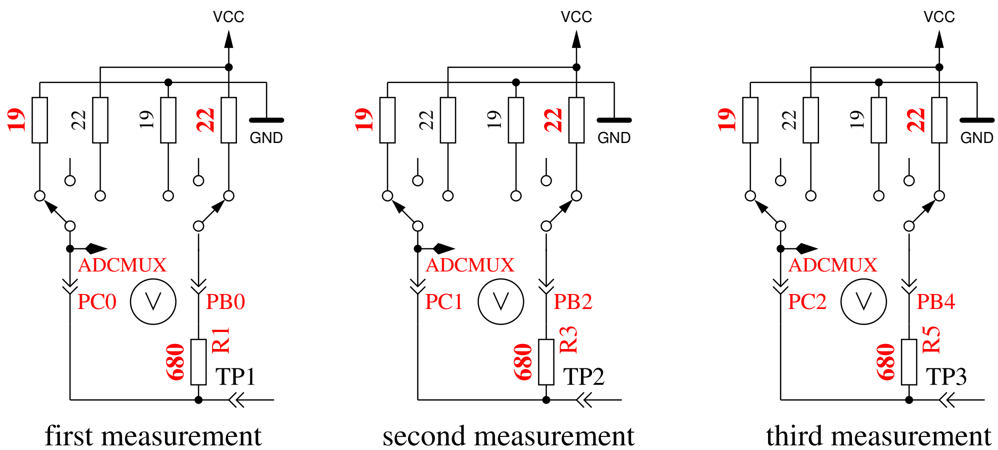


Figure 4.26: Measurement of internal resistance of Port C switched to GND

8. **Measuring of internal resistance of port outputs switched to the VCC (+)signal.** The needed current is generated with to GND connected 680Ω resistors . The text in the 1st LCD row is "Ri_Hi= (mV)". In the second row of the LCD three voltages are displayed (in difference to VCC). It are the same measurements as those in test 7 to the other side as you can see in Figure 4.27. With the following steps you can get the resistance: To get the current build: $(5001 - (resultoftest7) - (resultoftest8))/680$. Then you can get both resistor values by dividing the voltage (result of test 7 or 8) by this current.

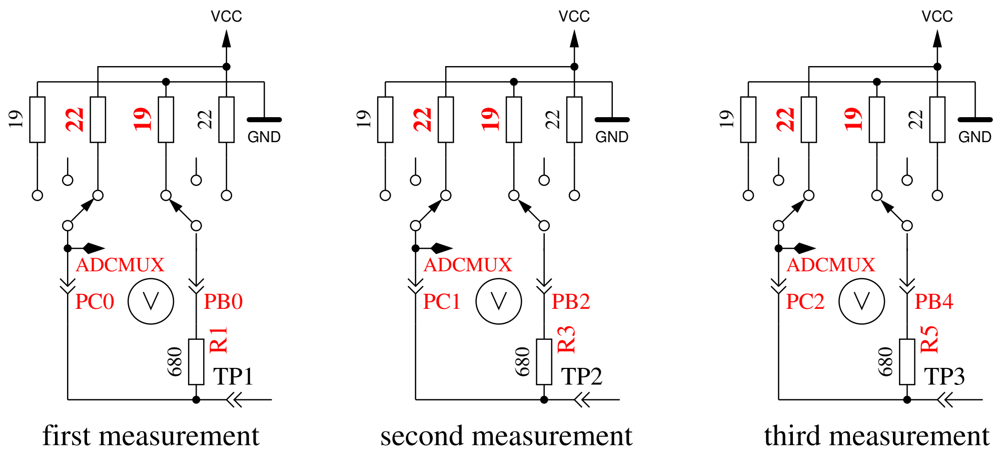


Figure 4.27: Measurement of internal resistance of Port C switched to VCC

9. 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 test is repeated 8 times with 5 seconds period 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 inexactly. A exactly clock frequency and wait time are important for measurement of capacity values.

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. Then the program continues with the normal measurement task.

Test No.	1. Result	2. Result	3. Result
Test 1 Reference	band gap Ref 1237 RHfakt 753	 RLfakt. 4887	
Test 2 comparison 680Ω best value: 2493	RL1+ RL2- 2488	RL1+ RL3- 2488	RL2+ RL3- 2484
Test 3 comparison 470$k\Omega$ best value: 2500	RH1+ RH2- 2493	RH1+ RH3- 2493	RH2+ RH3- 2493
test 4	isolate probe		
test 5 470$k\Omega$ isolation best value: 0	RH1- 0	RH2- 0	RH3- 0
Test 6 470$k\Omega$ isolation best value: 5001	RH1+ 4995	RH2+ 4995	RH3+ 4995
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

Table 4.5: Selftest Mega8 @ 8MHz, Signature 1E9307, WITH_AUTOREF

Test No.	1. Result	2. Result	3. Result
Test 1 Reference	band gap Ref 1090 RHfakt 865	 RLfakt. 5649	
Test 2 comparison 680Ω best value: 2493	RL1+ RL2- 2493	RL1+ RL3- 2493	RL2+ RL3- 2493
Test 3 comparison 470$k\Omega$ best value: 2500	RH1+ RH2- 2497	RH1+ RH3- 2498	RH2+ RH3- 2498
test 4	isolate probe		
test 5 470$k\Omega$ isolation best value: 0	RH1- 0	RH2- 0	RH3- 0
Test 6 470$k\Omega$ isolation best value: 5001	RH1+ 4998	RH2+ 4998	RH3+ 4998
Test 7 Pin resistance Low	TP1- RL1+ 131	TP2- RL2+ 132	TP3- RL3+ 132
Test 8 Pin resistance High	TP1+ RL1- 156	TP2+ RL2- 156	TP3+ RL3- 156

Table 4.6: Selftest Mega168 @ 8MHz, Signature 1E9406, WITH_AUTOREF

Chapter 5

Known errors and unsolved problems

Software Version 0.97k

1. The measurement results of little capacity values vary with the Pin combinations. Combination 1:2 values are about 3pF less than the values of the other pin combinations (1:3 and 2:3). This effect is equal on any tested AVR processor.
2. Germanium Diodes (AC128) are not detected in all cases.
3. Does program work correctly without the automatic shut off?
4. The shown Pin numbers of resistor measurement are sometimes identical and stay in this error state until power is switched off (Version 0.95k).
5. 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 normaly 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.
6. 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!

Chapter 6

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 adresses 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 avaiable as source.

Chapter 7

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. The difference of the ReadADC function alternatives $\frac{44}{9}, \frac{22 \cdot 2}{9} or \frac{11 \cdot 4}{9}$ can be monitored too.

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 680Ω 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 Manipolation 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/topic/248078` *Thread from Karl-Heinz,* Thread and new software versions, 2012