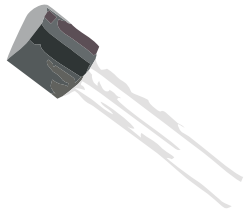
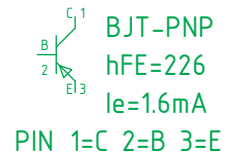
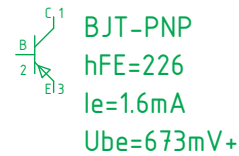


Transistor Tester with AVR microcontroller.

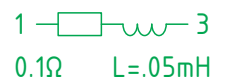
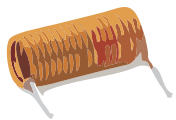
A device for determining and measuring
electronic components
and a little more ...



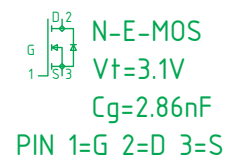
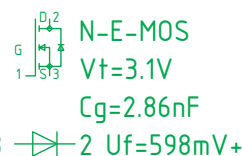
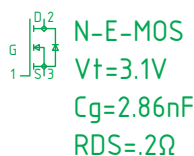
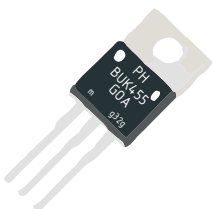
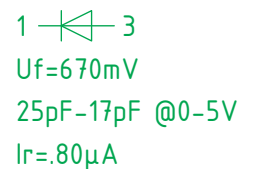
Version 1.13k



Karl-Heinz Kübbeler
kh_kuebbeler@web.de



September 23, 2022



13.05.2019/MOR

Contents

1	Features	5
2	Hardware	9
2.1	Circuit of the TransistorTester	9
2.2	Extensions for the TransistorTester	11
2.2.1	Protection of the ATmega inputs	11
2.2.2	Measurement of zener voltage above 4 Volt	12
2.2.3	Frequency generator	12
2.2.4	Frequency measurement	13
2.2.5	Using of a rotary pulse encoder	13
2.2.6	Connection of a graphical display	15
2.2.7	Connection of a graphical color display	20
2.3	Hints for building the TransistorTester	21
2.4	Changeover for tester versions designed by Markus F.	22
2.5	Extended circuit with ATmega644 or ATmega1284	23
2.6	Buildup of a tester with ATmega1280 or Arduino Mega	25
2.7	Chinese clones with text display	27
2.8	Chinese clones with graphical display	28
2.9	Chinese kits with graphic displays	32
2.10	And another clone from Hiland M644	34
3	Instructions for use	36
3.1	The measurement operation	36
3.2	Optional menu functions for the ATmega328	37
3.3	Selftest and Calibration	40
3.4	special using hints	41
3.5	Compoments with problems	41
3.6	Measurement of PNP and NPN transistors	42
3.7	Measurement of JFET and D-MOS transistors	43
3.8	Measurement of E-MOS transistors and IGBTs	43
3.9	Measurement of capacitors	43
3.10	Measurement of coils	44
4	Programming of the TransistorTester	45
4.1	Configuring the TransistorTester	45
4.2	Programming of the microcontroller	56
4.2.1	Using of Linux	56
4.2.2	Building the Software with Windows	64
4.2.3	Using the WinAVR package with Windows	64
4.3	Troubleshooting	66

5	Description of the measurement procedures	68
5.1	Measurement of Semiconductors	70
5.1.1	Measurement of PNP Transistor or P-Channel-MOSFET	71
5.1.2	Measurement of NPN Transistor or N-Channel-MOSFET	72
5.1.3	Simplified flowchart of the transistors tests	75
5.1.4	Measurement of Diodes	77
5.1.5	Results of different measurements	78
5.2	Resistor Measurement	82
5.2.1	Resistor Measurement with 680 Ohm Resistors	82
5.2.2	Resistor Measurement with 470 kOhm resistors	84
5.2.3	Results of the resistor measurements	85
5.3	Measurement of Capacitors	89
5.3.1	Discharging of Capacitors	89
5.3.2	Measurement of big Capacitors	89
5.3.3	Measurement of small Capacitors	92
5.3.4	Measurement of very small capacity values with the sampling technique	94
5.3.5	Measurement of the Equivalent Series Resistance ESR	95
5.3.6	Measurement of the Equivalent Series Resistance ESR, first way	98
5.3.7	Measurement of the Equivalent Series Resistance ESR, second way	101
5.3.8	Voltage loss after a load pulse, Vloss	107
5.3.9	Separate capacity and ESR measurement	108
5.3.10	Results of Capacitor measurement	109
5.3.11	Automatic calibration of the capacitor measurement	113
5.4	Measurement of inductance	117
5.4.1	Results of the inductance measurements	118
5.4.2	Measurement of small inductances with the sampling method	118
5.5	Selftest Function	122
5.5.1	Some Results of the Selftest Function	126
5.6	Measurement of frequency	130
5.6.1	Calibration of frequency with a GPS- or GLONASS-receiver	131
5.6.2	Calibration of the crystal frequency with a clock modul	132
6	Signal generation	134
6.1	Frequency Generation	134
6.2	Puls width generation	134
7	Known errors and unsolved problems	135
8	Special Software Parts	137
9	To Do List and new ideas	138

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. [2] 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 [7]. 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 in chapter 9. By the way, now I can program the EEprom of the ATmega with Linux operating system without errors.






At this place I would like to thank the originator and software author Markus Frejek, who has enabled the continuation with his initial work. In addition I would like to say thanks to the authors of numerous input to the discussion forum, which have assist me, to find new tasks, weak points and errors. Next I would like to thank Markus Reschke, who give me the permission, to publish his cheerful software versions at the SVN server. Furthermore some ideas and software part of Markus R. was integrated in my own software version, again thank you very much. Also Wolfgang SCH. has done a great job to support a graphical display with ST7565 controller. Many thanks to him for integrating his patch for 1.10k software version to the actual developer version. I have to thank also Asco B., who has developed a new printed board, to enable the reproductions for other hobbyists. Another thank I would like to send to Dirk W. , who has handled the omnibus order for this printed board. Never I had time enough to handle these things concurrently with my software development, at no time the state of further development of software would have the same level. Thanks for the many suggestions to improve the tester to the members of the local chapter of the

“Deutscher Amateur Radio Club (DARC)” in Lennestadt. Also I would like to say thanks for the integration of the sampling method to the radio amateur Pieter-Tjerk (PA3FWM). With this method the measurement of little capacity and inductance values could be improved noticeable. Last but not least I would say thanks to Nick L. from Ukraine, who has supported my with his prototype boards, has suggested some hardware add ons and also has organized the russian translation for this description.

Chapter 1

Features

1. Operates with ATmega8, ATmega168 or ATmega328 microcontrollers. Additionally ATmega644, ATmega1284, ATmega1280 or ATmega2560 microcontrollers can be used.
2. Displaying the results to a 2x16 or a 4x16 character LCD-Display. If the processor have at least 32K flash memory, also a graphical display with 128x64 pixel and a ST7565, ST7920, NT7108, KS0108 or a SSD1306 controller can be used. In doing so a 4-wire SPI interface or a I²C bus must be used instead of the 4-bit parallel interface for the SSD1306 controller. Even color displays with ILI9163 or ST7735 controller are usable with the SPI interface. For the NT7108 or KS0108 controller you must use a serial - parallel interface with a 74HC(T)164 or a 74HC(T)595. Displays with PCF8812 or PCF8814 controller can be only used without the big icons for transistors, because the display size is insufficient (102x65 and 96x65).
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. With software version 1.05k the sleep modus of the Atmega is used to reduce current if no measurement is required and no rotary encoder is used (not with ATmega8).
6. Automatic detection of NPN and PNP bipolar transistors, N- and P-Channel MOSFETs, JFETs, diodes, double diodes, N- and P-IGBTs, Thyristors and Triacs. For Thyristors and Triacs the ignition current and holding current must be reached with the tester. For IGBTs the 5V signal should be enough for driving the gate voltage.
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, the Gate capacity value and the R_{DSon} at almost 5V gate voltage of MOSFETs.
12. Up to two Resistors are measured and shown with symbols  and values with up to four decimal digits in the right dimension. 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 up to 0.01Ω, values up to 50MΩ are detected.

14. One capacitor can be detected and measured. It is shown with symbol  and value with up to four decimal digits in the right dimension. The value can be from $25pF$ ($8MHz$ clock, $50pF$ @ $1MHz$ clock) to $100mF$. The resolution can be up to $1pF$ (@ $8MHz$ clock).
15. For capacitors with a capacity value above $20nF$ the Equivalent Serial Resistance (ESR) is measured with a resolution of 0.01Ω and is shown with two significant decimal digits. This feature is only available for ATmega with at least 16K flash memory.
16. For capacitors with a capacity value above $5000pF$ the voltage loss after a load pulse can be determined. The voltage loss give a hint for the quality factor of the capacitor.
17. Up to two diodes are shown with symbol  or symbol  in correct order. Additionally the flux voltages are shown.
18. LED is detected as diode, the flux voltage is much higher than normal. Two-in-one LEDs are also detected as two diodes.
19. Zener-Diodes can be detected, if reverse break down Voltage is below $4.5V$. These 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 $700mV$!
20. 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.
21. 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. If a ATmega with more than 8K flash memory is used, also the reverse current of the diode is measured with a resolution of $2nA$. The value is only output to the screen, if it is greater than zero.
22. Only one measurement is needed to find out the connections of a bridge rectifier.
23. 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. For processors with at least 32K flash memory the tester changeover to a special capacity meter function, if a capacitor with more than $25pF$ is once measured at TP1 and TP3. In this capacitor meter mode you can measure capacitors below $25pF$ at TP1 and TP3 directly.
24. For resistors below 2100Ω also the measurement of inductance will be done, if your ATmega has at least 16K flash memory. For that a inductor icon  is shown behind the resistor icon . The range will be from about $0.01mH$ to more than $20H$, but the accuracy is not good. The measurement result is only shown with a single component connected.
25. Testing time is about two seconds, only capacity or inductance measurement can cause longer period.
26. Software can be configured to enable series of measurements before power will be shut down.
27. Build in selftest function with optional $50Hz$ Frequency generator to check the accuracy of clock frequency and wait calls (only for processors with at least 32K flash memory).
28. Selectable facility to calibrate the internal port resistance of port output and the zero offset of capacity measurement with the selftest (only for processors with at least 16K flash memory). A external capacitor with a value between $100nF$ and $20\mu F$ connected to 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.

29. Display the Collector cutoff current I_{CE0} with currentless base ($1\mu A$ units) and Collector residual current I_{CES} with base hold to emitter level (only for processors with at least 16K flash). This values are only shown, if they are not zero (especially for Germanium transistors).
30. For a ATmega with at least 32K flash memory the tester changes the operation modus from universal tester to a resistor tester, if only one resistor is detected at the test pin 1 (TP1) and test pin 3 (TP3). If you have selected the additional measurement of inductance for the resistor meter function with the Makefile option RMETER_WITH_L, also the inductance of components is measured in this mode. This special operating mode is indicated with a [R] or [RL] at the right side of the first display line. In the same way the tester changes to a capacity meter function, if a capacitor is detected at TP1 and TP3. This operation mode is indicated with a [C] at the right side of the first display line. With this capacity meter function capacity values are shown down to $1pF$. Only for the automatic start you need a capacitor with more than $25pF$. Both special modes can be finished by a key press. The tester will then continue with the normal tester function.
31. For processors with at least 32K flash a dialog function can be selected, which enable additional functions. Of course you can return from dialog to the normal Transistor Tester function.
32. With dialog function you can use a frequency measurement at port PD4 of the ATmega. The resolution is $1Hz$ for input frequencies above $33kHz$. For lower frequencies the resolution can be up to $0.001mHz$ by measuring the mean period. You should read the subsection Frequency measurement 2.2.4 at page 13 for details of connection a frequency signal.
33. With the dialog function and without the activated serial output a external voltage of up to $50V$ can be measured with the 10:1 voltage divider at the PC3 port. If the PLCC-Version of the ATmega328 is used, one of the additional pins can be used for the voltage measurement together with the serial output. If the zener diode measurement extension (DC-DC converter) is assembled, the measurement of zener diodes is also possible with this function by pressing the key.
34. With the dialog function a frequency output can be selected at the TP2 pin (PB2 Port of the ATmega). Currently frequencies from $1Hz$ up to $2MHz$ can be selected.
35. With the dialog function a fixed frequency output with selectable pulse width can be activated at the TP2 pin (PB2 port of the ATmega). The width can be enhanced with 1% by a short key press or with 10% by a longer key press.
36. With the dialog function can be started a separated capacity measurement with ESR measurement. Capacities from about $2\mu F$ up to $50mF$ can most be measured in circuit, because only a little measurement voltage of about $300mV$ is used.
37. If your ATmega has at least 32K flash memory (Mega328), you can also use the sampling method of the ADC, which can measure capacitors with less than $100pF$ at a resolution of up to $0.01pF$. With the same method you can also measure coils with less than $2mH$ with significant better resolution of the inductance by building a resonant circuit with a parallel capacitor of known capacity value.

You should make sure, that all capacitors have no residual charge before starting any measurement, especially if you use the tester for measurement of capacitor in circuit.

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 $6mA$! Also a IGBT can only be detected correctly, if a voltage of $5V$

is sufficient to drive the gate. Notice that many features can only be used with microcontroller with enough program memory such as ATmega168. Only processors with at least 32k flash memory like ATmega328 or ATmega1284 can take all features.

Attention: Always be sure 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.

Extra caution is required if you try to test components mounted in a circuit. In either case the equipment should be disconnected from power source and you should be sure, that **no residual voltage** remains in the equipment.

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 [2]. 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 10 nF 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 voltage 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 speed 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 4.1 at page 45.

Some different versions of R11 / R12 resistor combinations circulates in the internet. I have adapted my software to the original of Markus F. [2] with $10k\Omega$ and $3.3k\Omega$. The voltage ration of the divider can be set with Makefile options.

The additional 2.5V precision voltage reference connected at pin PC4 (ADC4) can be used to check and calibrate the VCC voltage, but is not required. You can use a LM4040-AIZ2.5 (0.1%), a LT1004CZ-2.5 (0.8%) or a LM336-Z2.5 (0.8%) as voltage reference. If you don't install the precision voltage reference and you don't add the relay extension, you should install a pull up resistor R16 to PC4 with a higher resistance value ($47k\Omega$). This helps the software to detect the missing voltage reference. A optional ISP connector has been added to easier load new software versions to the tester.

Port	char. LCD =1	ST7565 LCD =1	ST7565 LCD STRIP_GRID	additional function
PD0	pushbutton			
PD1	LCD-D7	LCD-SI	LCD-A0 (RS)	rotary encoder 2
PD2	LCD-D6	LCD-SCLK	LCD-REST	
PD3	LCD-D5	LCD-A0 (RS)	LCD-SCLK	rotary encoder 1
PD4	LCD-D4	LCD-REST	LCD-SI	frequency counter
PD5	LCD-E	(LCD-CE)		
PD7	LCD-RS	pushbutton	pushbutton	

Table 2.2. port assignments with option STRIP_GRID_BOARD

2.2 Extensions for the TransistorTester

2.2.1 Protection of the ATmega inputs

For better protection of the ATmega inputs one of the additional circuits 2.2 can be integrated. The de-energized contacts of the relay protect the ATmega without power. The contacts will be opened by software only for measurement. Also with the additional diode protection the chance of the ATmega will be better to survive the connection of a capacitor with higher residual voltage. A complete protection is not possible. Therefore capacitors should always be discharged before measuring.

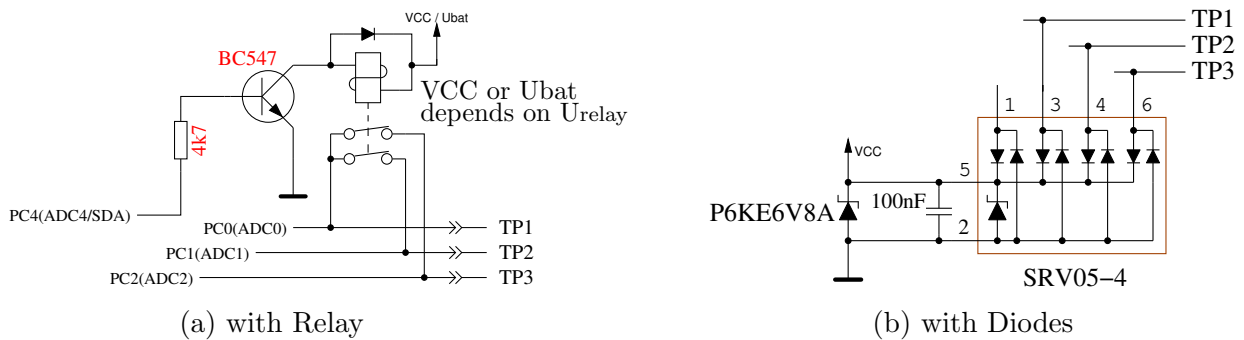


Figure 2.2. Additional protection of the ATmega inputs

You can get a better protection with a relay with 3 changeover contacts, which is shown in figure 2.3. The discharge current is limited with the resistors and the ATmega-inputs are disconnected in the protected mode. But you don't forget, that the tester remains unprotected during the measurement sequence.

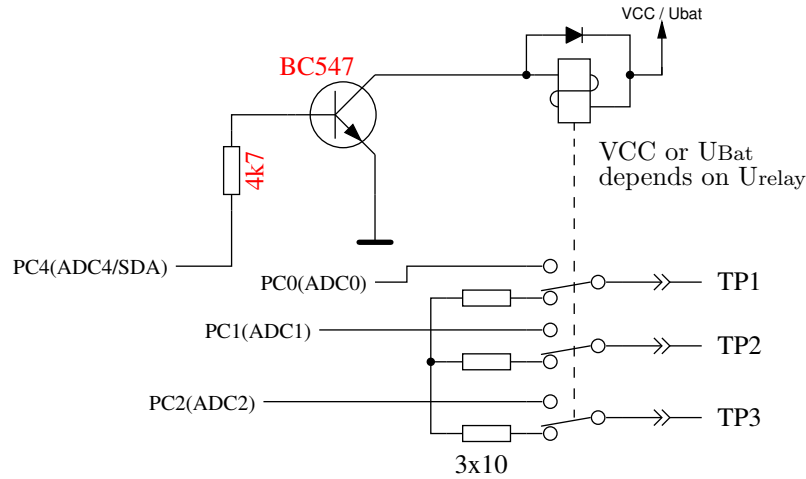


Figure 2.3. Improved protection with relay

2.2.2 Measurement of zener voltage above 4 Volt

If the serial output of text is not required, the Pin PC3 of the ATmega can be used as analog input for measuring a external voltage. The voltage can be up to 50V with the optional 10:1 resistor divider and can be used for measuring the breakdown voltage of a zener diode. A current limiting power supply with up to 50V can be switched on with low signal at PD7 pin of the ATmega to deliver current for testing the break down voltage of a zener diode. Figure 2.4 shows a suggestion for this expansion. The tester shows the external voltage as long as you hold the key pressed. About 40mA more battery current is used by this expansion during key pressing.

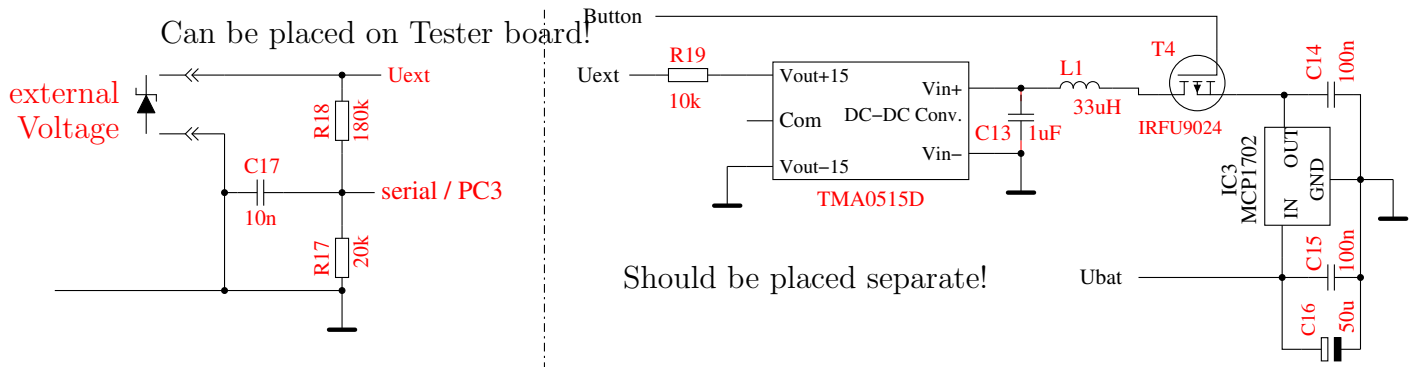


Figure 2.4. Expansion for measuring of break down voltage of Zener diodes

The 10:1 voltage divider can be used with the optional dialog part for the ATmega328 without the activated DC-DC converter for the zener diode measurement. Without the pressed key the voltage converter is not powered. For that the external voltage (for example battery voltage) can be measured at the zener diode port. You can only measure positiv DC voltages up to 50V. You have also to respect the correct polarity.

2.2.3 Frequency generator

With the dialog part of the ATmega you can also select a frequency generator, which supports currently a selection of frequencies from 1Hz up to 2MHz. The output of the 5V signal is done with a 680Ω resistor to test port TP2. You can use the GND signal from the minus pin of the zener diode extension or the test port TP1. The test port TP3 is connected to GND with a 680Ω resistor. Of course you can also connect a driver circuit to the ATmega port PB2 with a separate output driver circuit and a additional output terminal. But the input of this circuit should not insert a big capacity load for the ATmega output.

2.2.4 Frequency measurement

For using the with the dialog selectable frequency measurement is a little hardware extension necessary. The input pin PD4 (T0/PCINT20) of the ATmega is used for the frequency measurement. The same pin is also used for the connection of the LCD. With normal layout, the PD4 pin is connected to the LCD-RS signal, with the strip grid design it is connected to LCD-D4. For both signals the PD4 pin can be switched to input as long as no output to the LCD is required. The LCD respect the input value only, if the LCD-E signal is switched to GND. For driving the input pin from external clock source at least one serial resistor of 270Ω should be used. Better you should use the circuit of figure 2.5 . The voltage at the PD4 pin (LCD-RS or LCD-D4) should be adjusted to $2.4V$ without the assembled ATmega or during frequency measurement of the ATmega, to get the best sensivity for the input frequency signal. The LCD should always be be installed for adjusting, because the pull up resistor of the LCD change the voltage.

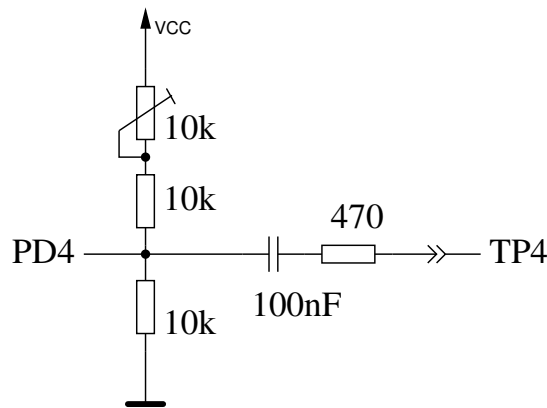


Figure 2.5. Extension for measurement of frequency

2.2.5 Using of a rotary pulse encoder

For easier control of the menu function for the ATmega328 you can expand the circuit by a rotary pulse encoder with a push button. The circuit 2.6 shows a standard expansion for a normal character LCD. All signals for the connection of the rotary pulse encoder are available at the plug connector for the LCD. For this reason the expansion can also be easy upgraded for many existing testers. In many cases the graphical LCD is assembled on a adapter board with some connection pins of the character LCD. Also in this cases the upgrade with a rotary encoder is not very difficult to do.

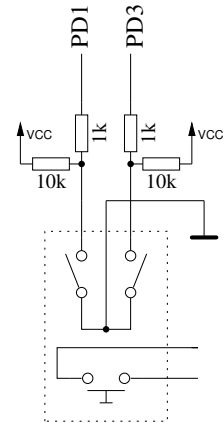


Figure 2.6. Expansion with a rotary pulse encoder

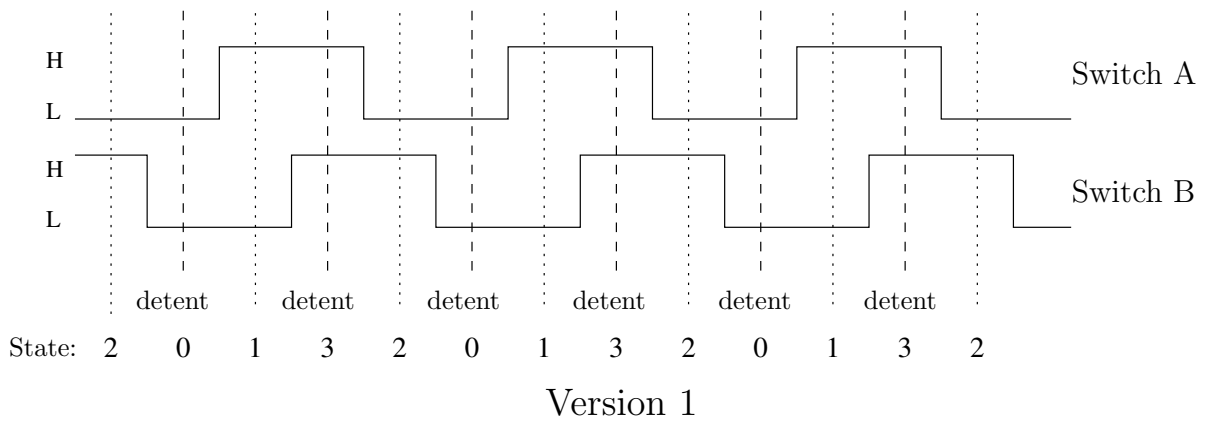
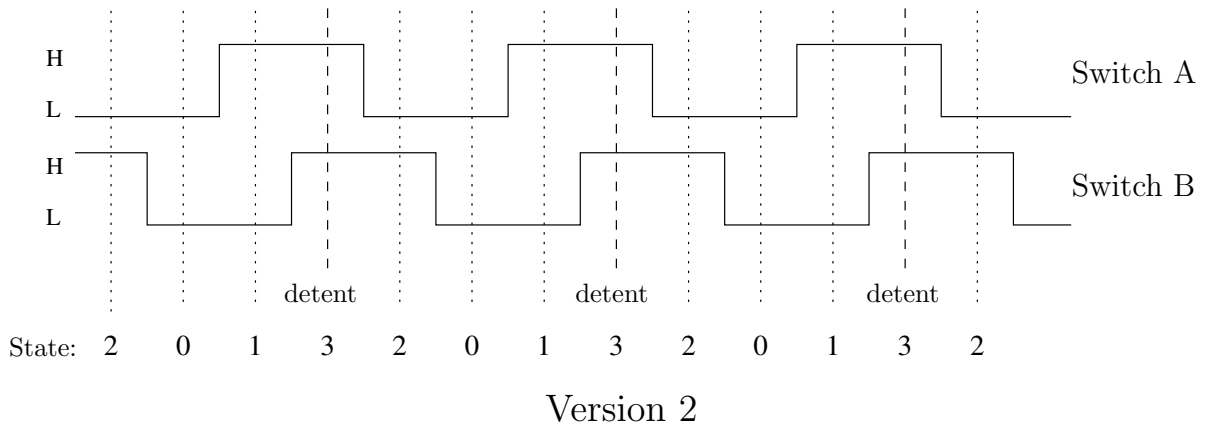
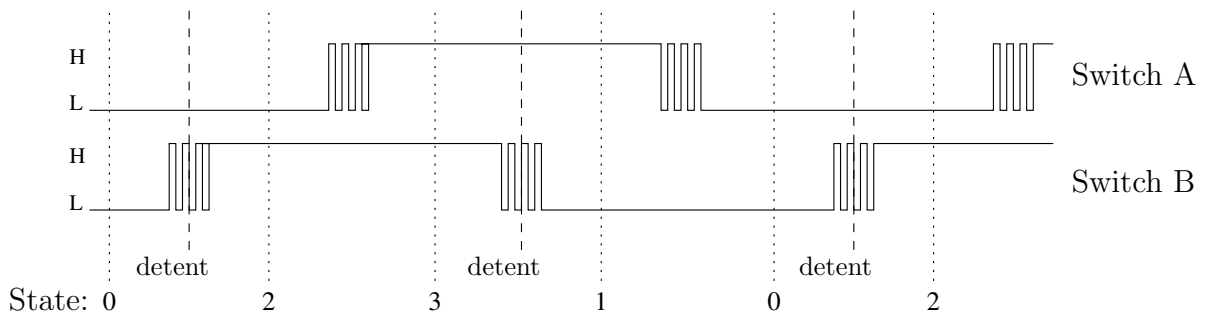


Figure 2.7. Feature of two different rotary pulse encoder

The figure 2.7 shows the feature of two different rotary pulse encoder. The version 1 has twice as much indexed positions (detents) per turn as pulses per turn. The version 2 has the same count of detents as pulses per turn. The slope of one of the two switch signals is sometimes exactly at the detent of the rotary pulse encoder.



Possible state history from left to right:

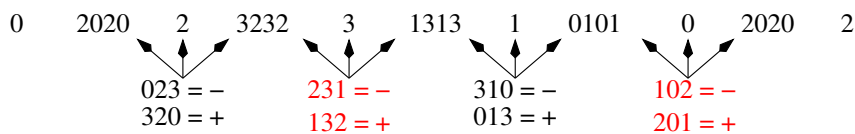


Figure 2.8. A rotary pulse encoder with bouncing switches

The figure 2.8 shows a rotary pulse encoder, which has not only bouncing contacts, but also has a unstable state of one of the switches at the indexed position (detent). Every change of the state of the switches is detected by the program and saved in a cyclic buffer. Therefore the last three

states of the switches can be checked after every status change. For every cycle of switch states a total of four sequences of state can be defined for every direction of rotation. If there is only one indexed position (detent) for every cycle of switch states, only one of the state sequence pair must be monitored (`WITH_ROTARY_SWITCH=2` or `3`) for correct counting. If there are two indexed positions for every cycle of the switch states, as shown in figure 2.8, you must monitor two pairs of switch state sequences (`WITH_ROTARY_SWITCH=1`). You can choose any resolution for the `WITH_ROTARY_SWITCH` for a rotary pulse encoder without indexed positions. A value of 2 or 3 selects the lowest resolution, 1 selects a middle resolution and 5 selects the highest resolution. A oscillation of the selection (counter up, counter down) can be avoided with the type of monitoring, but sometimes a count can be missed with a bad placement of the indexed position. In place of the two switches of a rotary switch encoder you can also install two separate key buttons for Up and Down movement, if no rotation switch encoder is present or favored. In this case the option `WITH_ROTARY_SWITCH` must be set to 4 for a correct handling of the program.

2.2.6 Connection of a graphical display

Many thanks to Wolfgang Sch. for his work to support a chinese version with ST7565 controller. By now you can also connect a graphical 128x64 pixel LCD with a ST7565 controller. Because the ST7565 controller is connected with a serial interface, only four signal lines are required. Two port D pins of the ATmega can be used otherwise. The ATmega processor should have at least 32k flash memory for the support of a graphical display. The ST7565 controller works with a operating voltage of $3.3V$. Therefore a additional $3.3V$ voltage regulator is required. The data sheet of the ST7565 controller does not allow the direct connection of input pins to $5V$ signal level. So the extension of figure 2.9 uses a additional 74HC4050 CMOS device for the conversion of the voltage level. You can try to exchange the four 74HC4050 gates with four resistors of about $2.7k\Omega$. With the voltage drop at the resistors you will prevent the increase of the $3.3V$ controller power voltage across the diodes of the controller inputs from the $5V$ ATmega outputs signals. You have to try, if the signal form with the resistors can be accepted from the ST7565 controller inputs. The signal form at the controller inputs will be more similar to the output shape of the ATmega with the 74HC4050 gates in any case.

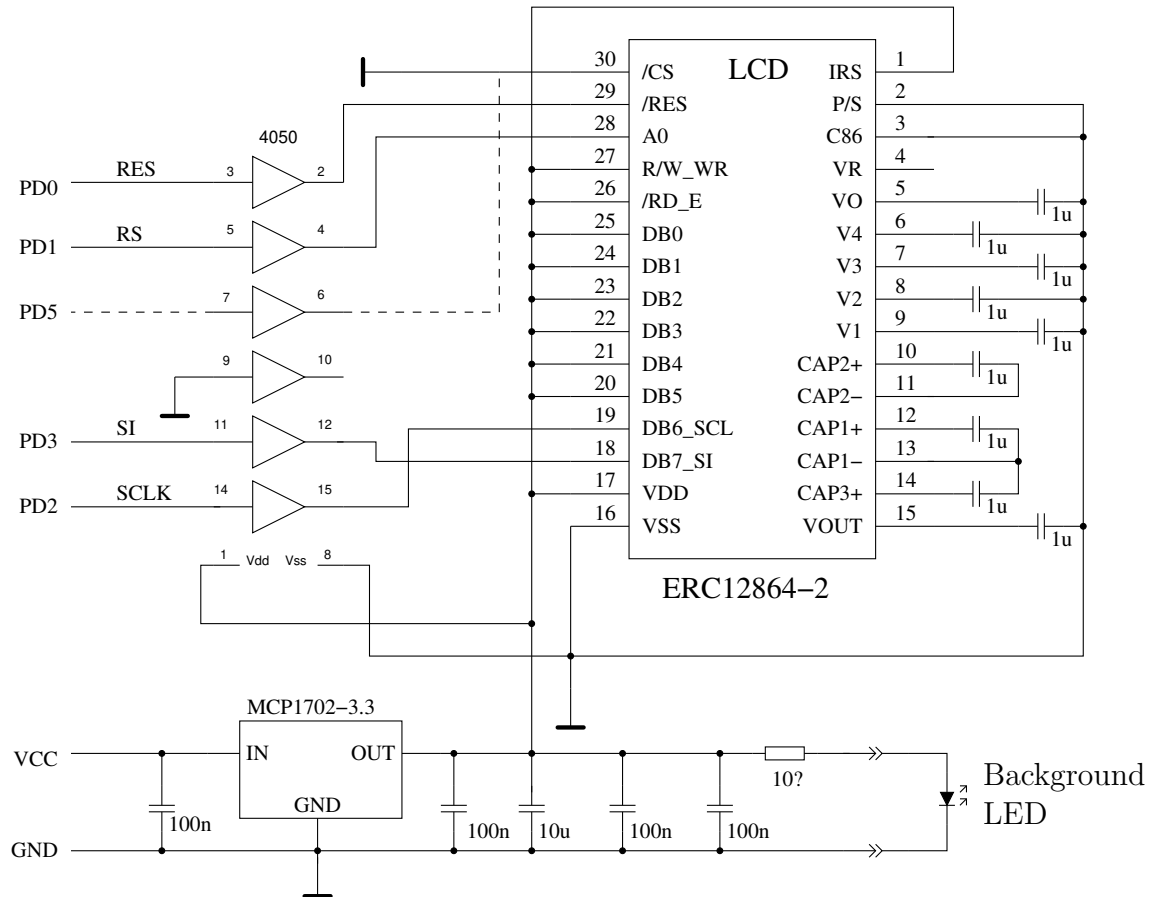


Figure 2.9. Connection of a graphical display with ST7565 controller

The table 2.3 shows more alternatives to connect a ATmega328 or other processors with the SPI (LCD_INTERFACE_MODE=4) or the 3LINE (LCD_INTERFACE_MODE=3) interface. The different connector lists for one processor type can be selected with the Makefile Option STRIP_GRID_BOARD. The connector pin list is defined in the file config.h. If you need a different wiring list, you should choose a new code number for the option STRIP_GRID_BOARD and supplement the setting in the config.h file.

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

Table 2.3. SPI wiring facility for different processors

Normally the ST7565 or SSD1306 controller is connected with a 4-wire SPI interface. But with the SSD1306 controller you can also use a I²C interface with PD2 as SDA and PD5 as SCL signal. The SDA and SCL signals must be equipped with pull-up resistors of about 4.7k Ω to 3.3V. A solution for connecting the OLED display shows the figure 2.10. The outputs of the ATmega is only switched to 0V for the I²C signals. Before connecting the pull-up resistors to 5V you must check, if your display module can tolerate a 5V signal level. Normally the data inputs of the controller are protected with diodes to 3.3V. You should make sure, that you have loaded a program with the I²C support to

the ATmega before the Display is connected. If you have loaded a program with another interface, the output are also switched to the 5V side. Because I have determined a influence to the tester results across the VCC connection of a OLED module, a additional decoupling with a serial resistor of 68Ω with a additional $10\mu F$ blocking capacitor is recommended. Instead of the 68Ω resistor you can also use a inductor with about $1mH$. Without the additional filter my tester has reported collector residual currents with bipolar transistors and a OLED display. Also you should check the pin sequence of your OLED module, some moduls have a different location of GND and VCC.

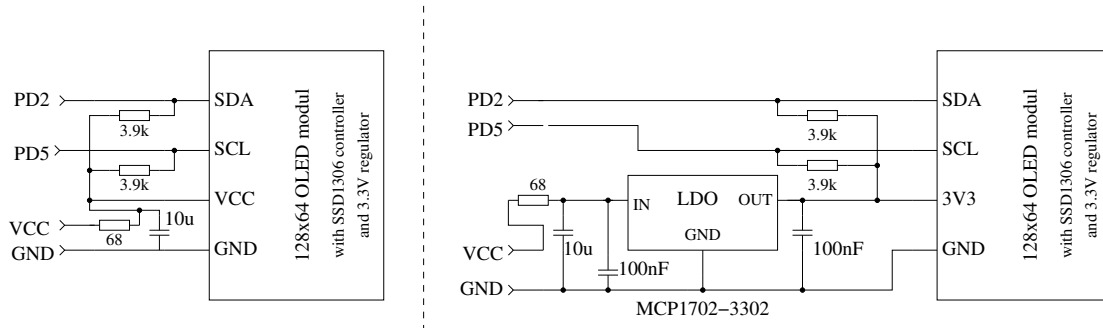


Figure 2.10. Connection of a graphical OLED display with I²C interface

For connection to the ATmega644 series of processors the pins PB3 (SCL) and PB4 (SDA) instead of PD5 and PD2 are used. For the ATmega1280 processor series the pins PA5 (SCL) and PA4 (SDA) are used. The exchange of a text display to a graphical display is possible with a adapter printed board because all required data signals and power signals are available at the LCD connector.

A little simpler is the connection of a graphical display with a ST7920 controller because the controller can operate with 5V power voltage. For that the display should offer 128x64 visible pixels. The display module with the ST7920 controller can be connected with the 4-bit parallel interface or with a special serial interface., as shown in figure 2.11.

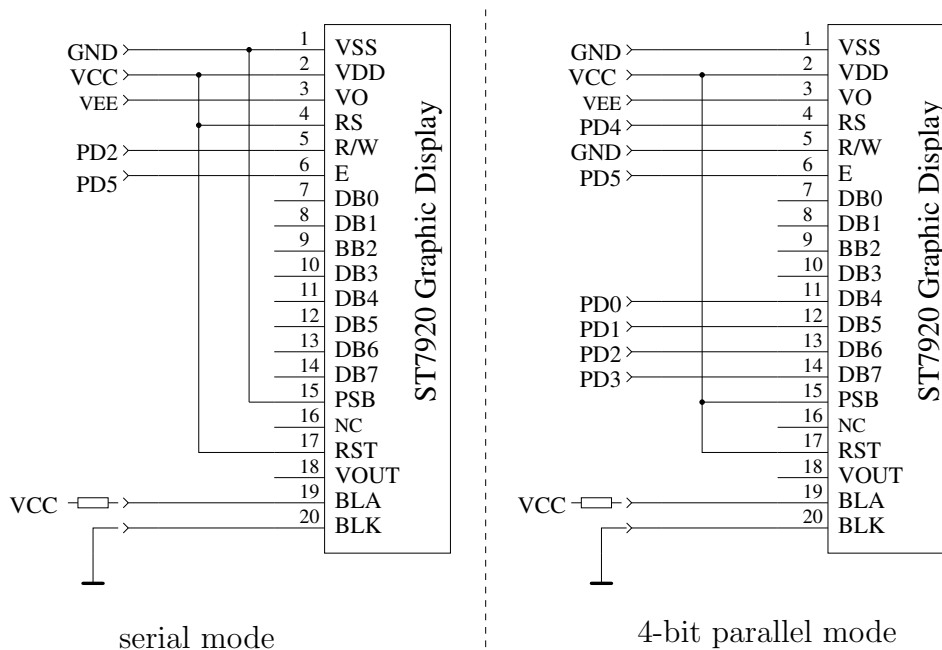


Figure 2.11. Connection of a display with ST7920 controller

For both connection types the software must be configured in a special way. The Makefile option "WITH_LCD_ST7565 = 7920" must be set in any case, for the serial connection type you must set also the option "CFLAGS += -DLCD_INTERFACE_MODE=5". The table 2.4 shows the connection of the serial signals with the INTERFACE_MODE 5 (ST7920) and 7 (SSD1803) for different processors.

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

Table 2.4. Serial connection for different processors

The orientation of the presentation can be changed with the options LCD_ST7565_H_FLIP and LCD_ST7565_V_FLIP in the same way as with the other graphical displays.

A special case is the connection of displays with a NT7108 (KS0108, S6B0108) controller. Because these displays can only use the 8-bit parallel interface, you must use a serial to parallel converter. The simplest way seems to be the use of a 74HCT164 or a 74HCT595 chip. A suitable suggestion of a connection circuit shows the figure 2.12 .

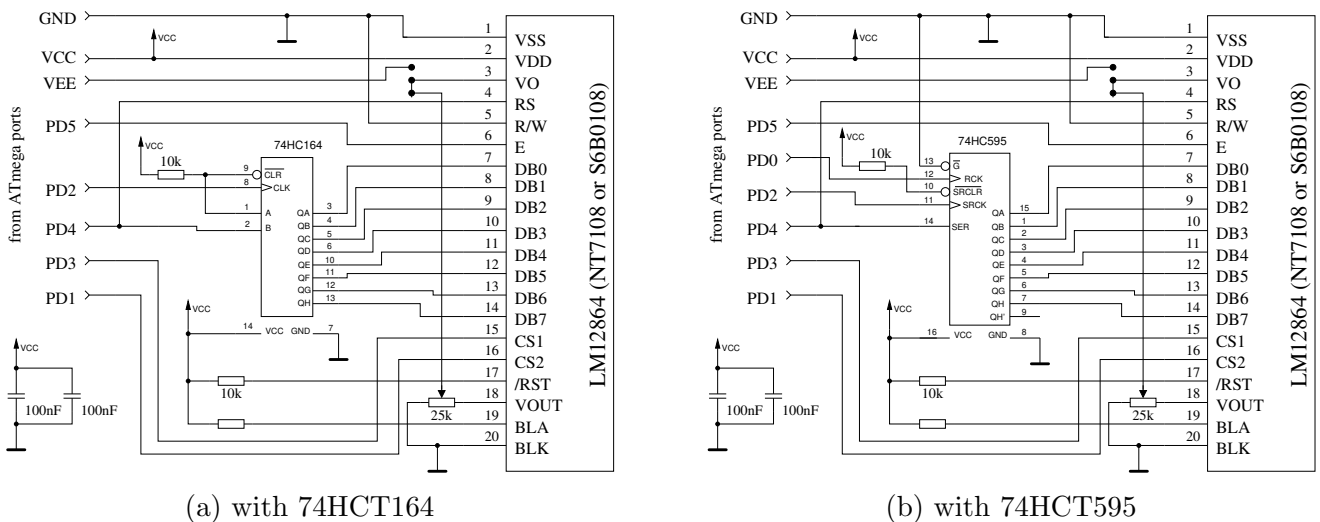


Figure 2.12. Connection of a graphical display with a NT7108 controller

You must check the pin layout of your LCD module, some moduls have different signal sequence. Some different pin layouts found in data sheets for the ABG128064 series are shown in table 2.5.

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

Table 2.5. Pin layout of different NT7108 moduls

The table 2.6 shows the connection of a serial NT7108 controller for different processor series.

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

Table 2.6. Serial NT7108 connection for different processors

You can also use a display with a PCF8814 controller, which is usually used in a Nokia 1100 handy for example. You must check, which interface of the controller is used in your display module. The PCF8814 controller can support the SPI-interface with 3 lines and 4 lines, the I²C-interface and a special 3-line interface, which receive the data/instruction signal as first serial bit before the 8 data bits. Because the display support only 96x65 pixels, the big icons for transistors are not used with this controller. The output is similar to the character display. As with the most graphical displays this controller also operated with 3.3 Volt. Therefore a signal conversion for the 5V outputs of the ATmega is required. For the SPI interface and the 3-line interface you can use the outputs of the ATmega like “open collector” outputs with the Makefile option LCD_SPI_OPEN_COL. You must use “Pull-Up” resistors with this option or you should not set the option PULLUP_DISABLE in the Makefile. Currently only the 3-line interface is tested with the PCF8814 controller.

Port	PCF8814 SPI	PCF8814 3-line	PCF8814 I ² C	additional function
PD0	LCD-RESet	LCD-RESet		
PD1	LCD-D/C	LCD-SCE		rotary encoder 2
PD2	LCD-SCLK	LCD-SCLK	LCD-SDIN	
PD3	LCD-SDIN	LCD-SDIN		rotary encoder 1
PD4				frequency counter
PD5		LCD-EN	LCD-SCLK	

Table 2.7. Pin assignments for different interface types of the PCF8814 controller

Some code for supporting a PCF8812 controller with 102x65 pixels is also implemented, but not yet tested.

2.2.7 Connection of a graphical color display

Some Chinese salesmen offer cheap color display modules with a SPI interface. The picture 2.13 shows the backsides of two supported modules with 128x128 pixel and 128x160 pixel. The modules are very small, so the shown text and symbols are very little. But the appearance is sharp and clear.



Figure 2.13. Back side of two Color-LCD's

The 128x128 pixel modul uses a ILI9163 controller. The 128x160 pixel modul uses a very similar ST7735 controller. The modules are tested with a adapter board, which connects the SPI signals and the power to the terminal for the normal character display. The adaption of the 5V output signals of the ATmega to the 3.3V inputs of the controller is ensured with serial 10k Ω resistors. The background light (LED) is required for this modules because nothing is readable without it. Because of the high count of pixels in vertical direction you can display more text lines for this modules. For the 128x128 pixel display you can display 8 lines of text with a 12x8 font, for the 128x160 pixel display

you can even show 10 lines of text. At the picture 2.14 you can see the result of the measurement of a Germanium transistor at a 128x128 pixel display.

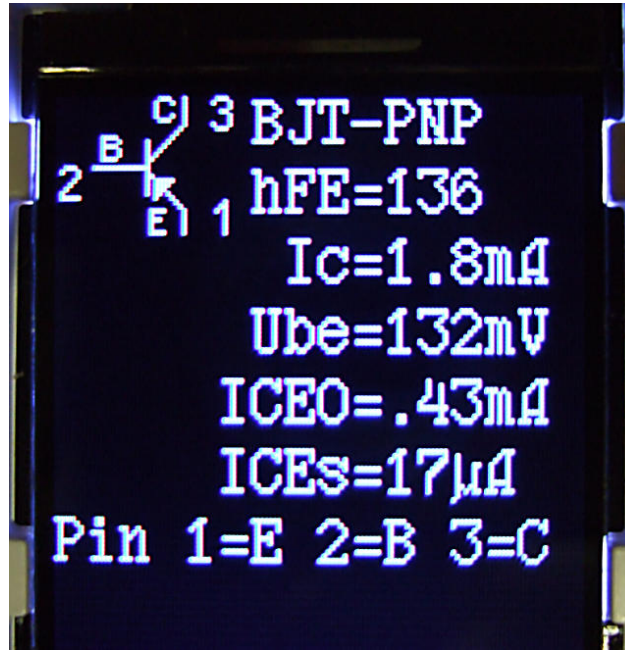


Figure 2.14. Measurement of a bipolar PNP transistors

The color feature of the modules is currently not used. Simply the background color and the foreground color can be changed in the file `lcd_defines.h` or in the Makefile. The 16-bit color model of the controller is used by the software, the foreground color can be changed with the constant `LCD_FG.COLOR` and the background color with the constant `LCD_BG.COLOR`.

2.3 Hints for building the TransistorTester

Every LCD-display with at least 2x16 character 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. Also loading of special characters for displaying the resistor symbol has caused problems with the OLED.

The resistors R1 to R6 are critical for measurements and this 680 Ω and 470k Ω 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, ATmega168 and ATmega328 can be used. Recommended is a 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 μA 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 sure, 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 sure 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!

2.4 Changeover for tester versions designed by Markus F.

Voltage control If the problem exist, the tester will shut down immediately with every switch on. With my suggested setting of the fuses (Makefile) the voltage control of the different ATmega versions is switched to 4V (brown out level). This may be the reason why the tester makes trouble with the power on sequence. The Pin PD6 tries to switch the 100nF capacitor C2 to VCC level causing a voltage breakdown of the VCC voltage (5V). The capacitor C2 can be reduced to $< 10nF$ without problems. If possible, the direct connection of PD6 should be replaced by a resistor $> 220\Omega$.

Improvement of power on circuit Often this problem is the reason, if the tester starts with the button hold pressed, but switch off directly by releasing. The problem is enforced by a high current background light for the LCD. The resistor R7 to the base of the PNP transistor T3 was optimized with the value $27k\Omega$ too much to save power consuming. To improve the switching with lower battery voltage or lower current amplification factor of the PNP transistor T3, you should reduce the resistance to $3.3k\Omega$.

Additional pull-up resistor at PD7 The missing pull-up resistor results to a switch off of the tester with the message "Timeout" after a short display time. The software is configured with the option PULLUP_DISABLE, that all internal pull-up resistors are switched off. For that reason the voltage of pin PD7 is not defined, if the level is not switched by the push button or transistor T2 to GND. One external pull-up resistor of $27k\Omega$ to VCC avoid this error.

Capacitor C1 at the AREF pin Many designs use a 100nF capacitor at the AREF pin, like the design of Markus F. too. As long as the reference voltage of the ADC is never changed, this is a good solution. The software of the TransistorTester for the ATmega168/328 uses a automatic selection of the internal 1.1V reference voltage of the ADC, if the input voltage is below 1V. With this solution a better resolution of the ADC can be reached for little input voltages. Unfortunately the switching from 5V to 1.1V reference is very slowly. A additional wait time of 10ms must be respected for this reason. With changing the capacity value to 1nF this wait time can be reduced significant. I have not noticed any degradation of measurement quality with this change. Even a removing of the capacitor has no significant change of measurement results. If you prefer to leave the capacitor unchanged, you can remove the option NO_AREF_CAP in the Makefile to activate longer wait times in the program.

Expanding of a 8MHz crystal With some skill you can expand a 8MHz crystal to the backside of the printed board directly to the pins PB6 and PB7 (pin 9 and pin 10). My own expansion was done without the both 22pF capacitors. This solution has operated well with all tested ATmega. But it is not required to use a crystal. You can still use the 8MHz RC oscillator by setting the fuses to get the better resolution of time constant for measuring the capacity values.

Smoothing of the operating voltage The original circuit of Markus F. shows only one 100nF capacitor to block the VCC voltage. This is clearly too little smoothing. You should at least

use one $100nF$ near the ATmega power pins and one near the voltage regulator. The input of the voltage regulator should be blocked with a $100nF$ too. Additional $10\mu F$ capacitors (electrolytic or ceramic) at the input and output of the voltage regulator can stable the voltage level. Ceramic $10\mu F$ capacitors with SMD mounting form are easier to use for backfitting and have usually a lower ESR value.

Selection of the ATmega processor The using of the base function of the tester is still possible with a ATmega8. The flash memory of that device is used near 100%. Because the ATmega168 or ATmega328 processors are pin-compatible to the ATmega8, I can recommend the replacement. Actually the price for ATmega328 is so cheap, that there is no reason to take a ATmega168 type. With a ATmega168/328 you get the following advantages:

- Self test with automatic calibration.
- Improvement of measurement quality by automatic switching of ADC scale.
- Measurement of inductors with resistance below 2100Ω .
- Measurement of ESR value of capacitors with value of above $20nF$.
- The resolution of resistor measurement below 10Ω is 0.01Ω .
- Using of pin PC4 as serial output.
- Using of a precision voltage reference.

Usually the software should detect the missing voltage reference with the unconnected pin PC4. In this case no $VCC=x.xV$ message should appear in row 2 of the LCD on power on. If this message appear without the reference, you should connect a $2.2k\Omega$ resistor to the PC4 input and VCC.

2.5 Extended circuit with ATmega644 or ATmega1284

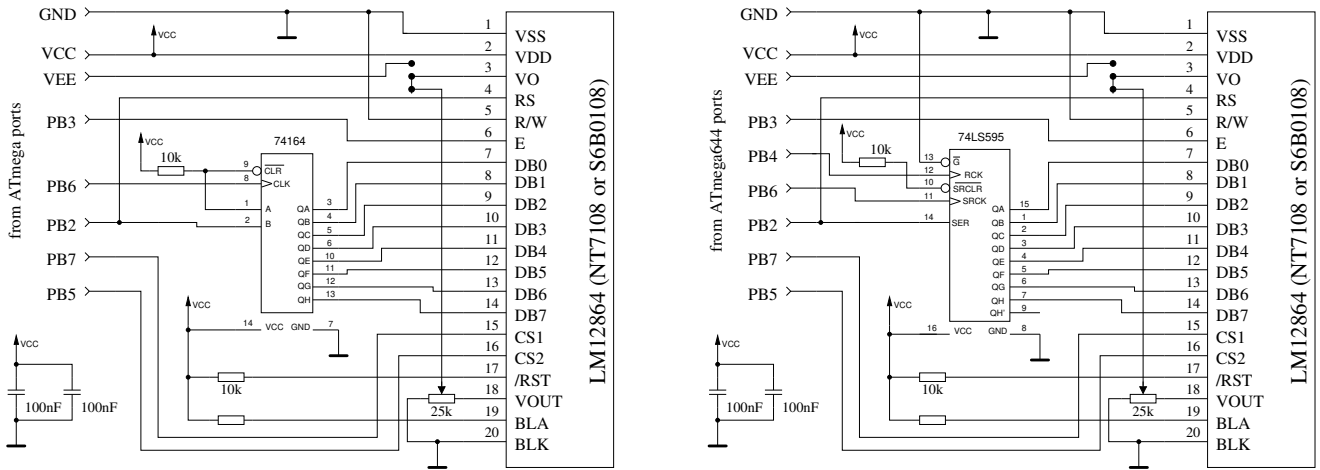
A extended circuit for ATmega644/1284 processors was developed with Nick L. from the Ukraine. The circuit 2.16 enables a additional test of crystals and a extended range for the frequency measurement. Although the basic circuit is very similar to the circuit 2.1, the port assignment is different. A rotary pulse encoder with circuit 2.6 can be connected here at the pins PB5 and PB7 (instead of PD1 and PD3). Both signals and also the power signals VCC and GND are available at the ISP connector, so that the extension can also be connected here.

The 16:1 frequency divider of the 74HC4060 is allways used for frequencies above $2MHz$. The frequency divider can also be used for frequencies between $25kHz$ and $400kHz$ to upgrade the frequency resolution by using the period measurement. For switching between the operational states (frequency divider and crystal oscillator) the analog switches 74HC4052 are used. The table 2.8 shows the pin assignments for the ATmega324/644/1284 microcontrollers for different display connections. The using of the I²C interface is only possible with the SSD1306 controller. The signals of the I²C interface require a pull-up resistor of $4.7k\Omega$ to $3.3V$. The outputs of the ATmega is only switched to $0V$ for the I²C signals.

Port	Character LCD	Graphic LCD SPI 4-wire	Graphic LCD I ² C	additional function
PB2	LCD-RS			
PB3	LCD-E	(LCD-CE)	LCD-SCL	
PB4	LCD-D4	LCD-REST	LCD-SDA	
PB5	LCD-D5	LCD-RS		ISP-MOSI Rotary encoder 2
PB6	LCD-D6	LCD-SCLK		ISP-MISO
PB7	LCD-D7	LCD-SI		ISP-SCK Rotary encoder 1

Table 2.8. Different variations of the display port assignments

You can also connect a display with the NT7108 (KS0108, S6B0108) controller to the ATmega644 or ATmega1284 by using a little circuit as shown in figure 2.15. You should also respect the different pin assignments of display modules with NT7108 controllers as shown in table 2.5 at page 19.



(a) with 74HCT164

(b) with 74HCT595

Figure 2.15. Connection of a NT7108 Controller to a ATmega644/1284

[illegible]

Of course you can connect all supported displays also to the ATmega1280 or ATmega2560 as shown in table 2.9 .

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

Table 2.9. Connections for different display to the ATmega1280/2560 processors

2.7 Chinese clones with text display

As I know, the tester is rebuild in China in two versions with character displays. The first model is rebuild from the first design of Markus F. without the ISP port. The assembled ATmega8 is placed in a socket, so you can replace it with a ATmega168 or ATmega328. For this version you should consider all the hints of section 2.4. Additional $100nF$ ceramic capacitors should be connected near by the VCC-GND and AVCC-GND pins of the ATmega for better stabilization of the power voltage. Because there is no ISP connector at the board, you must expand the board with a ISP connector or you must plug the ATmega to a external socket for programming. In addition you should notice, that if your ATmega should run with a $8MHz$ crystal, your external ISP programmer must have a external clock for programming or a crystal mounted at the socket.

The second version of rebuildted tester is build with SMD components. Also the fix installed ATmega168 is a SMD type with 32TQFP body. Fortunately on the board is a 10-pole ISP connector provided for the programming. I have analysed the board version “2.1 2012/11/06”. One error is the assembly of the part “D1”, which should be a precision $2.5V$ voltage reference. Assembled is only a zener diode. This part should be removed. You can mount a LM4040AIZ2.5 or LT1004CZ-2.5 precision voltage reference at this place. A missing voltage reference is noticed by the software, so that you must not install the voltage reference. My exemplar was delivered with software version 1.02k. The 10-pole ISP plug was not assembled and I must install a jumper from ISP pin 6 to ISP pin 10. My programmer expect a GND connection at pin 10, but the board has GND level only on pin 4 and pin 6 of the ISP. The label of the ATmega168 was rub away and there was no documentation delivered with the part. The lock fuses of the ATmega were set, so no readout was possible. But I could install the software version 1.05k without any problems. Another user has problems with the same software version 1.05k. This user has the chinese board “2.2 2012/11/26”. The software runs only without problems, if a additional $100nF$ keramic capacitor was placed between the pin 18-AVCC and 21-GND near by the ATmega. The software 1.05k uses the sleep state of the ATmega for waiting time. For this reason the current alternates often and the voltage regulator is stressed more. Further I have noticed, that the VCC voltage is blocked with a $100nF$ ceramic capacitor and with a $220\mu F$ electrolytic capacitor nearby the 78L05 voltage regulator. The $9V$ supply voltage is blocked with the same capacitors, but not at the input of the regulator but at the emitter of the PNP transistor (parallel with the battery). The printed circuit board track from the ATmega168 to the test port is very thin, so that a resistance of $100m\Omega$ could be measured for one path. This will be the reason for measuring a resistance of 0.3Ω for two direct connected pins. The ESR measuring can usually consider this by zero compensation. Beginning with version 1.07k the software does respect this offset for measuring resistors below 10Ω too.

2.8 Chinese clones with graphical display

Newer rebuilds of the tester like a version from Fish8840 use a 128x64 pixel graphical display. This version use a modified circuit for the switch on logic. The figure 2.18 shows a part of the modified circuit.

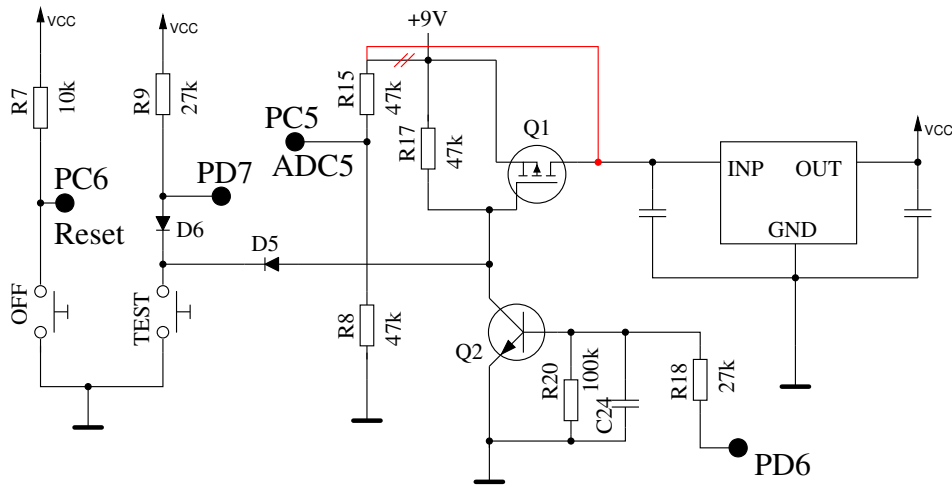


Figure 2.18. Part of the circuit from the Fish8840 version

How you can see at the values of resistor R8 and R15, a 2:1 scaling factor for the battery voltage measurement is used instead of the original scaling factor. In addition R15 is direct connected to the battery, what results to a power consumption in the switch off state. The R15 should better be connected to the drain of Q1 or the input of the voltage regulator to prevent this unneeded battery power consumption. A suitable change of the printed board is shown in picture 2.19. A circuit path is cut between R17 and D5 and a new conductive path is inserted between Q1 and R15 with a enameled copper wire.

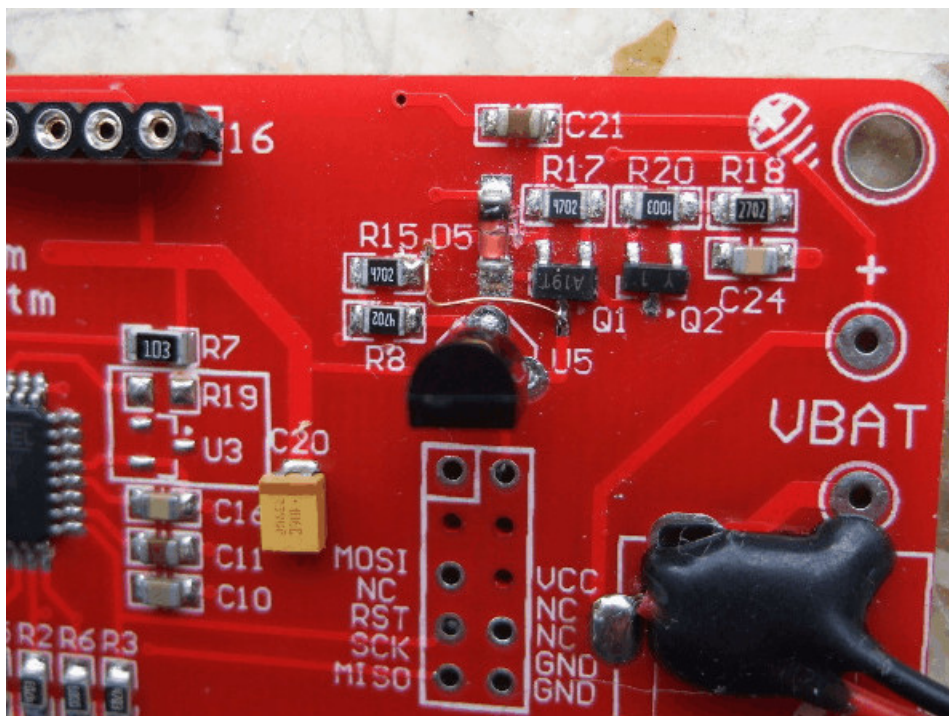
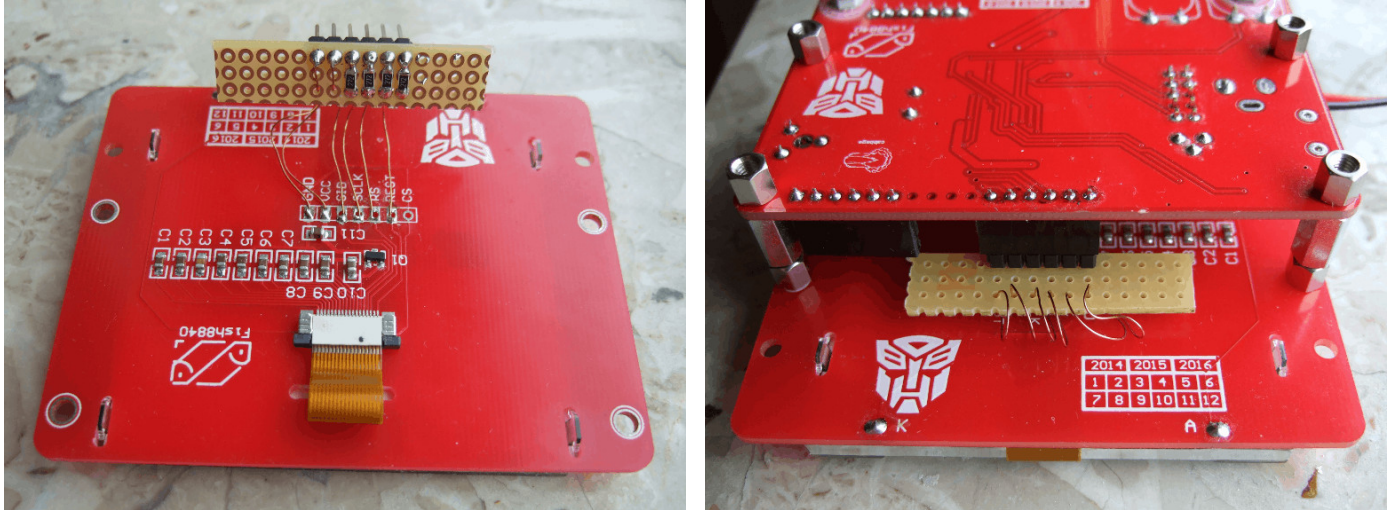


Figure 2.19. Picture of the changed Fish8840 printed board

The scaling factor of the battery voltage must be specified in any case in the Makefile before any attempt can be done to replace the original software (BAT_NUMERATOR=66 for example).

The display module of the Fish8840 tester is equipped with a 3.3V voltage regulator to adapt the operating voltage of the display controller. Because the 3.3V operating voltage can be increased with the 5V signal level of the data signals from the ATmega, a adapter circuit according to the picture 2.20 is recommended. The four data signal lines are equipped with four serial 2.7k Ω resistors at a little breadboard. Longer spacer bolts must be used to mount the display with the adapter board to the printed board of the Fish8840 tester now.



(a) Display with the breadboard adapter

(b) Ready mounted Tester

Figure 2.20. Adapter for a correctly display connection

Instead of this modification you can also use a special output mode of the 4 SPI signals of the ATmega with the Makefile option `LCD_SPI_OPEN_COL`. With this option the outputs are not switched to VCC level, but the internal “pull-up” resistors are switched on during the output of a high level. If the option `PULLUP_DISABLE` is set, additionally a external pull-up resistor for the “Reset” signal (PD0) is required. Because the data signals are never switched to VCC directly, the 3.3V power of the LCD controller can not be elevated. My version of the Fish8840 tester has all signals for connecting a character display routed to the LCD plug connector. Therefore you can prepare the board for connection of a character display, if the female header is completed and the potentiometer for the contrast level is added. However the supply pin 15 for the background light is connected directly to the VCC level. If you install a character display, you should check, that your display module is equipped with a serial resistor for the background LED. Of course you have to adapt the software for different displays. Also the hardware extension are possible with the Fish8840 board.

All attempt to replace the original software is allways done at one’s own risk. No guaranty can be given for operational capability of newer software versions. Unfortunately the original state of the chinese software can not be saved because the security bits of the ATmega328 are set. So there is no way to get back to the original state.

A additional version with graphical display is the WEIM8 printed board, which is shown in picture 2.21. This build use a LiIon accumulator with AA (mignon) size as power supply, which can be loaded with the micro USB plug. Operating is also possible without the accumulator, only with the USB power.

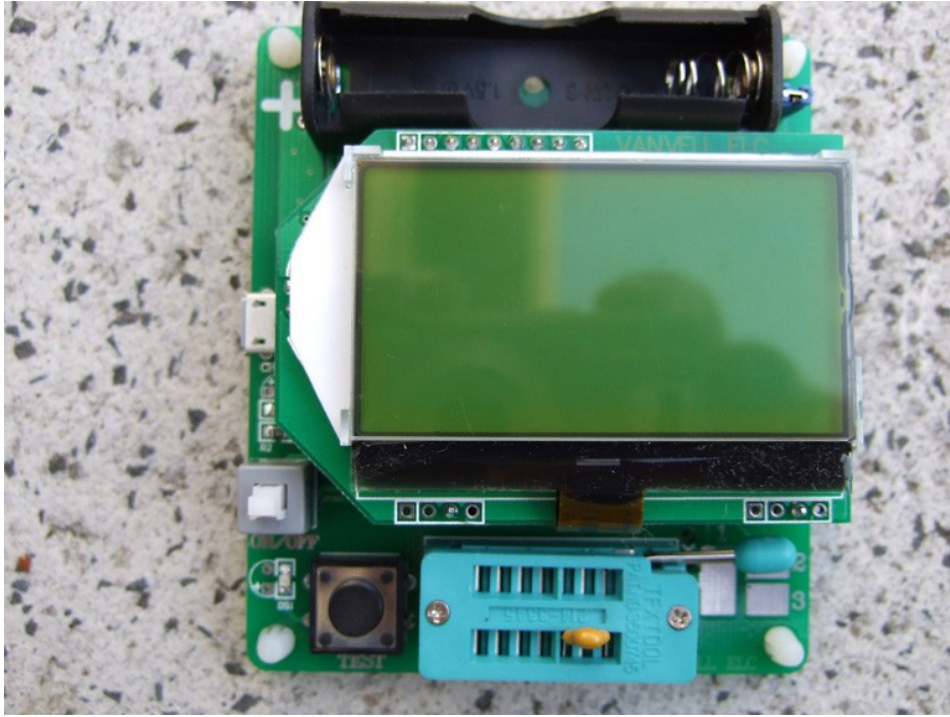
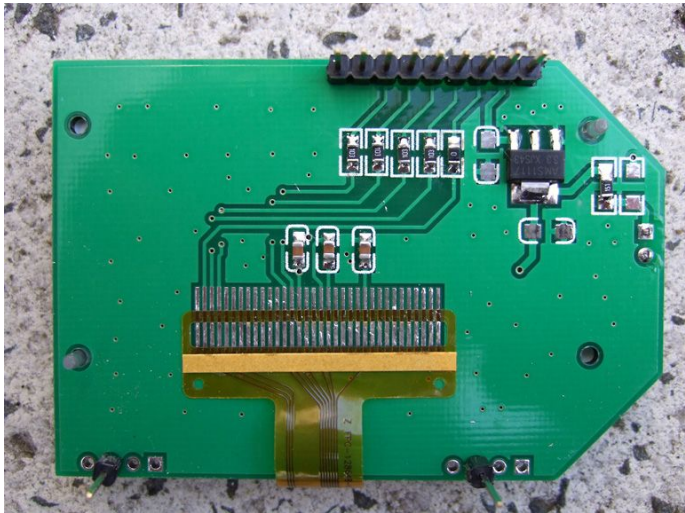
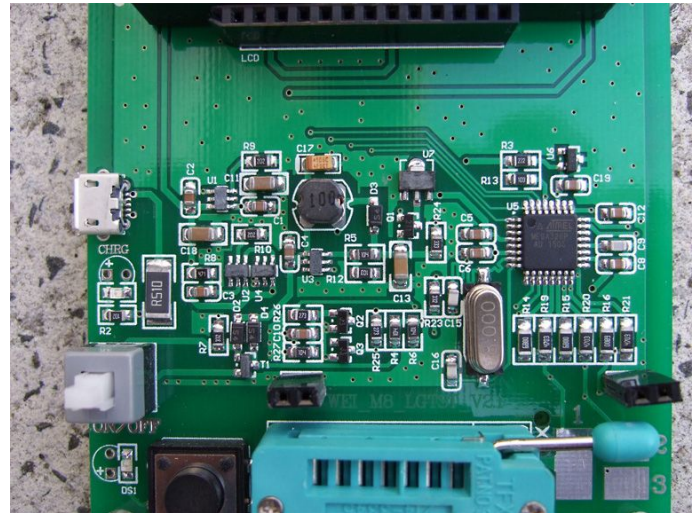


Figure 2.21. Picture of the chinese WEILM8 tester

It is pleasant, that the signal lines of the display are equipped with serial resistors at the adapter board, as you can see in the left picture of 2.22. Therefore you must not be afraid that the 5V signals of the ATmega can cause a excessive increase of the 3.3V power of the display controller.



(a) Adapter board for the display



(b) Base board

Figure 2.22. Inside look to the WEILM8 Testers

With the upgrade to version 1.12k some problems has occured. If the extended fuse is set to 0x04 (0xfc) as recommended, some measurements has caused a “Brown Out” reset of the processor. This resets are caused by short voltage drops of the VCC power. I have added a additional $4.7\mu F$ ceramic capacitor at the input side of the voltage regulator and a $10\mu F$ ceramic capacitor at the output side (VCC) of the regulator. Both before and after the upgrade I have noticed, that with bipolar transistors a residual collector current (I_{CEO} or I_{CEs}) of about $1\mu A$ was detected with this board. Not until the unknown LDO voltage regulator was replaced with a MCP1702-5002 this effect has disappeared. The figure 2.23 shows the modified board with the slantwise mounted capacitors and the MCP1702 regulator. If you don't want to modify your board, you must set the extended fuse to 0x07 (0xff) to enable a regular operation. With that setting the voltage drops remain undetected.

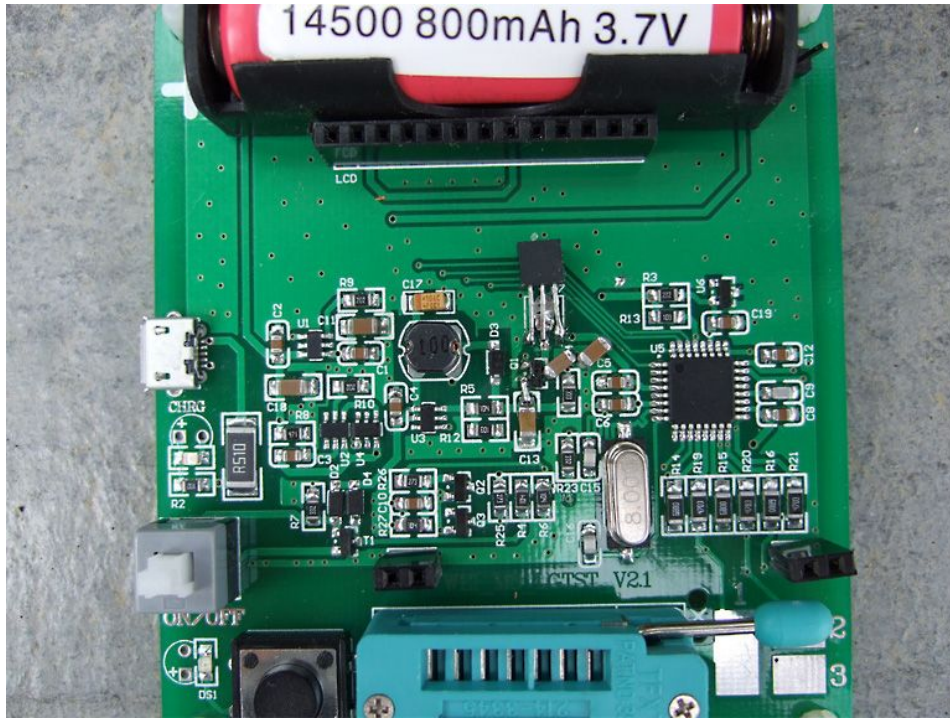
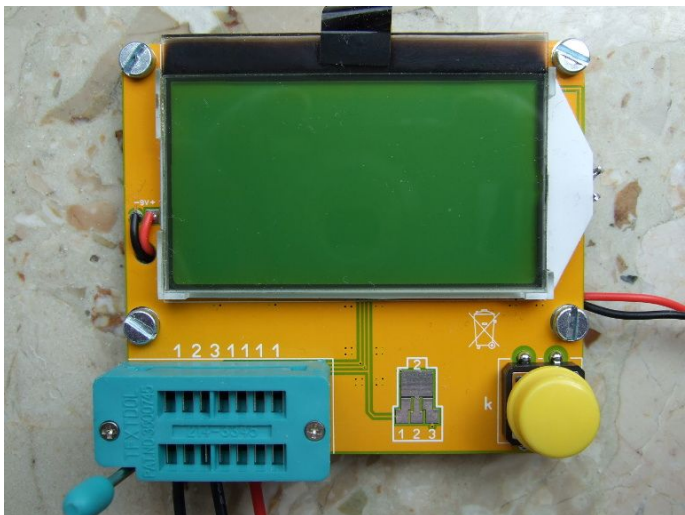
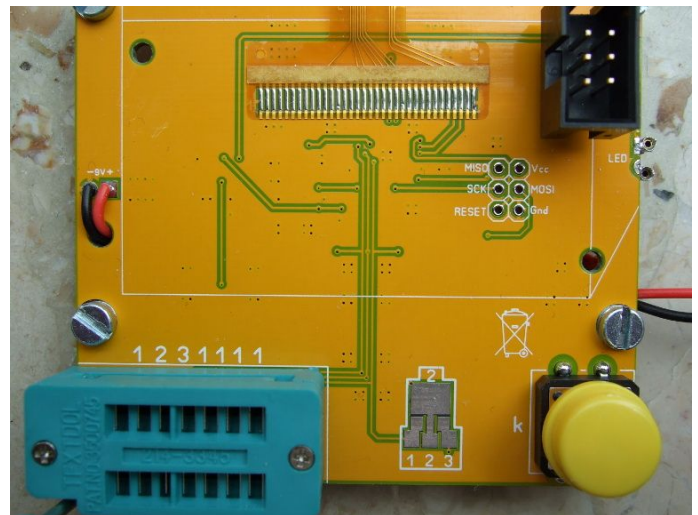


Figure 2.23. Picture of the modified WEI_M8 Tester

A additional chinese version with graphical display is the “LCD-T4” tester with a yellow printed board. I have removed the display for replacing the software with a newer version. At the right picture of figure 2.24 you can see in the right upper corner the ISP male connector with correct orientation beside the six holes of the board. For programming the ATmega I have not soldered the male connector. I have only pugged it into the holes instead and have fixed it with a finger during the programming. With that procedure the male connector can be easy removed an the display can be mounted again to it’s origin place. The chinese software could be replaced with the 1.12k without any noticeable problem. Also the activating of the “brown-out” monitor by setting the extended fuse has not shown any surprise.



(a) completely

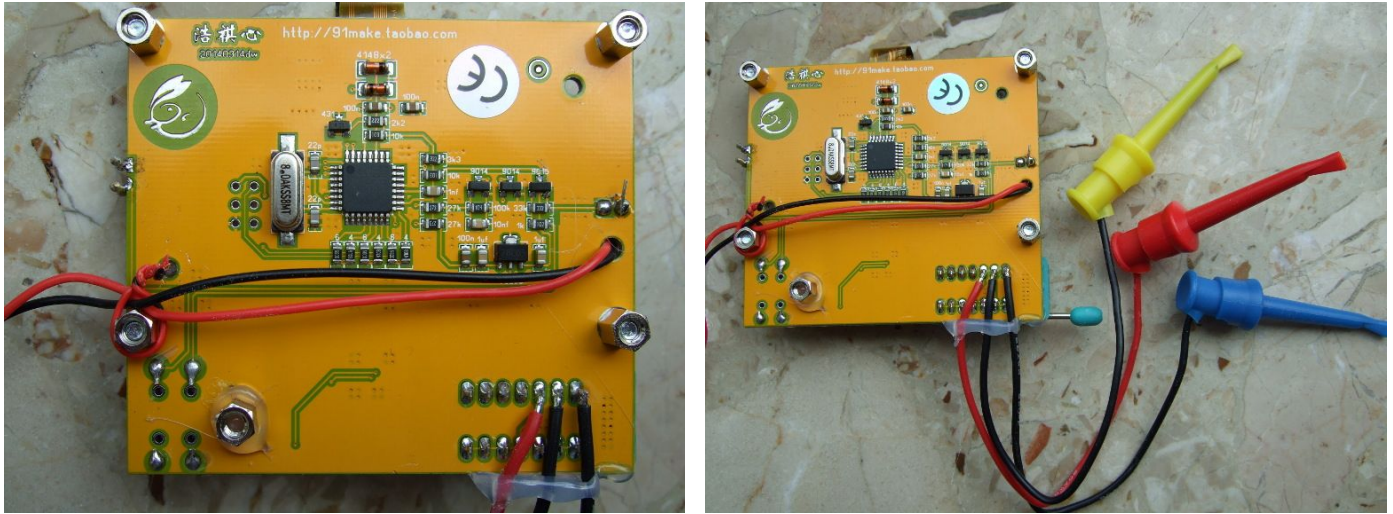


(b) with removed display

Figure 2.24. Front view of the T4 tester

You can see the upgraded 5mm threaded bolt and the upgraded cables with measurement clips at the pictures of the back side in figure 2.25. Because the data signals for the graphical LCD controller have also no level adaption (5V -> 3.3V), the setting of the option LCD_SPI_OPEN_COL

is recommended. Due to the fact, that the board can not be easily upgraded with a “pull-up” resistor, it makes sense to remove the option PULLUP_DISABLE instead. This board is less practical for later extensions, a replacement of the display is also difficult to realize.



(a) components side

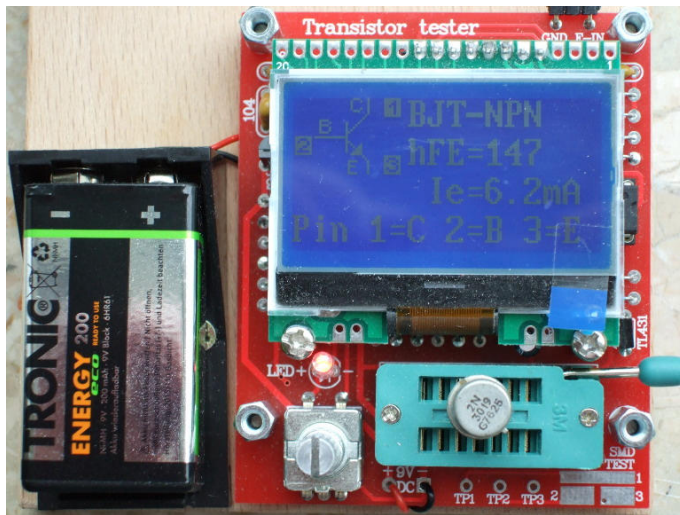
(b) with measurement cables

Figure 2.25. Back side of the T4 tester

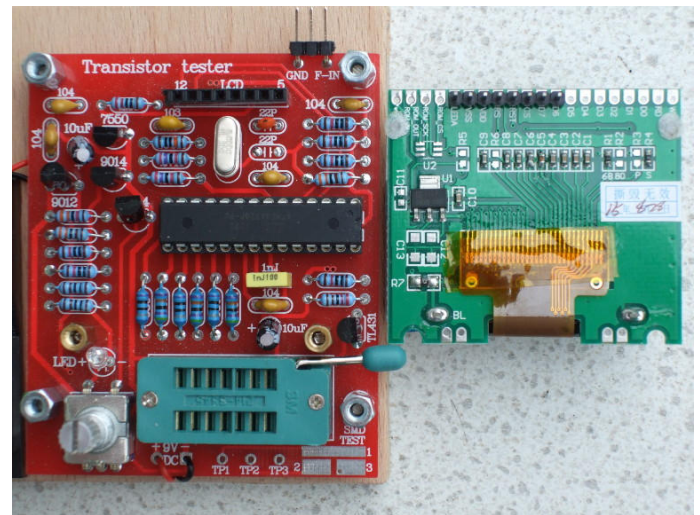
A additional chinese version with graphical display is offered with the name “GM328”. For this version the graphical display is connected with a adapter circuit board to the 16-pol female connector. The port PD5 of the ATmega is connected via pin 6 of the female connector to the CE (Chip Enable) input of the graphical controller. The signal CE is also connected at the adapter board to the 0V power (GND). This will result to a short-circuit if the ATmega will switch the PD5 output to 5V. But newer versions of the software also switch the CE signal, even if the signal is not necessary for correct operation. For correct operation of the “GM328” tester with newer software versions you should cut the connection of the CE signal to pin 6 of the male connector.

2.9 Chinese kits with graphic displays

Two different kit versions with graphical display and rotary encoder are known until now. The first released kit use a display with a ST7565 or compatible controller (128x64 pixels). In addition to the rotary encoder a additional input for frequency measurement is provided. For the test ports a 14-pin Textool socket, three soldering pads for cables and a testpad for SMD parts are provided at the main board. The pictures 2.26 shows the mounted kit. One of the both 22pF capacitors was replaced with a trimmer at the soldering side. With a trimmer the frequency of the crystal can be adjusted for better accuracy of frequency measurement and frequency generation.



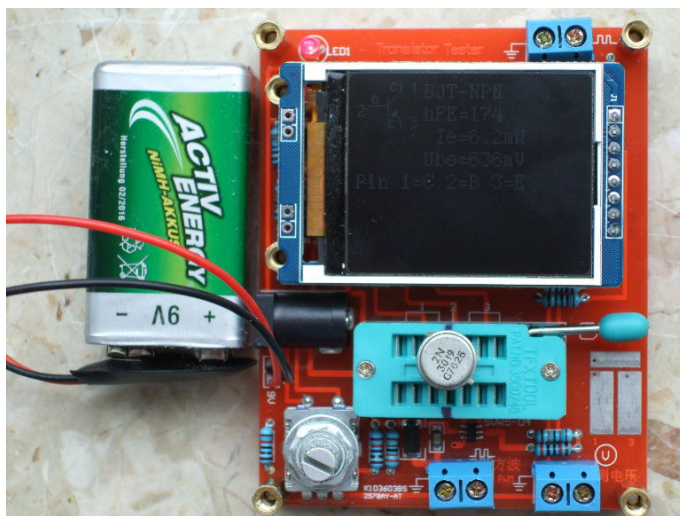
(a) build together



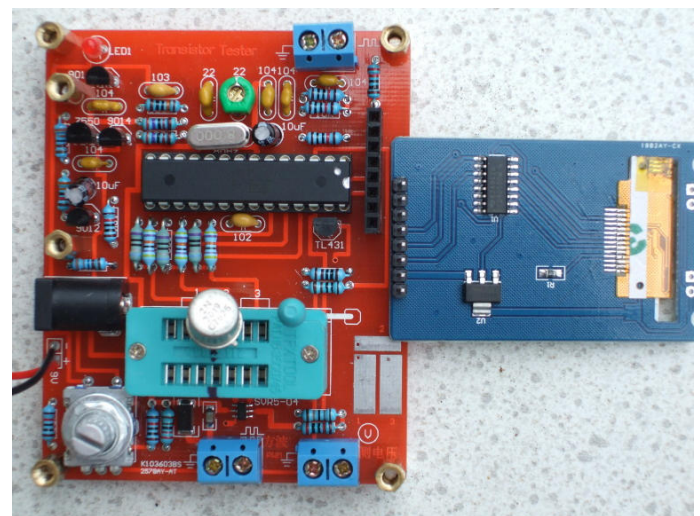
(b) with dismounted display

Figure 2.26. Assembled kit with 128x64 pixel display

The later released kit use a color display with a ST7735 controller (160x128 pixel) and is additionally equipped with a input for voltage measurement and a terminal for frequency output. But the frequency output terminal is not buffered, it is simply connected parallel to the TP2 terminal. The voltage measurement can handle positiv DC voltages up to 50V. A additional DC-DC converter for the measurement of Zener diodes is not provided. The pictures 2.27 shows this assembled kit. Also in this version was one of the both $22pF$ capacitors replaced by a trimmer (green color).



(a) build together



(b) with dismounted display

Figure 2.27. Assembled kit with 160x128 pixel color display

Both kits use a DIP version of the ATmega328P mounted at a sockets and are not equipped with a ISP connector for updating the ATmega with newer software versions. The first version use only leaded parts for the main circuit board. I have received measurement resistors 680Ω and $470k\Omega$ with 0.1% tolerance with the Chinese kit. Also a $220nF$ capacitor for calibration was added to the kit. The kit with the color display was equipped with a connector for a DC power supply additionally to the pads for the 9V battery. Some few SMD parts was allready assembled at the main board, so that also only simple soldering tasks are to do for this kit. A little disadvantage of the version with the color display is the speed of display output. Especially for the menu handling the slower output speed will be noticed. Otherwise the color display has a significant higher count of pixels, so more text can be shown at once. Both kits use a 3.3V voltage regulator for the power of the display controller at the full assembled display circuit boards. Only the pin header must be soldered at the

display circuit board. The color version use a CD4050 buffer to adapt the different signal voltage levels. I did not found any adaption of signal voltage levels at the board for the ST7565 version. Probably the selected controller version tolerates the 5V signal levels of the ATmega328. I could not find protection diodes at the input signals to the 3.3V supply side for this controller type.

2.10 And another clone from Hiland M644

This replica is based on the circuit diagram by Nick L. from Ukraine, see illustration 2.16 on page 25. **The tester is operated with a button that is both a button and an encoder.** It offers all of the the known measurement extensions:

- the frequency measurement
- the f-generator
- the 10-bit PWM
- rotary encoder
- the crystal measurement
- Voltage measurement and Zener diode measurement (up to nearly 50V).

The board is equipped with an 8MHz quartz. A 16MHz quartz would be a better selection. Retrofitting a trimmer for more accurate frequency measurement is difficult. The contacts for the ISP interface are on a 6-pin row of holes below the plug-in display module which is occupied as follows:

left to right: 1 -Reset; 2 -SCK; 3 -MISO; 4 -MOSI; 5 -+5V; 6 -GND.

To be able to update the tester, you need an adapted cable that you can relatively simple create yourself as shown in figure 2.28. When delivered, the zero force test base is connected to the board via socket strips.

In the tester shown below, the base was soldered directly and the socket strip saved with it soldered to an existing ribbon cable with post connector and fixed with shrink tubing.

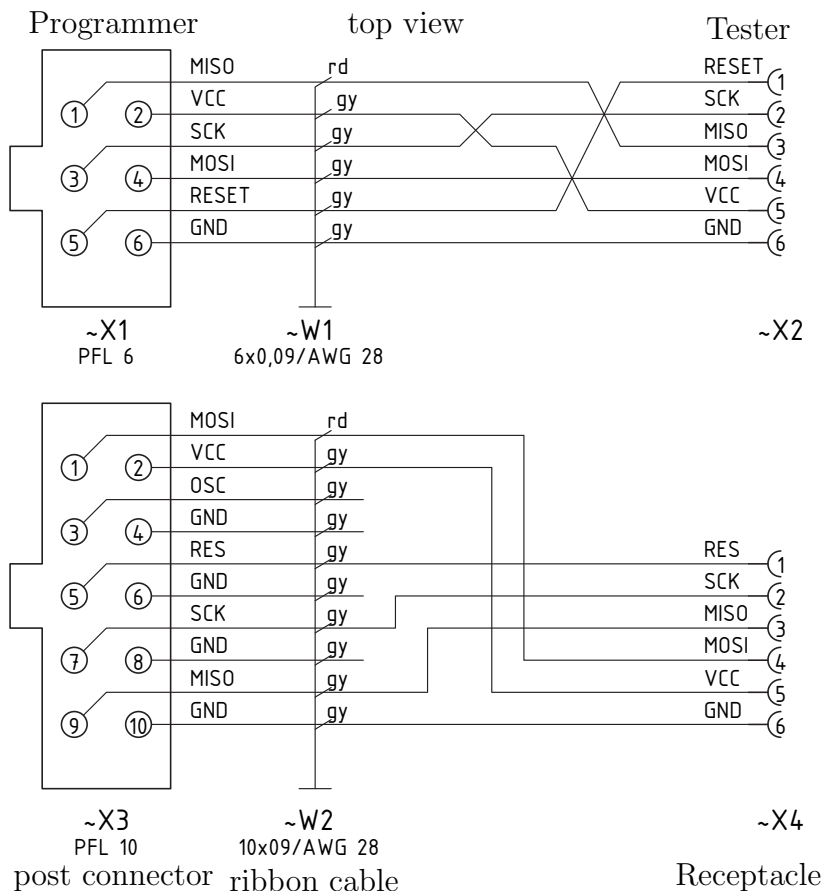
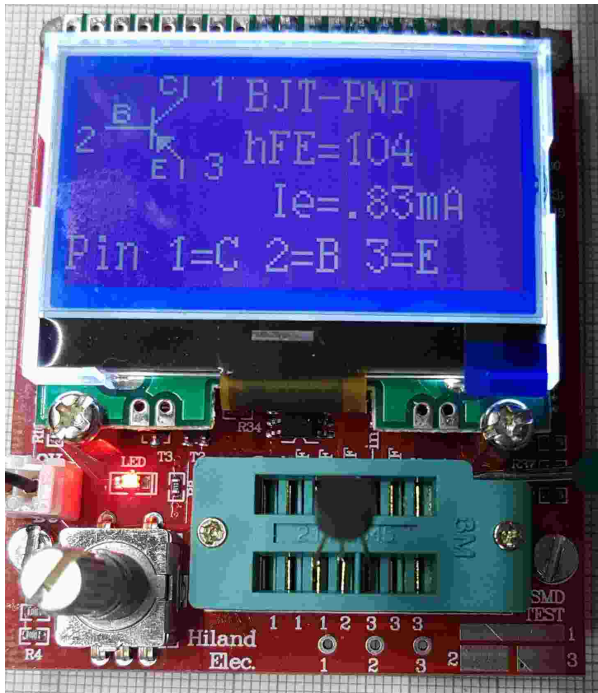
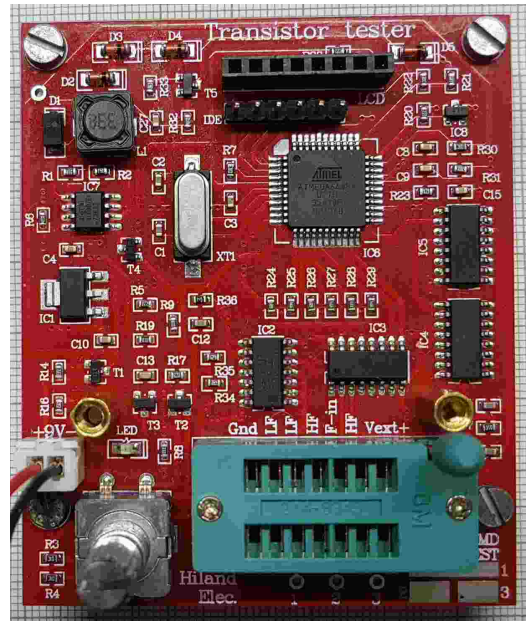


Figure 2.28. 6 and 10 pin cables for programming



(a) measurement ports TP1 to TP3



(b) Pins for special menu function

Figure 2.29. Hiland Tester with test base and 128x64 pixel display

The test ports **TP1**, **TP2** and **TP3** are used for automatic component recognition and are on the board with the the number sequence **1,1,1; 2; 3,3,3**.

The same numbers can be found on the SMD test field and at three holes to solder your own test cable.

The **Testport TP2** is also used for the output of the special function “f-Generator”.

The pins labeled **LF** are intended for the measurement of quartz crystals with a low resonance frequency,

und die mit **HF** are intended for crystals with a high resonance frequency.

The pin **F-in** is used together with **Gnd** for the function frequency measurement.

And the pin **Vext+** is used with **Gnd** for the voltage measurement **and** is also used for the Zener diode measurement.

Chapter 3

Instructions for use

3.1 The measurement operation

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 $0.3V$ and at most $1.3V$, 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. Otherwise 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 (`POWER_OFF` option), the tester shut off automatical after displaying the result for 28 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. If you have not installed the electronic for automatic shut down, your last measurement result will be displayed until you start the next measurement.

Endless measurement mode A special case is the configuration without automatical shut off. For this case the `POWER_OFF` option is not set in the Makefile. 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. For this condition a number (e.g. 5) is assigned to the `POWER_OFF` option. 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 is only shown for 5 seconds. But if you press and hold the start button until the first message is shown, the further measurement results are shown for 28 seconds. The next measurement can be started earlier by pressing the start button during the displaying of result.

3.2 Optional menu functions for the ATmega328

If the menu function is selected, the tester starts a selection menu after a long key press ($> 500\text{ms}$) for additional functions. This function is also available for other processors with at least 32K flash memory. The selectable functions are shown in row two of a 2-line display or as marked function in row 3 of a 4-line display. The previous and next function is also shown in row 2 and 4 of the display in this case. After a longer wait time without any interaction the program leaves the menu and returns to the normal transistor tester function. With a short key press the next selection can be shown. A longer key press starts the shown or marked function. After showing the last function "switch off", the first function will be shown next.




If your tester has also the rotary pulse encoder installed, you can call the menu with the additional functions also with a fast rotation of the encoder during the result of a previous test is shown. The menu functions can be selected with slow rotation of the encoder in every direction. Starting of the selected menu function can only be done with a key press. Within a selected function parameters can be selected with slow rotation of the encoder. A fast rotation of the encoder will return to the selection menu.


frequency The additional function "frequency" (frequency measurement) uses the ATmega Pin PD4, which is also connected to the LCD. First the frequency is always measured by counting. If the measured frequency is below 25kHz , additionally the mean period of the input signal is measured and with this value the frequency is computed with a resolution of up to 0.001Hz . By selecting the POWER_OFF option in the Makefile, the period of frequency measurement is limited to 8 minutes. The frequency measurement will be finished with a key press and the selectable functions are shown again.

f-Generator With the additional function "f-Generator" (frequency generator) you can select output frequencies between 1Hz and 2MHz. You can change the selected frequency only at the highest shown digit location. For the locations 1Hz up to 10kHz digits between 0 and 9 can be selected. For the highest 100kHz location you can select numerals between 0 and 20. In row 1 of the frequency line is indicated with the symbol $>$ or $<$, if a longer key press ($> 0.8\text{s}$) select a higher or lower digit location. You can only select a lower location ($<$), if the current location has selected the 0 digit and if the current location is not the lowest with 1Hz steps. If you have selected the 100kHz location, the $>$ symbol is replaced by a R character. A longer key press ($> 0.8\text{s}$) will result a reset of the frequency value to the initial value of 1Hz in this case. If the POWER_OFF option is selected in the Makefile, the key must be pressed longer, because a short key press ($< 0.2\text{s}$) only resets the time limit of 4 minutes. The elapsed time is shown with a point for every 30 seconds in row 1 of the display. With periodical short key press you can prevent the time out of the frequency generation. With a long key press ($> 2\text{s}$) you will stop the frequency generator and return to the function menu.

10-bit PWM The additional function “10-bit PWM” (Pulse Width Modulation) generates a fixed frequency with selectable pulse width at the pin TP2. With a short key press ($< 0.5s$) the pulse width is increased by 1%, with a longer key press the pulse width is increased by 10%. If 99% is overstepped, 100% is subtracted from the result. If the POWER_OFF option is selected in the Makefile, the frequency generation is finished after 8 minutes without any key press. The frequency generation can also be finished with a very long key press ($> 1.3s$).

C+ESR@TP1:3 The additional function “C+ESR@TP1:3” selects a stand-alone capacity measurement with ESR (Equivalent Series Resistance) measurement at the test pins TP1 and TP3. Capacities from $2\mu F$ up to $50mF$ can be measured. Because the measurement voltage is only about $300mV$, in most cases the capacitor can be measured “in circuit” without previous disassembling. If the POWER_OFF option is selected in the Makefile, the count of measurements is limited to 250, but can be started immediately again. The series of measurements can be finished with a long key press.

Resistor meter With the 1  3 symbol the tester changes to a resistor meter at TP1 and TP3. This operation mode will be marked with a [R] at the right side of the first display line. Because the ESR measurement is not used in this operation mode, the resolution of the measurement result for resistors below 10Ω is only 0.1Ω . If the resistor measurement function is configured with the additional inductance measurement, a 1   3 symbol is shown at this menu. Then the resistor meter function includes the measurement of inductance for resistors below 2100Ω . At the right side of the first display line a [RL] is shown. For resistors below 10Ω the ESR measurement is used, if no inductance is found out. For this reason the resolution for resistors below 10Ω is increased to 0.01Ω . With this operation mode the measurement is repeated without any key press. With a key press the tester finishes this operation mode and returns to the menu. The same resistor meter function is started automatically, if a single resistor is connected between TP1 and TP3 and the start key was pressed in the normal tester function. In this case the tester returns from the special mode operation to the normal tester function with a key press.

Capacitor meter With the 1  3 symbol the tester changes to a capacitor meter function at TP1 and TP3. This operation mode will be marked with a [C] at the right side of the first display line. With this operation mode capacitors from $1pF$ up to $100mF$ can be measured. In this operation mode the measurement is repeated without key press. With a key press the tester finishes this operation mode and returns to the menu. In the same way as with resistors, the tester changes automatically to the capacitor meter function, if a capacitor between TP1 and TP3 is measured with the normal tester function. After an automatic start of the capacitor meter function the tester returns with a key press to the normal tester function.

rotary encoder With the function “rotary encoder” a rotary encoder can be checked. The three pins of the rotary encoder must be connected in any order to the three probes of the transistor tester before the start of the function. After starting the function the rotary knob must be turned not too fast. If the test is finished successfully, the connection of the encoder switches is shown symbolic in display row 2. The tester finds out the common contact of the two switches and shows, if the indexed position has both contacts in open state ('o') or in closed state ('C'). A rotary encoder with open switches at the indexed positions is shown in row 2 for two seconds as “1-/2-/3 o”. This type of encoder has the same count of indexed positions as count of pulses for every turn. Of course the pin number of the right common contact is shown in the middle instead of '2'. If also the closed switches state is detected at the indexed positions, the row 2 of the display is also shown as “1—2—3 C” for two seconds. I don't know any rotary encoder, which have the switches always closed at any indexed position. The interim state of

the switches between the indexed positions is also shown in row 2 for a short time ($< 0.5s$) without the characters 'o' or 'C'. If you will use the rotary encoder for handling the tester, you should set the Makefile option `WITH_ROTARY_SWITCH=2` for encoders with only the open state ('o') and set the option `WITH_ROTARY_SWITCH=1` for encoders with the open ('o') and closed ('C') state at the indexed positions.

C(μF)-correction With this menu function you can change a correction factor for bigger capacity values. You can preset the same factor with the Makefile option `C_H_KORR`. Values above zero reduce the output value of the capacity with this percent value, values below zero will increase the shown capacity value. A short key press will reduce the correction value about 0.1%, a longer key press will increase the correction value. A very long key press will save the correction value. It is a characteristic of the used measurement method, that capacitors with low quality like electrolytic type will result to a too high capacity value. You can detect a capacitor with low quality by a higher value of the `Vloss` parameter. High quality capacitors have no `Vloss` or only 0.1%. For adjusting this parameter you should only use capacitors with high quality and a capacity value above $50\mu F$. By the way the exactly capacity value of electrolytic capacitors is unimportant, because the capacity value differ with temperature and DC-voltage.

Selftest With the menu function "Selftest" a full selftest with calibration is done. With that call all the test functions T1 to T7 (if not inhibited with the `NO_TEST_T1.T7` option) and also the calibration with external capacitor is done every time.

Voltage The additional function "Voltage" (Voltage measurement) is only possible, if the serial output is deselected or the ATmega has at least 32 pins (PLCC) and one of the additional pins ADC6 or ADC7 is used for the measurement. Because a 10:1 voltage divider is connected to PC3 (or ADC6/7), the maximum external voltage can be 50V. A installed DC-DC converter for zener diode measurement can be switched on by pressing the key. Thus connected zener diodes can be measured also. By selecting the `POWER_OFF` option in the Makefile and without key pressing, the period of voltage measurement is limited to 4 minutes. The measurement can also be finished with a extra long key press (> 4 seconds).

Contrast This function is available for display controllers, which can adjust the contrast level with software. The value can be decreased by a very short key press or left turn with the rotary encoder. A longer key press ($> 0.4s$) or a right turn of the rotary encoder will increase the value. The function will be finished and the selected value will be saved nonvolatile in the EEprom memory by a very long key press ($> 1.3s$).

BackColor For color displays this menu item can be enabled with the Makefile option `LCD_CHANGE.COLOR` for selecting the background color. The rotary encoder extension must be installed. You can select the colors red, green and blue with a longer key press. The intensity value of the selected color, which is marked with a $>$ symbol in column 1, can be changed by a turn of the rotary encoder.

FrontColor For color displays this menu item can be enabled with the Makefile option `LCD_CHANGE.COLOR` for selecting the foreground color. The rotary encoder extension must be installed. You can select the colors red, green and blue with a longer key press. The intensity value of the selected color, which is marked with a $>$ symbol in column 1, can be changed by a turn of the rotary encoder.

Show data The function "Show Data" shows besides the version number of the software the data of the calibration. These are the zero resistance (`R0`) of the pin combination 1:3, 2:3 and 1:2. In addition the resistance of the port outputs to the 5V side (`RiHi`) and to the 0V side (`RiLo`) are shown. The zero capacity values (`C0`) are also shown with all pin combinations (1:3, 2:3, 1:2

and 3:1, 3:2 2:1). Next the correction values for the comparator (REF_C) and for the reference voltage (REF_R) are also shown. With graphical displays the used icons for parts and the font set is also shown. Every page is shown for 15 seconds, but you can select the next page by a key press or a right turn of the rotary encoder. With a left turn of the rotary encoder you can repeat the output of the last page or return to the previous page.

Switch off With the additional function “Switch off” the tester can be switched off immediately.

Transistor Of course you can also select the function “Transistor” (Transistor tester) to return to a normal Transistor tester measurement.

With the selected POWER_OFF option in the Makefile, all additional functions are limited in time without interaction to prevent a discharged battery.

3.3 Selftest and Calibration

If the software is configured with the selftest function, the selftest can be prepared by connecting all three test ports together and pushing of the start button. To begin the self test, the start button must be pressed again within 2 seconds, or else the tester will continue with a normal measurement. If the self test is started, all of the documented tests in the Selftest chapter 5.5 will be done. If the tester is configured with the menu function (option WITH_MENU), the full selftest with the tests T1 to T7 are only done with the “Selftest” function, which is selectable as menu function. In addition the calibration with the external capacitor is done with every call from function menu, otherwise this part of calibration is only done first time. Thus the calibration with the automatically started selftest (shorted probes) can be done faster. The repetition of the tests T1 to T7 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. The test 4 will finish only automatically if you separate the test ports (release connection).

If the function AUTO_CAL is selected in the Makefile, the zero offset for the capacity measurement will be calibrated with the selftest. It is important for the calibration task, that the connection between the three test ports is released during test number 4. You should not touch to any of the test ports or connected cables when calibration (after test 6) is done. But the equipment should be the same, which is used for further measurements. Otherwise the zero offset for capacity measurement is not detected correctly. The resistance values of port outputs are determined at the beginning of every measurement with this option.

If you have selected the samplingADC function in the Makefile with the option “WITH_SamplingADC = 1”, two special steps are included to the calibration procedure. After the normal measuring of the zero capacity values, also the zero capacity values for the samplingADC function is measured (C0samp). As last step of the calibration the connection of a test capacitor at pin 1 and pin 3 is requested for later measurement of little coils with the message $1 \text{ --- } 3 \text{ } 10\text{-}30\text{nF}[\text{L}]$. The capacity value should be between 10nF and 30nF , to get a measurable resonant frequency by later parallel connection to a coil with less than 2mH . For inspection of coils with more than 2mH inductance the normal measurement method should result to a sufficient accuracy. The parallel connection of the capacitor to use the other measurement method should not be useful.

A capacitor with any capacity between 100nF and $20\mu\text{F}$ connected to pin 1 and pin 3 is required after the measurement of the zero capacity values. To indicate that, the message $1 \text{ --- } 3 >100\text{nF}$ is shown in row 1 of the display. You should connect the capacitor not before the message C0= or this text is shown. With this capacitor the offset voltage of the analog comparator will be compensated

for better measurement of capacity values. Additionally the gain for ADC measurements using the internal reference voltage will be adjusted too with the same capacitor for better resistor measurement results with the AUTOSCALE_ADC option. If the menu option is selected for the tester and the selftest is not started as menu function, the calibration with the external capacitor is only done for the first time calibration. The calibration with the external capacitor can be repeated with a selftest call as menu selection.

The zero offset for the ESR measurement will be preset with the option ESR_ZERO in the Makefile. With every self test the ESR zero values for all three pin combinations are determined. The solution for the ESR measurement is also used to get the values of resistors below 10Ω with a resolution of 0.01Ω .

3.4 special using hints

Normally the Tester shows the battery voltage with every start. If the voltage fall below a limit, a warning is shown behind the battery voltage. If you use a rechargeable 9V battery, you should replace the battery as soon as possible or you should recharge. If you use a tester with attached 2.5V precision reference, the measured supply voltage will be shown in display row two for 1 second with "VCC=x.xxV".

It can not repeat often enough, that capacitors should be discharged before measuring. Otherwise the Tester can be damaged before the start button is pressed. If you try to measure components in assembled condition, the equipment should be allways disconnected from power source. Furthermore you should be sure, that no residual voltage reside in the equipment. Every electronical equipment has capacitors inside!

If you try to measure little resistor values, you should keep the resistance of plug connectors and cables in mind. The quality and condition of plug connectors are important, also the resistance of cables used for measurement. The same is in force for the ESR measurement of capacitors. With poor connection cable a ESR value of 0.02Ω can grow to 0.61Ω . If possible, you should connect the cables with the test clips steady to the tester (soldered). Then you must not recalibrate the tester for measuring of capacitors with low capacity values, if you measure with or without the plugged test cables. For the calibration of the zero resistance there is normaly a difference, if you connect the three pins together directly at a socket or if you connect together the test clips at the end of cables. Only in the last case the resistance of cables and clips is included in the calibration. If you are in doubt, you should calibrate your tester with jumpers directly at the socket and then measure the resistance of shorten clips with the tester.

You should not expect very good accuracy of measurement results, especially the ESR measurement and the results of inductance measurement are not very exact. You can find the results of my test series in chapter 5 at page 68.

3.5 Compoments with problems

You should keep in mind by interpreting the measurement results, that the circuit of the TransistorTester is designed for small signal semiconductors. In normal measurement condition the measurement current can only reach about $6mA$. Power semiconductors often make trouble by reason of residual current with the identification an the measurement of junction capacity value. The Tester often can not deliver enough ignition current or holding current for power Thyristors or Triacs. So a Thyristor can be detected as NPN transistor or diode. Also it is possible, that a Thyristor or Triac is detected as unknown.

Another problem is the identification of semiconductors with integrated resistors. So the base - emitter diode of a BU508D transistor can not be detected by reason of the parallel connected internal

42Ω resistor. Therefore the transistor function can not be tested also. Problem with detection is also given with power Darlington transistors. We can find often internal base - emitter resistors, which make it difficult to identify the component with the undersized measurement current.

3.6 Measurement of PNP and NPN transistors

For normal measurement the three pins of the transistor will be connectet in any order to the measurement inputs of the TransistorTester. After pushing the start button, the Tester shows in row 1 the type (NPN or PNP), a possible integrated protecting diode of the Collector - Emitter path and the sequence of pins. The diode symbol is shown with correct polarity. Row 2 shows the current amplification factor B or hFE and the current, by which the amplification factor is measured. If the common emitter circuit is used for the hFE determinatation, the collector current I_C is output. If the common collector circuit is used for measuring of the amplification factor, the emitter current I_E is shown. Further parameters are shown for displays with two lines in sequence, one after the the other in line 2. For displays with more lines further parameters are shown directly until the last line is allready used. When the last line is allready used before, the next parameter is shown also in the last line after a time delay automatically or earlier after a key press. If more parameters are present than allready shown, a + character is shown at the end of the last line. The next shown parameter is anyway the Base - Emitter threshold voltage. If any collector cutoff current is measurable, the collector current without base current I_{CE0} and the collector current with base connected to the emitter I_{CES} is also shown. If a protecting diode is mounted, the flux voltage U_f is also shown as last parameter.

With the common Emitter circuit the tester has only two alternative to select the base current:

1. The 680Ω resistor results to a base current of about 6.1mA. This is too high for low level transistors with high amplification factor, because the base is saturated. Because the collector current is also measured with a 680Ω resistor, the collector current can not reach the with the amplification factor higher value. The software version of Markus F. has measured the Base - Emitter threshold voltage in this circuit ($U_f=...$).
2. The 470kΩ resistor results to a base current of only 9.2μA . This is very low for a power transistor with low current amplification factor. The software version of Markus F. has identified the current amplification factor with this circuit ($hFE=...$).

The software of the Tester figure out the current amplification factor additionally with the common Collector circuit. The higher value of both measurement methodes is reported. The common collector circuit has the advantage, that the base current is reduced by negative current feedback corresponding to the amplification factor. In most cases a better measurement current can be reached with this methode for power transistors with the 680Ω resistor and for Darlington Transistors with 470kΩ resistor. The reported Base - Emitter threshold voltage U_f is now measured with the same current used for determination of the current amplification factor. However, if you want to know the Base - Emitter threshold voltage with a measurement current of about 6mA, you have to disconnect the Collector and to start a new measurement. With this connection, the Base - Emitter threshold voltage at 6mA is reported. The capacity value in reverse direction of the diode is also reported. Of course you can also analyse the base - collector diode.

With Germanium transistors often a Collector cutoff current I_{CE0} with currentless base or a Collector residual current I_{CES} with base hold to the emitter level is measured. Only for ATmega328 processors the Collector cutoff current is shown in this case at the row 2 of the LCD for 5 seconds or until the

next keypress before showing the current amplification factor. With cooling the cutoff current can be reduced significant for Germanium transistors.

3.7 Measurement of JFET and D-MOS transistors

Because the structure of JFET type is symmetrical, the Source and Drain of this transistores can not be differed. Normally one of the parameter of this transistor is the current of the transistor with the Gate at the same level as Source. This current is often higher than the current, which can be reached with the measurement circuit of the TransistorTester with the 680Ω resistor. For this reason the 680Ω resistor is connected to the Source. Thus the Gate get with the growing of current a negative bias voltage. The Tester reports the Source current of this circuit and additionally the bias voltage of the Gate. So various models can be differed. The D-MOS transistors (depletion type) are measured with the same methode.

3.8 Measurement of E-MOS transistors and IGBTs

You should know for enhancement MOS transistors (P-E-MOS or N-E-MOS), that the measurement of the gate threshold voltage (V_{th}) is more difficult with little gate capacity values. You can get a better voltage value, if you connect a capacitor with a value of some nF parallel to the gate /source. The gate threshold voltage will be find out with a drain current of about $3.5mA$ for a P-E-MOS and about $4mA$ for a N-E-MOS. The R_{DS} or better R_{DSon} of E-MOS transistors is measured with a gate - source voltage of nearly $5V$, which is probably not the lowest value. In addition, the R_{DS} resistance is determined at a low drain current, which limits the resolution of the resistance value. Often in the case of IGBTs and sometimes also withn enhancement MOS transistors, the available $5V$ of the tester is not sufficient to drive the transistor across the gate. In this case, a battery with about $3V$ will help to make a detection and measurements with the tester possible. The battery is connected to the gate of the transistor with one pole and the other pole of the battery is then connected to a test port (TP) of the tester instead of the transistor gate. When the battery is correctly polarized, the battery voltage is added to the control voltage of the tester and the detection of the transistor succeeds. The battery voltage must then be added to the indicated gate threshold voltage of course, in order to get the correct threshold voltage for this component

3.9 Measurement of capacitors

The capacity values are always computed from the time constant, which is build by the serial connection of the build in resistors with the capacitor during charging. With little capacity values the $470k\Omega$ resistors are used for the measurement the time to reach a threshold voltage. For bigger capacity values with some $10\mu F$ the voltage grow at the capacitor is monitored after charge pulses with the 680Ω resistors. With this voltage grow the capacity can be computed together with the count of fixed length pulses. Very low capacity values can be measured with the samplingADC method. For analysing the same load pulse is repeated many times and the voltage is monitored with the time shift of the ADC S&H time using interval-tics build from the processor clock. But a complete AD conversion take 1664 processor tics! Up to 250 ADC samples are build by this way and from the voltage curve the capacity value is computed. If the samplingADC function is selected in the Makefile, all capacitors with less than $100pF$ are measured with the samplingADC function in the capacitor-meter mode [C]. The resolution is up to $0.01pF$ with a clock frequency of $16MHz$. The calibrated condition is difficult to build with this high resolution. You can assume the use of the samplingADC methode every time, fractions of $1pF$ are displayed at the screen. By the way it should

mentioned, that the junction capacitance of single diodes can also be measured with this method. Because this method can measure the capacity value by charging or discharging, two capacity results are shown. Both values differ because of the capacity diode effect.

3.10 Measurement of coils

The normal measurement of the inductance is based on the measurement of the time constant of the current growth. The detection limit is about $0.01mH$, if the resistance of the coil is below 24Ω . For bigger resistance values the resolution is only $0.1mH$. If the resistance is above $2.1k\Omega$, this technique can never be used to detect coils. The measurement results of this normal measurement is shown in the second line (resistance and inductance). With the samplingADC method a resonant frequency of coils can be detected with greater inductance values. If this effect is noticeable, the frequency and the quality factor Q of the coil is shown additionally in line 3.

The method of resonant frequency measurement can also be used for the determination of the inductance value, if a sufficient big capacitor with known capacity value is connected parallel to a little inductance ($<2mH$). With a parallel connected capacitor the normal measurement of inductance can no more operate well. If the resonant frequency let assume a parallel connected capacitor, the inductance of the normal measurement is not shown and the resistance value is shown in line 1. For this resonant circuit the quality factor Q is also computed and shown behind the frequency value in line 3. You can identify this type of measurement with the inductance value at the first position of line 2, followed by the text “ if ” and the value of the assumed parallel capacity. The value of this parallel capacitor can currently only be set with the calibration function (1 —||— 3 10-30nF(L)). For displays with only two lines, the content for the third line is shown time-delayed in line 2.

Chapter 4

Programming of the TransistorTester

4.1 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 6.3) was taken by the Makefile, if you call “make upload”. The program `avrdude` [14] is available for Linux and Windows operating system. The gnu C-compiler gcc is also taken by the AVR studio software and by the WinAVR [18],[19] 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 ATmega168. The software compiled for a ATmega328 does not run on the ATmega168! A exeption from this rule is the software compiled for ATmega168, this data can also be used for a ATmega328 without changes. Be careful, if you don’t use my Makefile.

With the correct options set, my software runs on the unchanged hardware of Markus F. You must set the option `PARTNO=M8`, **NOT** the option `NO_AREF_CAP` and **NOT** the `PULLUP_DISABLE` option. The clock rate can also be set to $8MHz$ with fuses, no crystal is required!

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

PARTNO describes the target processor:

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

UI_LANGUAGE specifies the favored Language

- `LANG_BRASIL`, `LANG_CZECH`, `LANG_DANISH`, `LANG_DUTCH`, `LANG_ENGLISH`,
- `LANG_GERMAN`, `LANG_HUNGARIAN`, `LANG_ITALIAN`, `LANG_LITHUANIAN`,
- `LANG_POLISH`, `LANG_RUSSIAN`, `LANG_SLOVAK`, `LANG_SLOVENE`,
- `LANG_SPANISH` and `LANG_UKRAINIAN` is currently avaiable. The russian or ukrainian

language requires a LCD with cyrillic character set.

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. You should set this option, if your display shows wrong characters instead of μ or Ω .

Example: `CFLAGS += -DLCD_CYRILLIC`

LCD_DOGM must be set, if a LCD with ST7036 controller (Type DOG-M) is used for displaying. The LCD-contrast is then set with software commands. If you have changed the contrast value to a wrong value, so that you can not read anything at your display, you should first try to read something from a side look to the display. If this fails also, you should reset the EEPROM to the initial values with a ISP programmer.

Example: `CFLAGS += -DLCD_DOGM`

FOUR_LINE_LCD can be used with a 4x20 character display for better using the additional space. Additional parameters, which are shown only short in row 2, will be shown in row 3 and 4 with this option. With graphic displays, the number of lines is determined from the display height and the line spacing.

Example: `CFLAGS += -DFOUR_LINE_LCD`

LCD_LINE_LENGTH specifies the number of characters that can be displayed in one line. With graphic displays, the number of characters is calculated from the character width and the display width.

Example: `CFLAGS += -DLCD_LINE_LENGTH=20`

DD_RAM_OFFSET Some character displays use different DD-RAM starting addresses for the beginning of each line. Usually the DD-RAM starting address for line 1 is 0. Some displays like TC1604 or TC1602 use a 128 (0x80) for the beginning of line 1. This can be respected with this option. Example: `CFLAGS += -DDD_RAM_OFFSET = 128`

WITH_LCD_ST7565 This option must be used, if a 128x64 pixel LCD is connected with serial interface. For this display type further options must be set, which are described in table 4.1. You can also use the similar SSD1306 controller instead of the ST7565 controller for example. This must be done by setting the variable `WITH_LCD_ST7565` to 1306. A PCF8812 or PCF8814 Controller is also supported, if the Option is set correctly. Also a display with a ST7920 or NT7108 controller can be connected. For the NT7108 controller a additional serial-parallel converter 74HC(T)164 or 74HC(T)595 must be used.

Example: `WITH_LCD_ST7565 = 1`

LCD_INTERFACE_MODE For the SSD1306 controller also the I²C type interface with address 0x3c can be used instead of the 4-wire SPI interface by setting this option to 2. For the ST7920 controller a special serial interface can be selected by setting this option to 5. If only one connection type is provided for a controller, you need not set the constant `LCD_INTERFACE_MODE`. All currently used values for `LCD_INTERFACE_MODE` and `WITH_LCD_ST7565` are shown in table 4.1.

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

Table 4.1. Number setting for controller and interface mode

The values in brackets are used software internal and are shown for information only. You should not set the values in brackets here in the Makefile.

Example: `CFLAGS += -DLCD_INTERFACE_MODE=2`

LCD_SPI_OPEN_COL With the option `LCD_SPI_OPEN_COL` the data signals of the SPI interface are not switched to VCC directly. The signals are switched to GND only, for high signals the pullup resistors of the ATmega are used. For the RESET signal a external pull-up resistor is required, if the option `PULLUP_DISABLE` is set. For the other signals the internal pullup resistors of the ATmega are temporary used, even if option `PULLUP_DISABLE` is set.
Example: `CFLAG += -DLCD_SPI_OPEN_COL`

LCD_I2C_ADDR The I²C address of the SSD1306 controller can be selected to 0x3d by presetting the constant `LCD_I2C_ADDR` to 0x3d.

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

LCD_ST7565_RESISTOR_RATIO With this option the resistor ratio for the voltage regulator of the ST7565 controller is set. Usually values between 4 and 7 are practical. The value can be set between 0 and 7.

Example: `LCD_ST7564_RESISTOR_RATIO = 4`

LCD_ST7565_H_FLIP With this option the display content can be flipped in horizontal direction.
Example: `CFLAGS += -DLCD_ST7565_H_FLIP = 1`

LCD_ST7565_H_OFFSET This option can be used to adapt the display window to the used memory area. The controller uses more horizontal pixel (132) as the display window shows (128). Depending of your display module a value of 0, 2 or 4 can be required for proper presentation.

Example: `CFLAGS += -DLCD_ST7565_H_OFFSET = 4`

LCD_ST7565_V_FLIP With this option the display content can be flipped in vertical direction.

Example: CFLAGS += -DLCD_ST7565_V_FLIP = 1

VOLUME_VALUE You can predefine a contrast value for ST7565 or SSD1306 controllers. The value for the ST7565 controller can be between 0 and 63. For the SSD1306 controller you can select a value between 0 and 255.

Example: CFLAGS += -DVOLUME_VALUE = 25

LCD_ST7565_Y_START With this option you can set the first row correctly to the top of screen. The first row is shifted to the middle of the screen for some display variants. For this variants you can shift the first row to the top of the screen again, if this option is set to 32 (half of the screen height).

Example: CFLAGS += -DLCD_ST7565_Y_START = 32

LCD_CHANGE_COLOR This option expand the menu functions with a selection item to change the background and the foreground color. If the value is set to 2, the colors blue and red are swapped. You can select this option only for color displays (controller ST7735 or ILI9163).

Example: CFLAGS += -DLCD_CHANGE_COLOR=1

LCD_BG_COLOR With this 16-bit value you can select a background color. Normally the upper 5 bits are used for the color red, the middle 6 bits are used for the color green and the lower 5 bits are used for the color blue. Sometimes the bits for the colors red and blue are swapped. You can select this option only for color displays (controller ST7735 or ILI9163).

Example: CFLAGS += -DLCD_BG_COLOR=0x000f

LCD_FG_COLOR With this 16-bit value you can select a foreground color. The example selects the color white for text and symbols. You can select this option only for color displays (controller ST7735 or ILI9163).

Example: CFLAGS += -DLCD_FG_COLOR=0xffff

FONT_8X16 You must select one font size for the ST7565 controller. Selectable are different fonts with the name "FONT_" with appended size information (width X height). Currently the font sizes 6X8, 8X8, 7X12, 8X12, 8x12thin, 8X14, 8X15, 8X16, and 8X16thin are available. Font size 8X16 or 8x16thin is the most efficient use of graphics space for a 128x64 pixel LCD.

Example: FONT_8X16

BIG_TP The pin numbers for the graphical presentation can be shown bigger with this option.

Example: CFLAGS += BIG_TP

INVERSE_TP With this option you can select a inverse presentation (white background) of the pin numbers on the graphical display. Because a boarder is required for this presentation, you can not combine this option with the BIG_TP option.

Example: CFLAGS += INVERSE_TP

STRIP_GRID_BOARD This option adapts the software to a changed port D connection for strip grid printed boards. You can find the details in the chapter hardware 2.1 at page 9. You can also choose alternative assignments of ATmega pins for graphical displays. For the chinese "T5" board you must set the STRIP_GRID_BOARD option to 5. For alternative pin assignments of graphical displays the assignment of the pushbutton signal is unchanged.

Example: CFLAGS += -DSTRIP_GRID_BOARD

WITH_MENU activated a menu function for a ATmega328. You can select some additional functions with a selection menu, which you can call with a long key press (> 0.5s).

Example: CFLAGS += -DWITH_MENU

MAX_MENU_LINES This option specifies a maximum count of lines for the shown choices of menu items. Normally the count of lines for the menu items is given by the present count of

lines of the display. Because there are usually more items selectable as the display can support, the choices are replaced in a cyclic manner. Building the display content in this cyclic way will take several time, especially for big color displays with many lines. With the limitation of the line count by this option you can reduce the output time for the menu choices significant, which will speed up the operation. The default value for this item is 5.

Beispiel: CFLAGS += -DMAX_MENU_LINES=3

WITH_ROTARY_SWITCH The menu function can be easier controlled with a the extension of a rotary pulse encoder. See the description 2.6 in the Hardware section for details of the required extension. If your rotary pulse encoder has the same count of indexed positions (detent) as pulses of the switch for every turn, you must set the option WITH_ROTARY_SWITCH to 2. If the rotary pulse encoder has twice the count of indexed position, you must set the option WITH_ROTARY_SWITCH to 1. Setting the WITH_ROTARY_SWITCH to 5 selects the highest resolution for the rotary switch. Every cycle of the two switches results to a count of 4. Usually this setting is only usefull for rotary switch encoders without indexed positions. A setting of the WITH_ROTARY_SWITCH to 4 is required for correct handling of two separate push buttons for Up and Down, which are installed instead of the normal rotary encoder switches. Do not use a setting of 4 for normal rotary encoders!

Example: CFLAGS += -DWITH_ROTARY_SWITCH=1

CHANGE_ROTARY_DIRECTION You can change the direction of the detected rotary direction by hardware swap of the two switch signals or by setting the this option.

Example: CFLAGS += -DCHANGE_ROTARY_DIRECTION

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. If the menu function is selected, only the calibration part of the self test is executed by automatic start with shorted probes. The selftest parts T1 to T7 are only executed, if the selftest is started with menu selection.

Example: CFLAGS += -DWITH_SELFTEST

NO_COMMON_COLLECTOR_HFE disables the hFE measurement of transistors with the common collector circuit. You can save memory to enable the extended selftests T1 to T7 for a ATmega168 processor. By default both measurement circuits for the hFE measurement are enabled, but there is no place in the program memory of the ATmega168 for the extended selftests.

Example: CFLAGS += -DNO_COMMON_COLLECTOR_HFE

NO_COMMON_EMITTER_HFE disables the hFE measurement of transistors with the common emitter circuit. You can save memory to enable the extended selftests T1 to T7 for a ATmega168 processor. By default both measurement circuits for the hFE measurement are enabled, but there is no place in the program memory of the ATmega168 for the extended selftests.

Example: CFLAGS += -DNO_COMMON_EMITTER_HFE

NO_TEST_T1_T7 This option disable the execution of the selftest parts T1 to T7. This tests are usefull to find errors in the hardware like incorrect measurement resistors or isolation problems. If your hardware is well, you can omitt this selftest parts T1 to T7 by setting this option to get a faster calibration. With enabled menu function the selftest parts T1 to T7 are only started by selection of the menu function "Selftest". The ATmega168 processor does not use the selftest parts T1 to T7, if both measurement types for hFE determination are used.

Example: CFLAGS += -DNO_TEST_T1_T7

AUTO_CAL The zero offset for capacity measurement will be written additionally to the EEprom with the selftest routine. Additionally the offset voltage of the analog comparator (with option

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 a capacity value between $100nF$ and $20\mu F$ to pin 1 and pin 3 after measurement of capacity zero offset. All found values will be written to EEprom and will be used for further measurements automatically. The port output resistance values will be determined at the beginning of each measurement.

Example: CFLAGS += -DAUTO_CAL

SHORT_UNCAL_MSG After the test of a part a message is shown for processors with at least 32K flash memory, if the tester is still uncalibrated. Normally follows after the hint a short description, how the tester can be calibrated. This description is not shown, if you set the option SHORT_UNCAL_MSG in the Makefile. With this option set, the tester only display a one line hint. This reduces the required space of flash memory and also the display time for the user, which already know, how to calibrate the tester.

Example: CFLAGS += -DSHORT_UNCAL_MSG

NO_ICONS_DEMO This option will switch off the additional demonstration of the icons and the output of the character set with the menu function “Show data”. This reduces the required space of flash memory and also the display time for the user.

Example: CFLAGS += -DNO_ICONS_DEMO

WITH_ROTARY_CHECK This option enables the additional menu function for the test of a rotary encoder. For the test you must connect a rotary encoder to the test pins TP1, TP2 and TP3. Please note, that you can not check the build-in rotary-encoder of the tester! You can also use a rotary encoder for ease in operation of the tester with the option WITH_ROTARY_SWITCH.

Example: CFLAGS += -DWITH_ROTARY_CHECK

NO_FREQ_COUNTER With this option you can deselect the frequency counter function of the tester. This is especially useful if the pin PD4 (ATmega328) can not be used together with the connected display. The corresponding entry in the list of menu functions then no longer appears and also Flash memory space is saved.

Example: CFLAGS += -DNO_FREQ_COUNTER

WITH_FREQUENCY_DIVIDER With this option the menu is expanded by a selectable prescaler for the frequency counter. The scaler can be selected to 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64 and 1:128 . This option is only useful, if a external prescaler can be connected to the frequency input of the tester. The shown frequencies and periodes of the measurements will respect the selected scaling factor.

Example: CFLAGS += -DWITH_FREQUENCY_DIVIDER

NO_FREQUENCY_SWITCH With this switch you can deselect the support of the frequency input switch of the ATmega644/ATmega1284 circuit. The support of the LF and HF crystal measurement will not be supported with this option.

Example: CFLAGS += -DNO_FREQUENCY_SWITCH=1

PWM_SERVO will select a special PWM generation for testing servos instead of a general purpose PWM generation.

Example: CFLAGS += -DPWM_SERVO=1

WITH_SamplingADC With this option set, the tester make use of the sampling method of ADC in special cases. By shifting the sampling time of the ADC with increments of 1, 4 or 16 processor clock intervals for repeatable signals fast changes of voltages can be monitored. The load time of little capacitors below $100pF$ can be monitored with a resulting resolution of $0.01pF$ with a $16MHz$ processor clock. With the same method the resonant frequency of little coils below $2mH$ can be monitored with a parallel capacitor to build a LC-resonator. If the capacity of the parallel capacitor is known, the inductance of the coil can be calculated with

high resolution from the resonant frequency. As a side product the quality factor Q can be estimated from the resonant behavior. This features are switched on by setting the option `WITH_SamplingADC`. At the calibration sequence additionally the zero capacity values of the sampling method is measured and after that the capacity value of a suitable capacitor for later building the LC-resonator with a unknown coil is measured.

Example: `WITH_SamplingADC = 1`

WITH_XTAL This option enables additional tests for crystals and resonators, if the `SamplingADC` function is also enabled and a 16 MHz crystal is used for clock generation (`OP_MHZ = 16`). If possible, the frequencies for serial and parallel circuit is measured and than the serial capacity C_m of the equivalent circuit is tried to compute from the frequency offset.

Example: `CFLAGS += -DWITH_XTAL`

WITH_UJT This option enables additional tests for Unijunction transistors. If the `SamplingADC` function is enabled, the tester tries to build a oscillator with the part. But the UJT type is also detected without the `SamplingADC` function. Without the option `WITH_UJT` the unijunction transistors are detected as double diode.

Example: `CFLAGS += -DWITH_UJT`

WITH_PUT This option enables a additional test for “Programmable Unijunction Transistor”. Without this option PUTs are usually detected as Bipolar Junction Transistor.

Example: `CFLAGS += -DWITH_PUT`

FET_Idss This option enables additional measurements to compute the drain current I_{dss} , if the estimation is not above $60mA$. The estimation and calculation is done with a assumed quadratical current propagation.

Example: `CFLAGS += -DFET_Idss`

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. This option should be set only for special cases to check the delay function.

Example: `CFLAGS += -DFREQUENCY_50HZ`

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 $40\mu F$).

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. 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=14`

REF_L_KORR specifies a additional offset in mV units to the reference voltage for the measurement of inductance values. The `REF_C_KORR` offset and respectively the offset value from the calibration is additionally used with the inductance measurement. The `REF_L_KORR` value will be subtracted for measurements without a 680Ω resistor, for measurements with a 680Ω resistor the value will be added. A correction value of 10 will change the result about 1 percent.

Example: `CFLAGS += -DREF_L_KORR=40`

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.

Example: CFLAGS += -DC_H_KORR=10

WITH_UART uses the pin PC3 as output for the serial text (V24). If the option is not set, the pin PC3 can be used for reading a external voltage with a 10:1 resistor divider. With this equipment you can check the breakdown voltage of zener diodes, which have more than 4.5V breakdown voltage. This measurement will repeat with 3 measurements per second until you release the Start button.

Example: CFLAGS += -DWITH_UART

WITH_HARDWARE_SERIAL is only used to enable serial output for testers build with the Arduino UNO board.

Example: CFLAGS += -DWITH_HARDWARE_SERIAL

TQFP_ADC6 The Option TQFP_ADC6 uses the additional input ADC6 of the ATmega with TQFP or QFN package instead of the PC3 pin (ADC3). With this option the external voltage input can be used independent of the usage of PC3 pin for serial output. The ADC6 input is then used for the zener diode measurement and for the dialog selectable external voltage measurement for a ATmega328.

Example: CFLAGS += -DTQFP_ADC6

TQFP_ADC7 The Option TQFP_ADC7 uses the additional input ADC6 of the ATmega with TQFP or QFN package instead of the PC3 pin (ADC3). With this option the external voltage input can be used independent of the usage of PC3 pin for serial output. If this option is used without the option TQFP_ADC6, both the zener diode measurement and the measurement of external voltage with the dialog is done with the ADC7 analog input. If this option is used together with the TQFP_ADC6 option, is the zener diode measurement done with the ADC6 pin and both pins are used for voltage measurement with the dialog of the ATmega328. Both ADC input pins should be assembled with a 10:1 voltage divider.

Example: CFLAGS += -DTQFP_ADC7

WITH_VEXT enables the measurement of a external voltage with a 10:1 voltage divider. For the ATmega168 or ATmega328 processor usually the PC3 pin is used as input, if no option TQFP_ADC6 or TQFP_ADC7 is set. In this case this option is only possible, if the WITH_UART option is not set.

Example: CFLAGS += -DWITH_VEXT

RMETER_WITH_L select for the resistor measurement function, which is selected with a resistor at TP1 and TP3, additionally the measurement of inductance. The operation mode is indicated with a [RL] at the end of the first display line. With the additional test for inductance the measurement time is increased for resistors below 2100Ω considerably. Also resistors below 10Ω will not be measured with the ESR method Without this option, because a part with inductance can not be excluded. Because the ESR measurement method uses short current pulses, parts with inductance can not be measured. The resistors below 10Ω can only measured with a resolution of 0.1Ω without this option, because only with the ESR method a resolution of 0.01Ω can be obtained. If this option is set, the previous limitations are not affected, but the measurement time can be longer.

Example: CFLAGS += -DRMETER_WITH_L

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. For the ATmega8 the automatic scale switchover is not used any more.

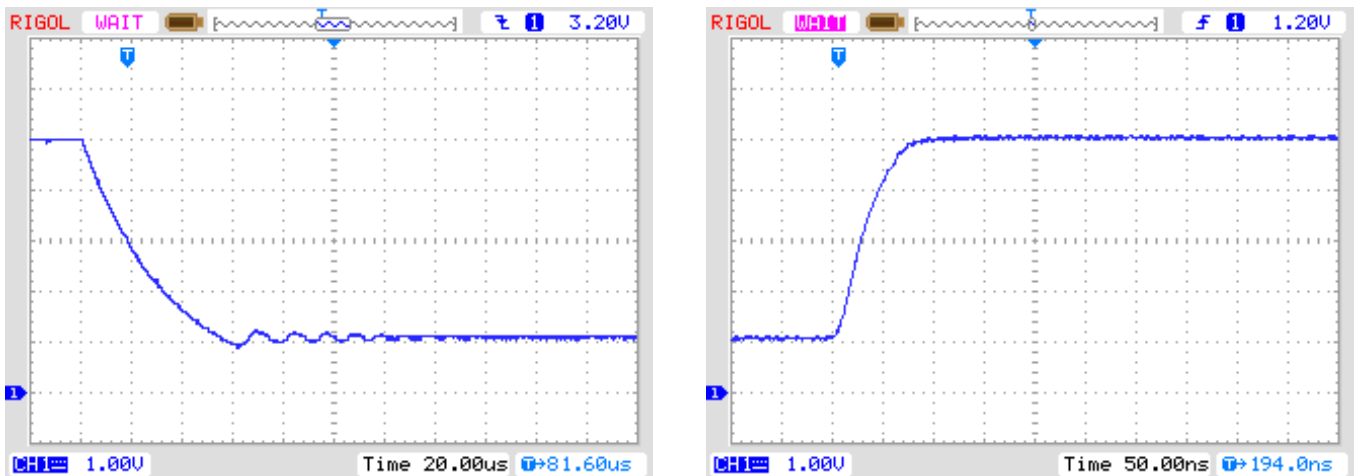
Example: CFLAGS += -DAUTOSCALE_ADC

ESR_ZERO defines a zero offset for ESR measurements. The zero offsets for all three pin combinations will be determined with the selftest and replaces the preset zero offset. This zero offsets will be subtracted from all ESR measurements.

Example: CFLAGS += -DESR_ZERO=29

NO_AREF_CAP tells your Software, that you have no capacitor ($100nF$) 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



(a) from 5V to 1.1V

(b) from 1.1V to 5V

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

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 only a additionally 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 1MHz, 8MHz and additionally 16MHz. The 8MHz operation is recommended for better resolution of capacity and inductance measurement.

Example: OP_MHZ = 8

RESTART_DELAY_TICS must be set to 6, if the ATmega168 or ATmega328 is used with the internal RC-oszillator instead of the crystal oszillator. If this value is not preset, the software respects the 16384 clock tics delay for restart from sleep mode with the crystal operation.

Example: CFLAGS += -DRESTART_DELAY_TICS=6

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

EBC_STYLE specifies, that the output of transistor pin layout is done with format "EBC=..." or "GDS=...". This way of output save program memory for the ATmega. Without this option the layout is shown with the format "123=...", where every point represent a E (Emitter),

B (Base) or C (Collector). For FET transistors every point can be a G (Gate), D (Drain) or S (Source). If the sequence of the test pins is not 1, 2 and 3 in the reading direction, you can invert the sequence with the option `EBC_STYLE=321`. The pin assignment is then shown with style “321=...”, which will better match the usual reading direction, if the testpin sequence is 3,2,1.

Example: `CFLAGS += EBC_STYLE`

NO_NANO specifies that the decimal prefix nano will not be used to display the measurement results. So capacity values will be shown in μF instead of nF .

Example: `CFLAGS += NO_NANO`

NO_LONG_PINLAYOUT can be set to prevent the long style of pin layout for graphical displays like “Pin 1=E 2=B 3=C”. If the option is set, the short style is used instead like “Pin 123=EBC”.

Example: `CFLAGS += NO_LONG_PINLAYOUT`

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 5ms.

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 also shut down the tester, if you have selected the `WITH_MENU` option.

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 28 seconds for the single measurement, for the multiple measurement version display time is reduced to 5 seconds (set in config.h). If the start key is pressed a longer time on power on time, the display time is also 28 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 1mV value. The warning level of battery voltage is 0.8V higher than the specified poor level, if the poor level is more than 5.3V. If the poor level is 5.3V or less, the warning level is 0.4V higher. If the poor level is below 3.25V, the warning level is only 0.2V higher than the selected poor level and if the poor level is below 1.3V, the warning level is only 0.1V 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=5400

Example for 7805 type regulator (6.4V): CFLAGS += -DBAT_POOR=6400

DC_PWR This voltage level in mV units specify the battery voltage above which the tester changes to the “DC_Pwr_Mode”. Normally the tester operates in a battery mode, where all additional functions are limited in time. With the “DC_Pwr_Mode” the tester runs the additional functions with unlimited time. Because there is no DC-DC converter operating with 0.9V input voltage, the “DC_Pwr_Mode” is also entered, if the battery voltage is detected below 0.9V.

Example: CFLAGS += -DDC_PWR=9500

BAT_NUMERATOR defines the numerator of a fraction used for scaling the input voltage to get the right battery voltage. For a normal voltage divider build with a 10kΩ and a 3.3kΩ resistor you get a fraction of (10000 + 3300)/3300. You should reduce the fraction to 133/33 .

Example: CFLAGS += -DBAT_NUMERATOR=133

BAT_DENOMINATOR specifies the denominator of a fraction, which is used to scale the voltage value.

Example: CFLAGS += -DBAT_DENOMINATOR=33

EXT_NUMERATOR defines the numerator of a fraction used for scaling the external input voltage. With a voltage divider build with a 180kΩ and a 20kΩ resistor the fraction is (180000 + 20000)/20000 . You should reduce the fraction to 10/1 .

Example: CFLAGS += -DEXT_NUMERATOR=10

EXT_DENOMINATOR specifies the denominator for the fraction to scale the external voltage.

Example: CFLAGS += -DEXT_DENOMINATOR=1

INHIBIT_SLEEP_MODE disable the use of the sleep mode of the processor. Normaly the software uses for longer work breaks the sleep mode to avoid unneeded current consumption. The usage of this sleep mode indeed spare battery capacity, but produce additional stress for the voltage regulator.

Example: INHIBIT_SLEEP_MODE = 1

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. Usually the right setting for a Diamex ALL-AVR is preset to the `Makefile` . You can get a full list of supported programmers with the command `avrdude -c ?` . Some other selections like USBasp may be predefined with comment character `#` in row 1. You can only select another one by deselecting the preset one with the `#` character in row 1 and enable another programmer with a new entry or by removing the `#` charactor of a other one.

Example: PROGRAMMER=avrisp2

BitClock selects the Bit clock period for the Programmer. See the description of the -B parameter of `avrdude`.

Example: BitClock=5.0

PORT select the port where `avrdude` can reach your microcontroller (atmega). Example: PORT=usb

Additional parameters can be set in the files `transistortester.h` and `config.h`. The file `config.h` contains global settings, defines the port / pin constellation, the clock frequency of the ADC and the resistor values used for measurement. The file `Transistortester.h` contains the global variables and tables and also the text used for LCD output. Normally there is no reason to change these values.

For further information please look to the manual pages of `avrdude` and online documentation [14].

4.2 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 sure, 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 `Software/default` and to the chapter 4.1 at page 45. The result of compilation have the extensions `.hex` and `.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` [14], 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 crystal 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 sure with the fuses, leave them as default set by manufacturer and first bring 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.

The program `avrdude` probably reports a error for setting the extended fuse `efuse`. The reading of unused fuse bits is specified as “1” for the ATmega, but the `avrdude` program mask the unused bits, so that it expect a “0” for all unused bits. Normally the `efuse` should be set to 0xfc, but `avrdude` read back 0x04 with the mask. You can change the file `avrdude.conf` to change the behaviour of `avrdude` or you can set the `efuse` to 0x04. The value for all `efuses` can be set with the identifier `EFUSE_VAL` at the begin of file `setup.mk` in the source directory. Probably the fuses are also set correctly with the error message.

4.2.1 Using of Linux







In order to spare other colleagues my experience of desperation and “sleepless nights”, this subchapter was written. I had acquired a clone tester without any AVR experience and wanted to let it spell the German language. The experience gained in this way should help other “willing” inexperienced people to SUCCESSFULLY program their tester. We take this opportunity to thank the developer of the transistor tester and author of this document, Karl-Heinz Kübbeler see `cite karlheinz1`, for his commitment and patience, because the following pages would never have been created without his help. So that the translation of the firmware and burning into the MCU succeed and at the same


time ... the wheel doesn't have to be reinvented'', a part of the following pages has been taken from the original. So once again ... **MANY THANKS** to Karl-Heinz Kübbeler.


Operating system Linux

The programming below Linux brings many advantages, because this OS was developed by experts who orientate themselves on the wishes of the users. In addition, the environment is available free of charge and perfectly maintained. Another advantage is the security of the OS itself but also when using the Internet. Today's editions are much easier to use than the competitor OS. There are also very powerful editors such as vim or emacs, but they require some training time. Vim in particular has the advantage that it is preinstalled on practically every Linux distribution. But with this editor in particular, the change between input mode and command mode is unusual and takes time to getting used to. These guide intended to encourage all ''not'' Linux users to test it NOW by programming their tester with it. The following instructions are tested with Linux-Mint, which is released with three desktop environments (cinnamon, MATE or Xfce). All following hints should operate with any of the three desktop environments. The Installation can be done with different ways and brings its own boot manager, so that you can continue to use your existing OS in parallel.

Tips for the Linux usage


At first I would like to give a hint for everyone, who do not like to type texts with the keyboard. You can copy this manual to a USB-stick and open the PDF file with a double click of the left mouse key , when the mouse cursor points to the file. Alternatively, you can use the  key to select a different PDF viewer program than the preselected one with the double click. The opened window can be moved on the screen and the size can also be adjusted. There are usually various ways of doing this. One of them is to press the  key when the mouse pointer is on the window header and then select the **move** function. For example, you can move the window with the PDF documentation to the left side of the screen. By pressing the , the window will snap into the new position. To move, you can also grab the header bar of the window with the  key and move it while keeping the  key pressed. With this way the window will snap in the new position, if the mouse button is released.

In the next step the key combination **Ctrl**+**Alt**+**T** is pressed at the same time, to open a (new) command window. This is now moved to the right half of the screen in the manner already described and its size can also be changed. The size change can be done with the LMB key at the edges or corners of the window, or with the  on the header with the **change size** function. The handling is otherwise the same as for moving.












As a rule, the desktop environment of Linux have more than one workspace, which can be switched with the key combination **Ctrl**+**Alt**+**→** or **ctrl**+**Alt**+**←**. With the  key at the header line of a window you can select one of the workspaces on which this window is displayed.



So you can put together the windows you need for your work on a separate working desk and you don't get stifled in all the different windows for all work areas. By the way, with all commands and file identifiers, it is important to know that Linux is case-sensitive.

Installing of Program packages

For installing software packages you need a access to the internet. Before you can program the tester, the program packages binutils-avr, avrdude, avr-libc and gcc-avr must be installed first. Now you can type the command given below with the keyboard. You can also scroll to this point in the open PDF document and select the following text with the left mouse button pressed  :

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

At the end of the text the mouse button  must be released, to complete the selection. If the text is shown in a line as in this example, you can select the full line also with three  mouse clicks anywhere at the line. Then you can move the mouse pointer into the right command window and insert the previously marked text into the command line by pressing the middle mouse button further abbreviated as . For many mice, the scroll wheel is also the middle mouse button. For mice without a middle mouse button, it is possible to replace the middle mouse button by pressing the  mouse buttons at the same time. Another way to copy the selected text is to copy the mouse selection with the + and to insert the text with the  and the function  with mouse pointer in the right window. Regardless of how the command text got into the command line, you should check the text again before sending the command by pressing the  key. As a rule, you do not need to be afraid that something bad will happen when you install packages due to incorrect operation. Before executing the operation, the `apt-get` program checks whether the packages listed are already installed and whether the dependencies are met. But in some cases a previously installed package can be replaced by a newer version. Then the `sudo` program will first ask you for the user password before it will execute the rest of the command line. You should finish and confirm the password entry with  or . Now all specified software packages are downloaded and installed by `apt-get`.

It may be that `apt-get` asks questions when installing the packages, which can usually be answered with . **Don't forget**, Linux is case sensitive. So don't answer with ! Of course there are other ways to install the packages, which use the graphical interface like `synaptic` or `dpkg`. But it is not easier to select a mixed group of packages. If you use one of these package managers, they should know each installed package, any way which way it was installed. The graphical package managers can help you to find out the names of the packages.





Download of sources

You can check the correct installation of the `git` package by typing the command:

```
git version
```

The `git` program should respond with the output of its version number. If a folder `TransistorTester-source` already exists in your home directory, you should rename or delete it. The package “git” is used for downloading the sources and the documentation from the Git archive. With the command:

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

you can download the actual `transistortester` source archive. The directory `TransistorTester-source` is now added to your home directory, which is usually the full file name `/home/` followed by your user ID. So you must see a new directory `TransistorTester-source`, which can be checked by typing `ls` (  ). You get more information about files and directories, if you call the command with options like `ls -lh`. In this example two options for the `ls` command have been combined here, the `-l` option and the `-h` option. This input form is a short form for the command `ls -l -h` or `ls -l --human-readable`. Some options appear in two forms, a short one with one - like `-h` and a long form with two - like `--human-readable`. Usually the sequence of the options doesn't care, but the options must be separated by at least one space . If you write some short form options (one character) together, there is no space allowed and you need only one - character in front of

the collection. You can get help about the command and the options for nearly every command by appending the `-help` option. Of course, this also applies to the `git` command.

For getting updates, you should change your working directory to `TransistorTester-source` and call:

```
git pull
```

You can do that also from every working directory with some command interpreters:

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

You should better not change anything in the `~/TransistorTester-source` directory tree to let the `git pull` run properly.

compiling of transistor tester source

For compiling the source you must only type the simple command `make` in the right working directory. Because the ATmega can be easy programmed with a `make upload` command, we should first prepare the access to the interface of the ISB-programmer. The further operating steps you can find in section 4.2.1 at page 61.

usage of interfaces

All modern ISP-programmers with a serial interface like to use the USB interface, as this interface also ensures the power supply. For these devices you should check, which device is assigned to this device. When inserting a USB device in Linux, an entry to the system log file is made. Since the system log is a text file, you can simply display it on the screen. To do this, you can use the command `dmesg` in the console window, and After you have plugged in the programmer, you should use:

```
dmesg|tail
```

The `dmesg` command output the full logfile and the command `tail` will show only the last 10 lines. So the result for a Pololu programmer will be like:

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

Important are only the lines 7 and 8 with the entries `ttyACM0` and `ttyACM1`. These names are assigned to the two serial interfaces. With Linux, all devices are also part of the directory tree and appear in the folder `/dev/`. With their full names, the two serial interfaces are called `/dev/ttyACM0` and `/dev/ttyACM1`. For the Pololu programmer you have to know that the first serial interface is used for the ISP programmer and the second serial interface can be freely used for other purpose. You can check the existence of these device enties with:

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

The result should look like:

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

You can see with this output, that the access is allowed for the user `root` and for members of the `dialout` group. You can use the `id` command to check your group membership. The group `dialout` should appear in the list, otherwise the change of group membership is described in the next working point. Another example for the system log of a programmer with serial interface is a “Diamex ISP-PRog NG” programmer:

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

For this example the device name is identical (`ttyACM0`).

If everything is OK up to this step, you must only change the `PORT` entry in the Makefile, for this example `"PORT = /dev/ttyACM0"` would be the correct setting. So the `avrdude` program can access the programmer. If additional USB-serial interfaces are used in your system, the last digit of the device name can differ to the shown examples. I would like to mention, that another group of USB-serial interfaces use a name `ttyUSB` instead of `ttyACM`.

If your programmer does not use a standard USB-serial interface, but a special USB interface, you must probably prepare your system to enable user access for this device. Your `PORT` setting in the Makefile should be `"usb"` for this device. If you have connected a ISP programmer with USB interface, you can see the recognized USB devices with the command `lsusb` in any console window. A sample of the result of `lsusb` you can see here:

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

A Device 58 is detected here as a AVR ISP mkII type (DIAMEX ALL-AVR). The also detected USB device 59 is a “USB-serial” type device. The ID 03eb is a vendor ID and the ID 2104 is a product ID of this ISP-programmer. Both IDs are required for an entry in the file `/etc/udev/rules.d/90-atmel.rules` and can be added with:

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

Of course you can select another editor than `xed`, if you like. In this example the file `90-atmel.rules` has one line:

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

This entry allows the access to the USB device 58 for members of the group `plugdev`.

Because the file location is in a system part of your file system, you need super user rights for building this entry. You can build this entry with any editor. For the `xed` editor your call would be:

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

You should add the above “SUBSYSTEM==” line to this file and close the file. You can also build the entry without any editor only with the command feature:

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

You should combine these two lines to one command line!


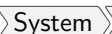
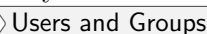
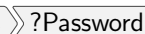




Then you should disconnect and replug your ISP-programmer. Now your system give you access to the device, if you are member of the `plugdev` group. Therefore your user identification should be a member of the group `plugdev` and also a member of the group `dialout` for USB-serial devices.. The correct setting for a non USB-serial type of ISP-programmer in your Makefile should be “PORT=usb”.

group membership

You can add the membership of your account to the groups `dialout` and `plugdev` with one command:

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

Now the membership of both groups should be established. You can check the membership of your account with the command `id`. If your membership is reported well, the program `avrdude` should now have permission to access a ISP programmer device with both interface types.

Another way to take a look to the group membership is to use a tool with graphical interface:    . You can also start the  with the  key (between  and ). But the layout is different with the type of desktop environment.

Working environment and compiling of sources

In order to preserve the original, it is recommended to create a duplicate of the sources with the name **Mytester**. Usually the home directory is called `/home/` followed by your user short name. The name of your home directory is stored in the system variable `$HOME`. You can also write `~/` in short instead of the full path name in commands. Don't forget the `/` after the `~` character, otherwise the command interpreted would try to find a user name with the following name! To do this task, first create an empty directory with:

```
mkdir ~/Mytester
```

If you have downloaded the transistortester archive in the directory `~/TransistorTester-source`, you can give the next instruction to copy the source files and subdirectories in the Mytester directory:

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

Because the `-v` Option is given to the `rsync` program, all copies are logged at the screen. If you had specified the `-au` option instead of the `-auv`, the `rsync` would do the same copy without any log. You can take a look to the copied subdirectory with:

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

A clear representation of the directory structure and the files is possible with the command

```
tree ~/Mytester
```


The tree command is not installed by default, which you can easily do with

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

If you now know which subdirectory is suitable for your tester, you can now change to this subdirectory. Let's assume you have a Chinese kit with a monochrome graphical display, then the subdirectory `mega328_st7565_kit` would be correct. The command to do this would then be:

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

method with graphical interface

You can also view the files in `~/Mytester/` with the Files window `Menu` `Accessories` `Files`, the directory `Mytester` can be selected by double-clicking `•○○`. Many subdirectories now appear here, including that desired `mega328_st7565_kit`. If you now select this directory with a double click on `•○○`, you will also see that `Makefile` is one of the files.

edit Makefile

If you have done that, you can now use the `○○●` key to activate the `open in terminal` action. A command window opens with the correct working directory, which is also directly active. Now all you have to do is type `make` and the transistor tester program is translated again. You now have two windows that are already in the correct working directory. The files window and the terminal window for entering commands (`make`). By double-clicking `•○○` on the `Makefile` file in the files window, another window with the preset editor opens. If you'd prefer a different editor here, you can select this editor either directly via `○○●` `Open with`. If you always want to set a different editor for text files, this is also possible via `○○●` `Properties`. Then a new `Makefile Properties` window opens, where you can select the function `Open With` by clicking `○○●`. Here you can now see the current setting and a variety of editing the `Makefile`. Here you can select another application by clicking `•○○`. By clicking `•○○` you can now choose whether you want this application `Add to list` or `Set as default`. You can also use the `Reset to system defaults` to reset all changes.

For the next step it is important that the settings for **your ISP programmer** are correctly set in the `Makefile`. See the subsection 4.1, on the page 55, topic **PROGRAMMER** and **PORT**. When you have checked these settings, you can reactivate the terminal window by clicking `•○○`. When the window is activated, the header has a higher contrast! After the working directory is selected, you can compile the transistor tester sources with `make`.

Finally, the time has come:





programming testers

If your ISP programmer is now completely connected, i.e. has a connection to the computer and the ATmega, you only need to enter the command:

```
make upload
```

The transistor tester program is now compiled again and then immediately loaded into the ATmega with the `avrdude` program. If the result does not look as expected, you can now immediately switch to the `Makefile` editor, make any necessary changes and repeat the process.

Tips to the terminal

You do not need to type in the command every time. Commands already issued in the terminal window can be displayed again with the  key. You can repeat the displayed command by pressing the  or  key. But you can also edit the displayed command before sending it, or switch to newer commands with the .

Possible Makefile calls

At last I would like to list the most important Makefile orders:

Instruction	Function
<code>make clean</code>	to clean old files
<code>make</code>	compile the transistor tester sources
<code>make fuses</code>	to set the ATmega “fuses” without crystal RC-clock
<code>make fuses-crystal</code>	to set the ATmega “fuses” for crystal operation!
<code>make upload</code>	to load the complete program with the ISP-programmer to the ATmega.

Hints for updates of the transistortester sources

The copy of the transistortester sources can be kept up to date with the command:

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

If you work in a copy under `~/Mytester` as recommended here, the changes will only be transferred if you also execute the command:

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

It can happen that the Makefile on the github server has a newer date than the locally changed copy. Then the changes made to the options in the Makefile locally in the Mytester folder would be lost. It is therefore a good idea to save the successfully modified Makefile as a copy. This can be done, for example, with the command:

```
cp Makefile Makefile.bak
```

Then you can use the command

```
diff Makefile Makefile.bak
```

to let the computer compare the files after an update. A better overview of the changes is also available with:

```
kdifff3 Makefile Makefile.bak
```

The kdifff3 program is probably missing because it is developed for a KDE desktop environment, but you can install it with:

```
sudo apt-get install kdifff3
```

4.2.2 Building the Software with Windows

Probably the easiest way right now is to install the Arduino IDE (Integrated Development Environment) first. This software package installs a actual version of the avr-gcc compiler, which can then also be used outside the IDE,s because the path to the programs is registered in the Windows PATH variable. Usually the programs are installed in C:\Arduino\hardware\tools\avr\bin. The avrdude program for operating the IDE programmer can also be found there. Only a GNU make program is missing to be able to use the Makefile I installed this program with the Cygwin64 package. To download the transistor tester software, the git program is used, which is also included in the Cygwin64 package an can be selected during installation. The Cygwin64 package uses its own installer setup-x86_64.exe, which can be used for new installations or updates as well as to add missing tools. The formerly shared transistor tester archive is now divided into three different github archives. The documentation and sources of old versions can be found at <http://github.com/kubi48/TransistorTester-old-versions>. The current documentation can be found at <http://github.com/kubi48/TransistorTester-documentation>. The last sources can be found at <http://github.com/kubi48/TransistorTester-source>. In a command window you can copy the archives with "git clone" and the respective address of the github archive into the local directory. The sources from github can be copied to your own computer with the following command:

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

If the command has been processed without errors, there should be a copy of the last source codes in the TransistorTester-source directory. The sources of the k-version described here are located in the trunk subdirectory. You can also find compressed tar archives (.tgz) of the last m-versions in the Markus subdirectory.

The source code of the k version in the trunk directory has several subdirectories, each with a Makefile. The different Makefiles are used to make adjustments to the different tester models. To find the name of the appropriate subdirectory, you can open the file picture-link.pdf in the TransistorTester-source directory with Acrobat Reader. There is one link or several links to a suitable subdirectory under each of the testers shown. However, there are currently not photos of all testers entered. To translate the sources, you have to open the command window and you must change into the subdirectory that corresponds to your tester. The following command sequence is required for the Chinese transistor tester kit with the graphic display (ST7565 controller):

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

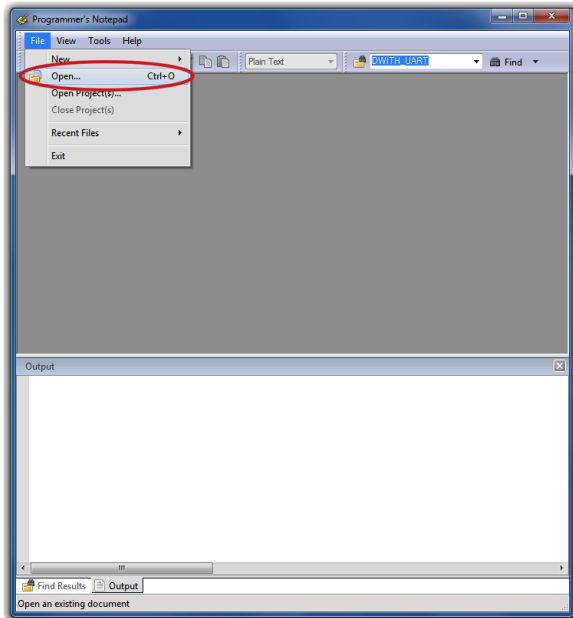
Of course, you can change the options in the Makefile with an editor of your choice before calling make. If you have adapted the PROGRAMMER and PORT settings to your ISP programmer in the Makefile, you can transfers the translated program to the connected transistor tester with a "make upload". With new AVR processors or if the correct setting of the fuses is not certain, you should set the fuses appropriately with "make fuses-crystal".

4.2.3 Using the WinAVR package with Windows

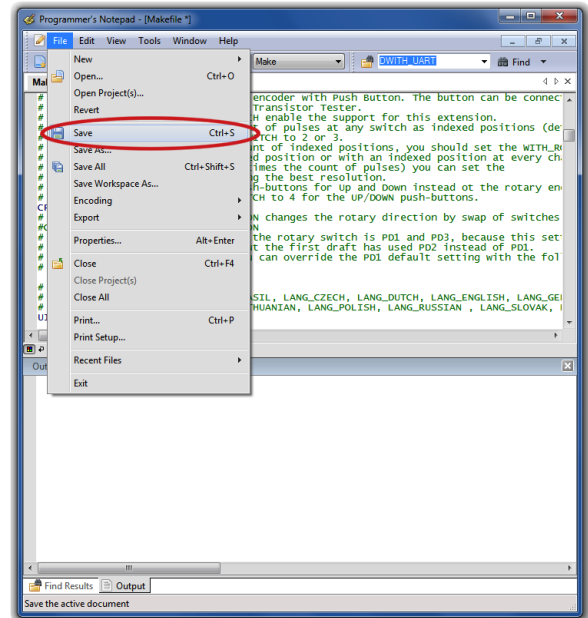
The WinAVR package can currently no longer be recommended because the package is very old and has not been maintained. The integrated avr-gcc compiler is therefore also very old. The newer versions of the avr-gcc compiler optimize the program much better. Because the ATmega memory is often full to the limit with well-optimizing compilers, you can only use WinAVR if you deselect functions in the Makefile. Then you can use the WinAVR package [18],[19]. With my patch [20]

you can also set the fuses by using the Makefile. Of course the `avrdude` program must support your programmer and the configuration in the Makefile must match to your environment. But the WinAVR package use a very old compiler version. You should better install the Arduino package and overwrite the avr executables.

The figures 4.2 show the File menu of the graphical user interface of WinAVR for open the file Makefile and for saving the changed Makefile (Save).



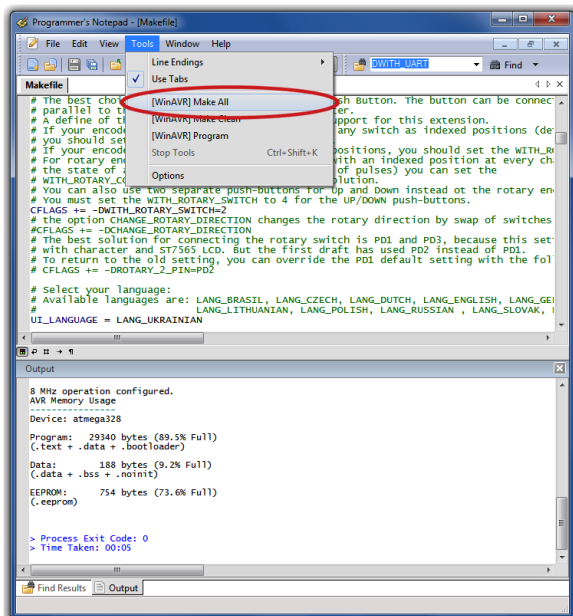
(a) open Makefile



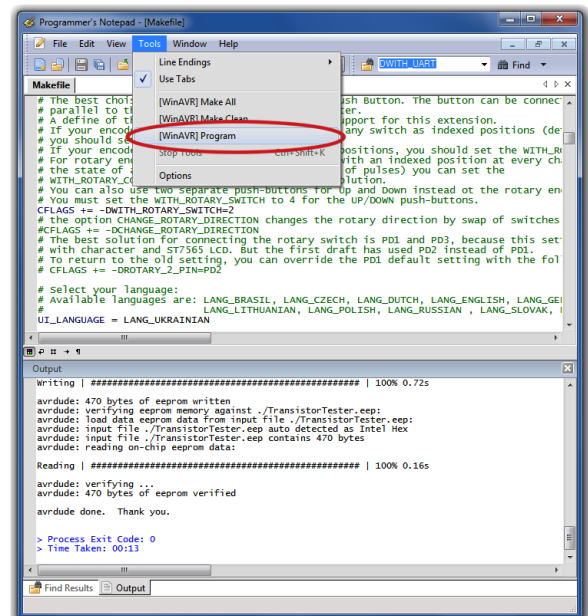
(b) save Makefile

Figure 4.2. Using of the WinAVR user interface Programmer's Notepad

The next figures 4.3 show the Tools menu of the Programmer's Notepad for compiling the program (Make All) and for programming the ATmega (Program) with `avrdude`.



(a) Build programming data (.hex/.eep)



(b) Programming the ATmega

Figure 4.3. Using of the WinAVR user interface Programmer's Notepad

4.3 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.

The tester software can be configured for many different controller with different connection types. You should check, if your software matches to your mounted display type. 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. If the tester is build without the switch off electronic, you can test with a LED connected to the test pins, if the program operates normally. If the LED flickers, the program operates well. The missing text on the LCD must be caused by wrong connection or timing. For some graphical displays the contrast is changeable with a menu function. If you have changed the contrast value, that nothing is readable at the screen, you cannot handle the menu function any more. You can try to read the display from a slanting look to the display, not from the front side. In this case you can try to handle the menu function with this view. Otherwise you can write the EEprom data new with a ISP programmer to reset the contrast value.

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 5.5.

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.

The Tester switch off the power after 2 seconds display time This condition exists, if the external Pull-Up resistor at the PD7 input is missing or the key button is keep pressed. The software switch off the internal Pull-Up resistors to prevent a influence to the measurement results. Therefore a external Pull-Up resistor (27k) is required.

Der Tester shows only Vext=xx.xV in row 2 This problem exists, if the Pull-Up resistor at the PD7 input is missing or the key button is keep pressed. Additionally the software is configured without the serial output (without option WITH_UART) and without the internal Pull-Up resistors (with option PULLUP_DISABLE). You should install the Pull-Up resistor at pin PD7.

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.

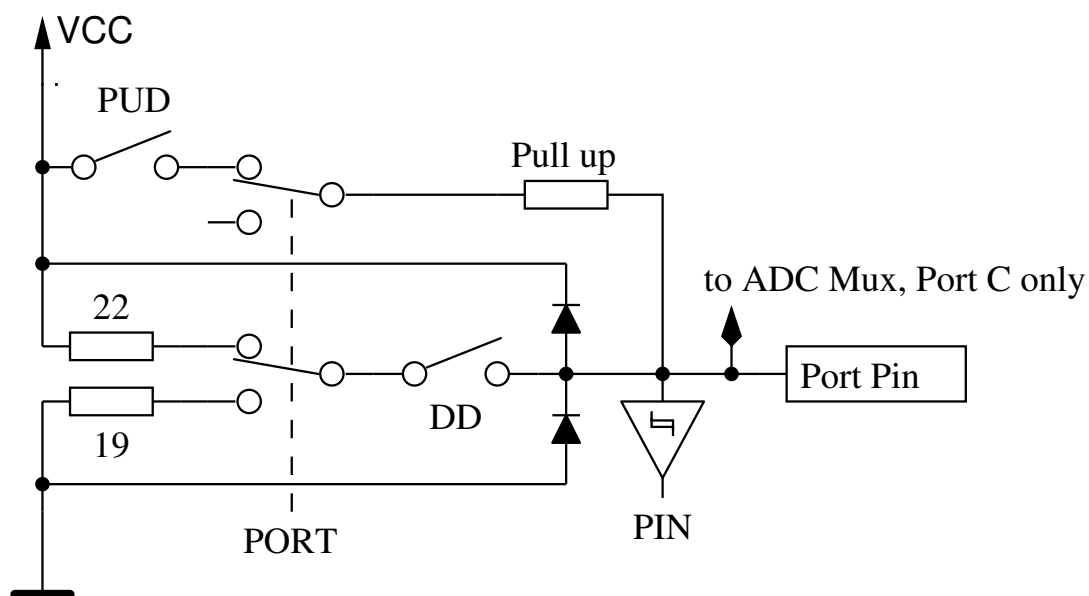


Figure 5.1. simplified diagram of each ATmega port pin

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.

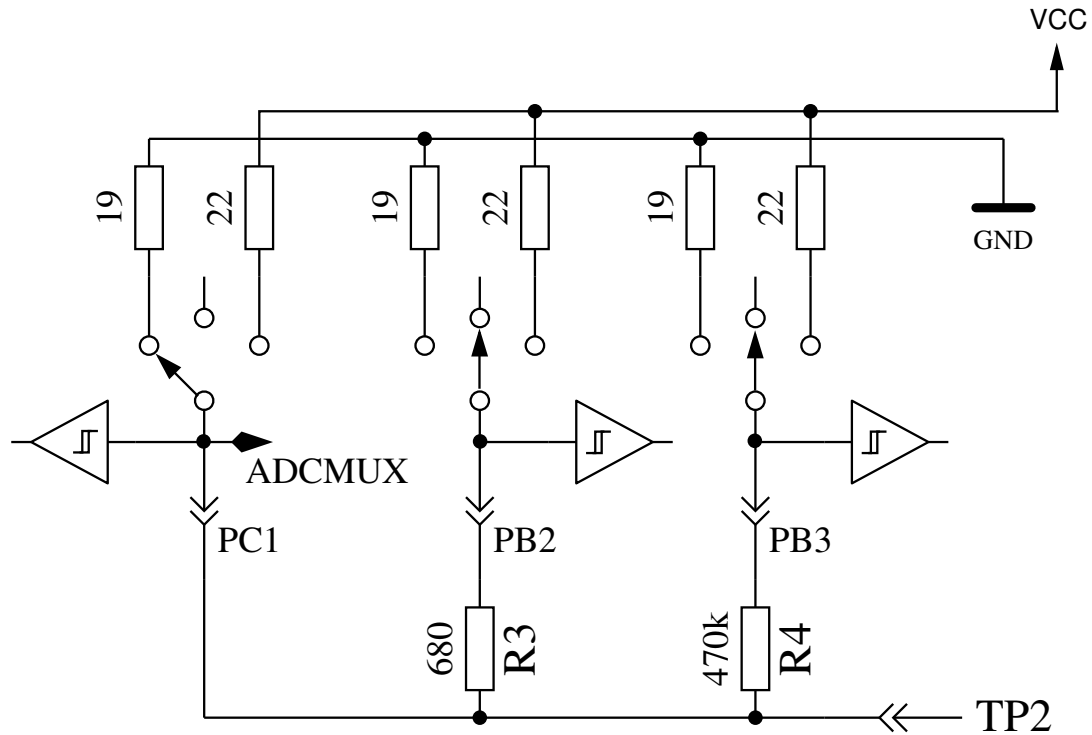


Figure 5.2. simplified circuit of each measurement terminal probe TP

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 470kΩ 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 470kΩ 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

The currentflow of the device with currentless control gate (third pin, also called Tristate pin) is to be examined first. The Tristate pin of the device under test is the base or gate for example. One probe pin is selected as the positive side of the device and connected directly to VCC. The other probe pin selectes as negative side of the device. The negative side is connected with the $680\ \Omega$ resistor to GND. With fieleffect transistors the state of the device depends on the voltage of the gate. The Tristate pin is first connected with the $680\ \Omega$ resistor for $5\ ms$ to the GND side and the voltage at the negative side is measured. After that the voltage of the negative side is measured again during the Tristate pin switched as input (High Impedance). Then the assumed gate is connected with the $680\ \Omega$ resistor for $5\ ms$ to the VCC side and the voltage on the negative side is measured again. If the measured voltage is lower than the first measurement result, this circuit will be assumed as the right one. Then the voltage is measured again with currentless Tristate pin.

If the voltage of the negative pin with fixed Tristate pin is higher than $115\ mV$ and this level is not $100\ mV$ lower than the voltage measured with currentless Tristatepin, a depletion transistor type is assumed. With bipolar transistors, which have a high collector residual current, the residual current with currentless base is usually significant higher. With the checking of both voltages we can avoid the wrong detection of some Germanium transistors with a higher collector cutoff current as depletion transistors (JFET).

Then additional tests are done to differ N-channel JFET or N-D-MOSFET and P-channel JFET or P-D-MOSFET. The MOSFET versions can be differed by the missing of gate current in any state of the TriStatePin.

To get parameters of the depletion types, they will be measured with a $680\ \Omega$ resistor at the source pin, as shown in figure 5.3 . This measurement will be done instead of the usually measurement of current with the gate hold at source level, because the I_{DSS} current of the FET transistor can often not be reached with the relative high resistance of the $680\ \Omega$ resistor.

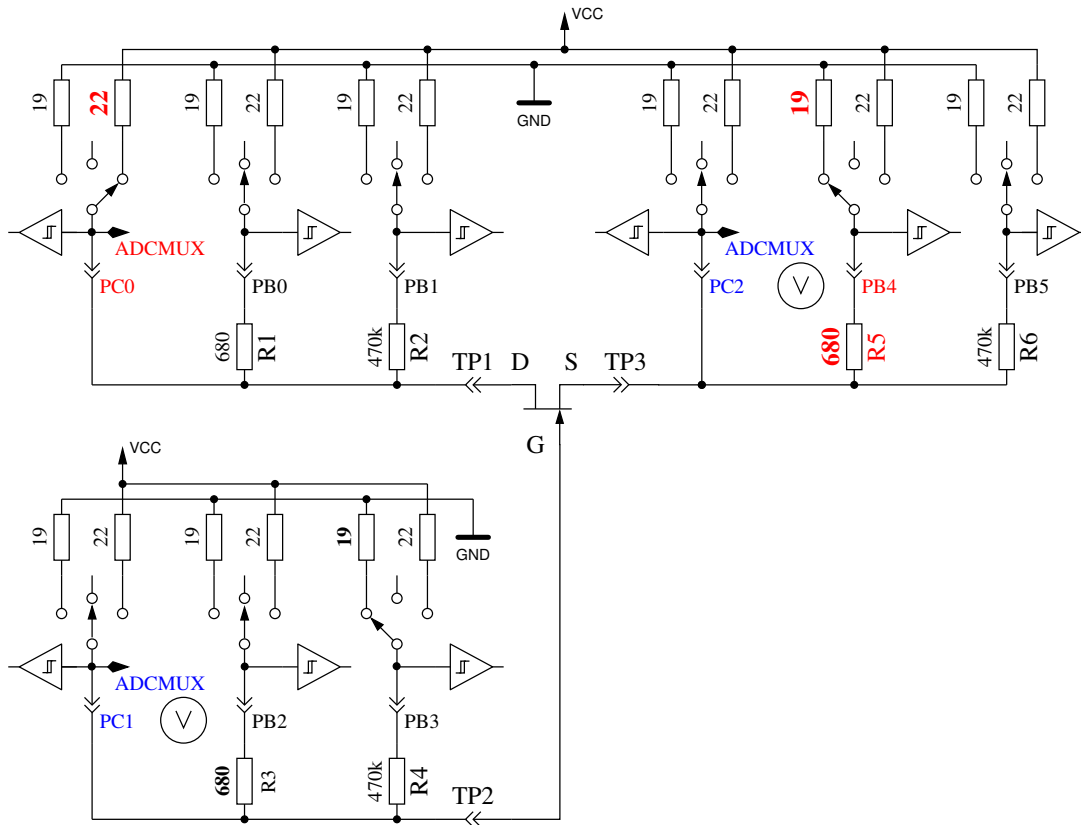


Figure 5.3. Measurement of the Gate-Source voltage and Source current of a N-JFET transistor

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 5.1.4.

5.1.1 Measurement of PNP Transistor or P-Channel-MOSFET

First the current amplification factor is measured with common collector (emitter follower) for the assumed PNP transistor. The measuring situation is shown in figure 5.4. If the measured voltage at the Base (U_B) is above 9 mV with the 680 Ω resistor, the hFE is build as $hFE = \frac{U_E - U_B}{U_B}$. The voltage U_E is the difference of the Emitter-voltage to VCC. The difference between the 22 Ω and 19 Ω resistors are not respected. If the U_B voltage is below 10 mV, the measurement is done with the 470 k Ω resistor at the base. In this case the current amplification factor is build as $hFE = \frac{U_E \cdot 470000}{U_B \cdot (680 + 22)}$.

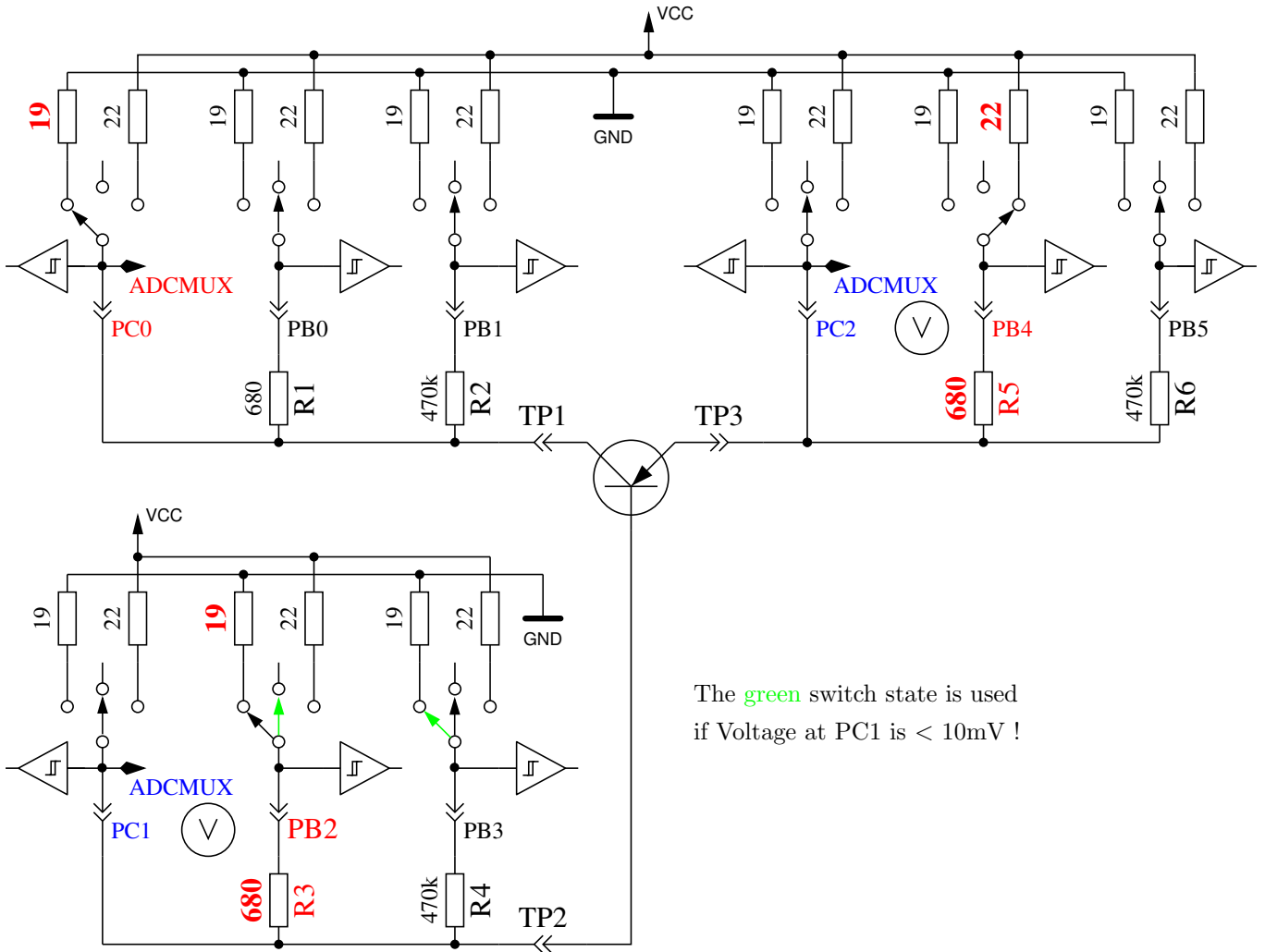


Figure 5.4. hFE measurement of PNP transistor with common collector circuit

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 680 Ω resistor is connected to GND as shown in Figure 5.5. If the negative side of component has a voltage of above 3.4 V, when the base side 680 Ω 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 0.97 V, it must be a PNP. For measuring the current amplification factor, the 470 k Ω resistor is taken as Base resistor instead of the 680 Ω . The current amplification factor is build by $hFE = \frac{(U_C - U_{C0}) \cdot 470000}{U_B \cdot (680 + 19)}$. The voltage U_{C0} is the voltage at the collector resistor without base current. 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 0.97 V, 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 Ω resistor up and down, waiting for a digital input signal change of the Drain side and then read the voltage of the gate pin.

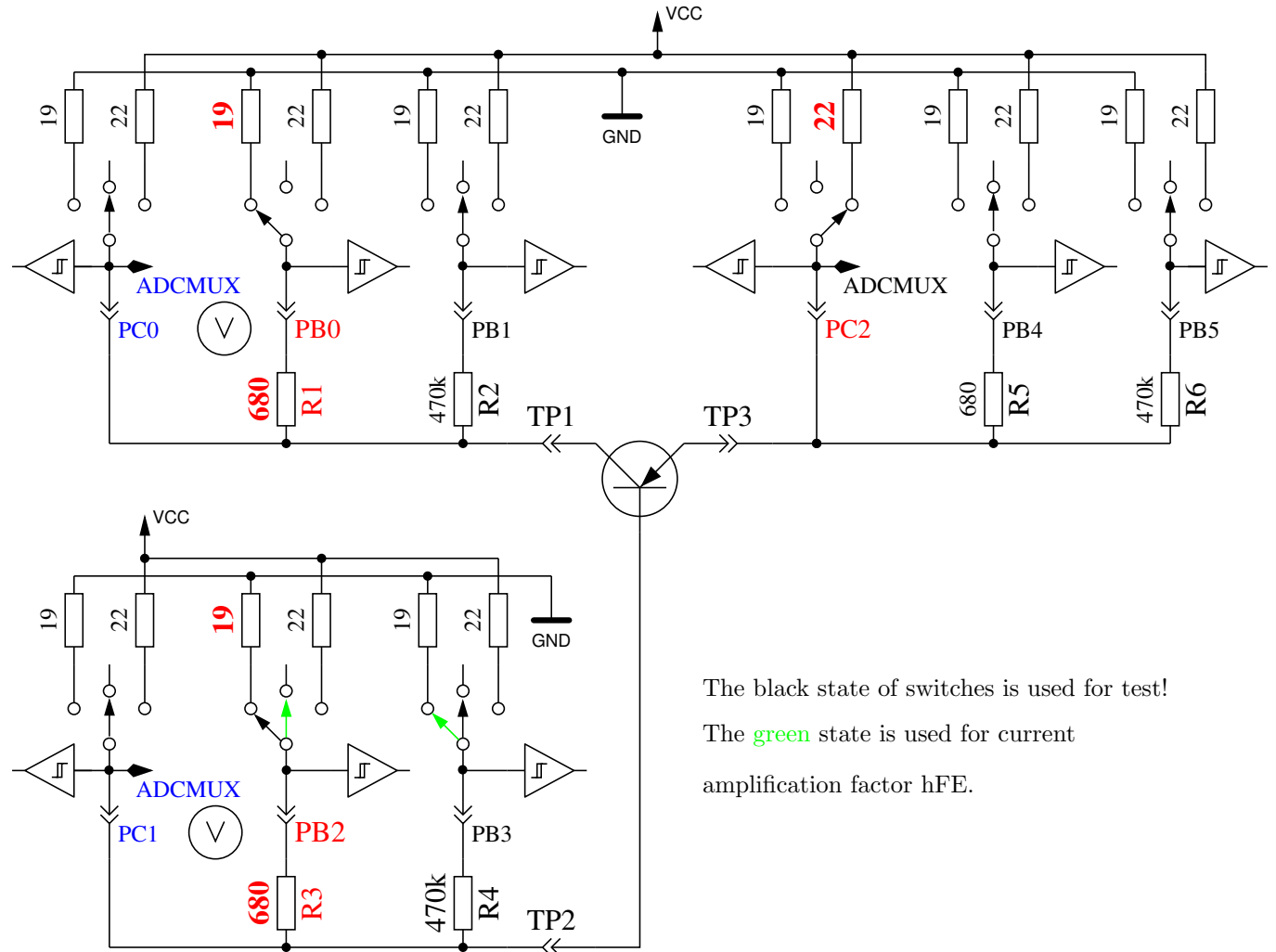


Figure 5.5. test and hFE measurement of PNP transistor with common emitter circuit

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 680 Ω base resistor switched to VCC. If the voltage at the base resistor ist too low, the 470 k Ω resistor is taken instead. Measurement then continues with the common emitter circuit as shown in figure 5.6.

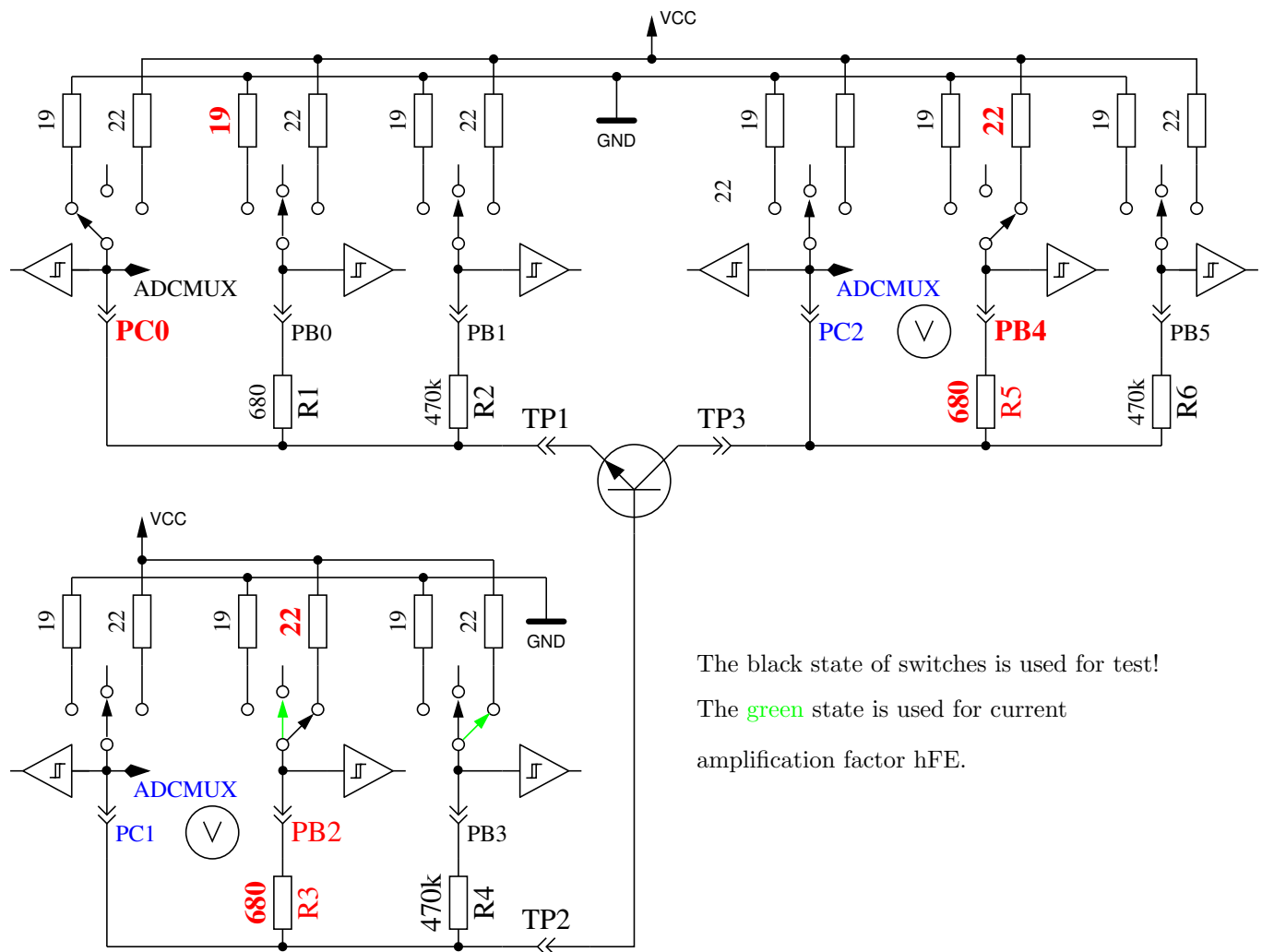


Figure 5.6. test and hFE measurement of NPN transistor with common emitter circuit

If the voltage of collector sinks below 1.6 V , when the $680\ \Omega$ base resistor is connected to VCC , it 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 10 ms to GND and then 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 6 mA . 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 $hFE = \frac{(VCC - UC - UC0) \cdot 470000}{(VCC - UB) \cdot (680 + 22)}$. 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 $470\text{ k}\Omega$ 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 5.7. The result is multiplied by four and divided by 9 to get the voltage in mV resolution.

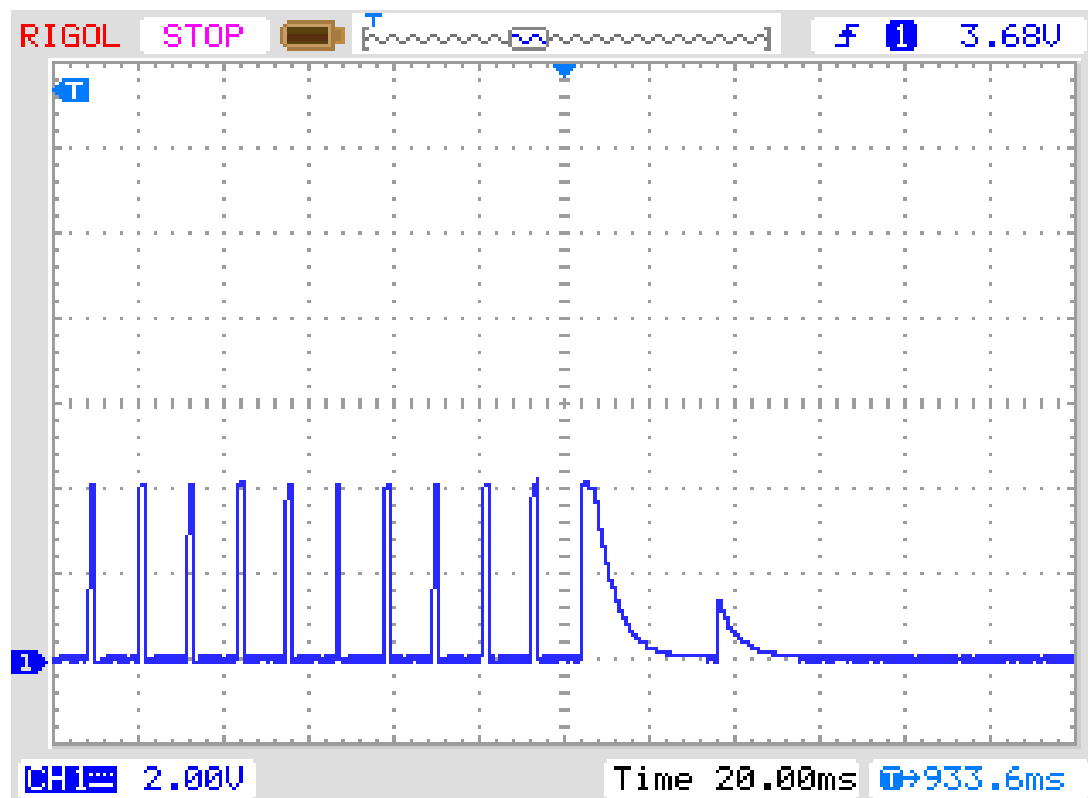


Figure 5.7. measuring of threshold voltage of N-Channel-MOSFET

5.1.3 Simplified flowchart of the transistors tests

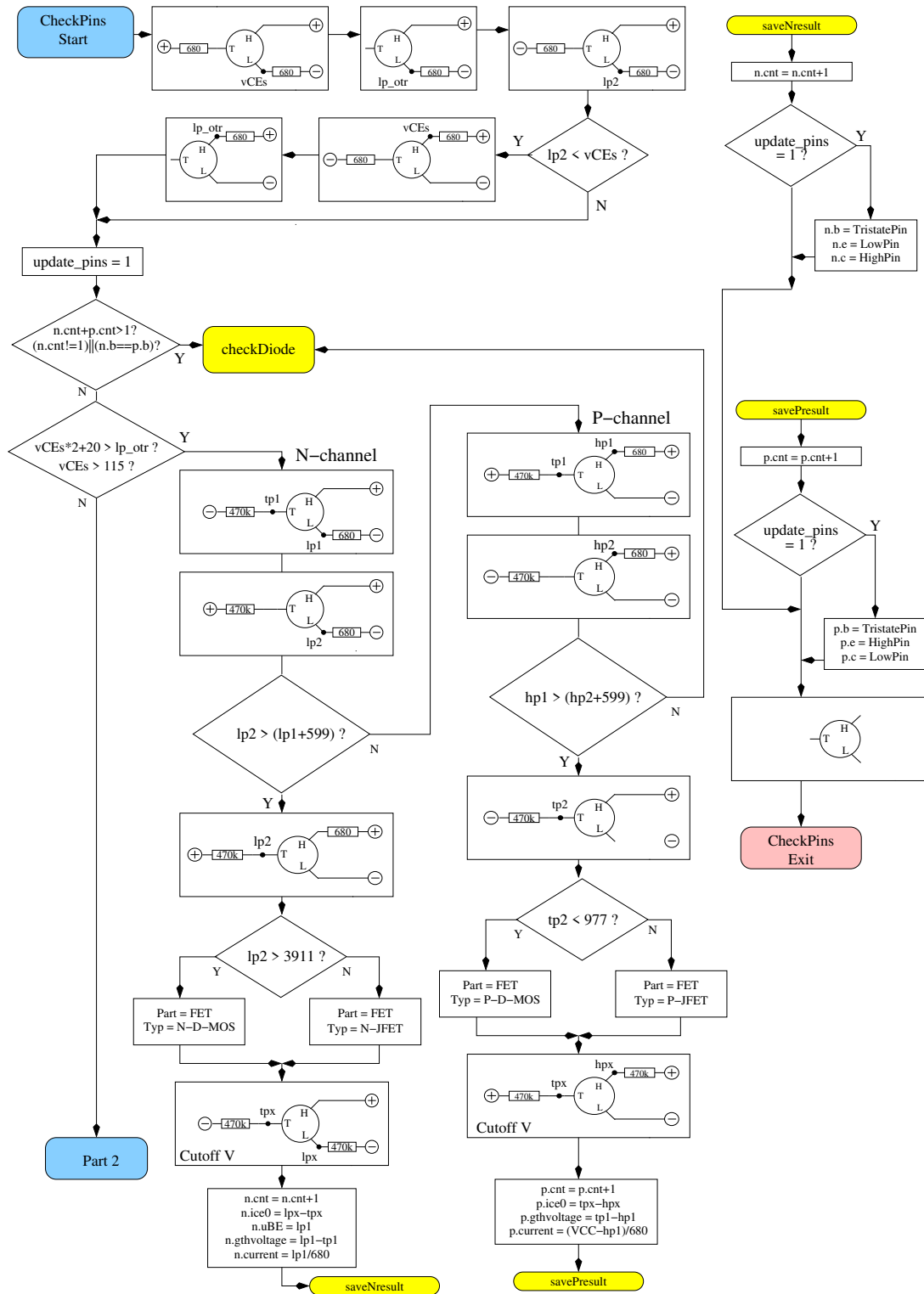


Figure 5.8. Flowchart transistor test Part 1, JFET and D-MOS

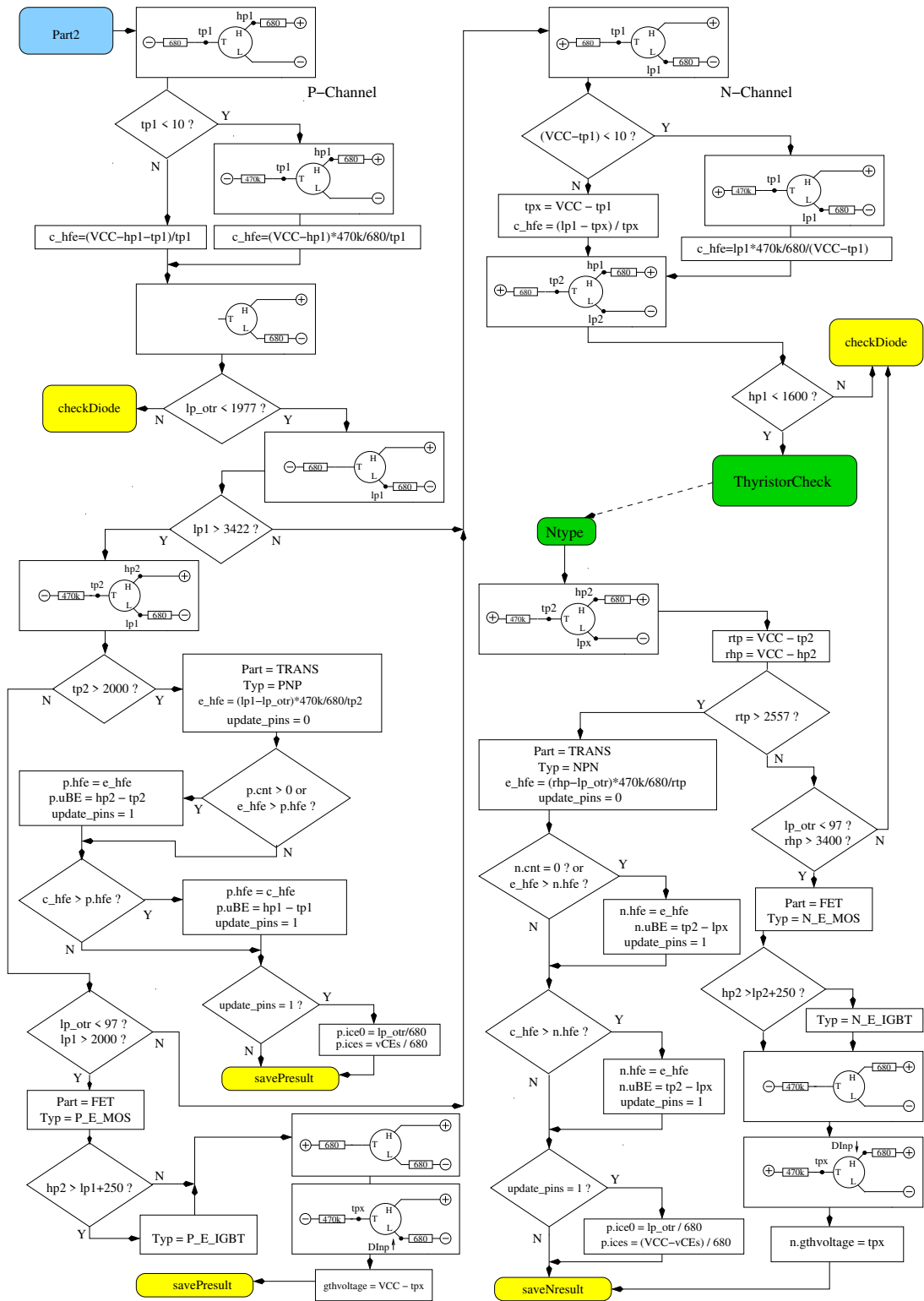


Figure 5.9. Flowchart transistor test Part 2, BJT and E-MOS

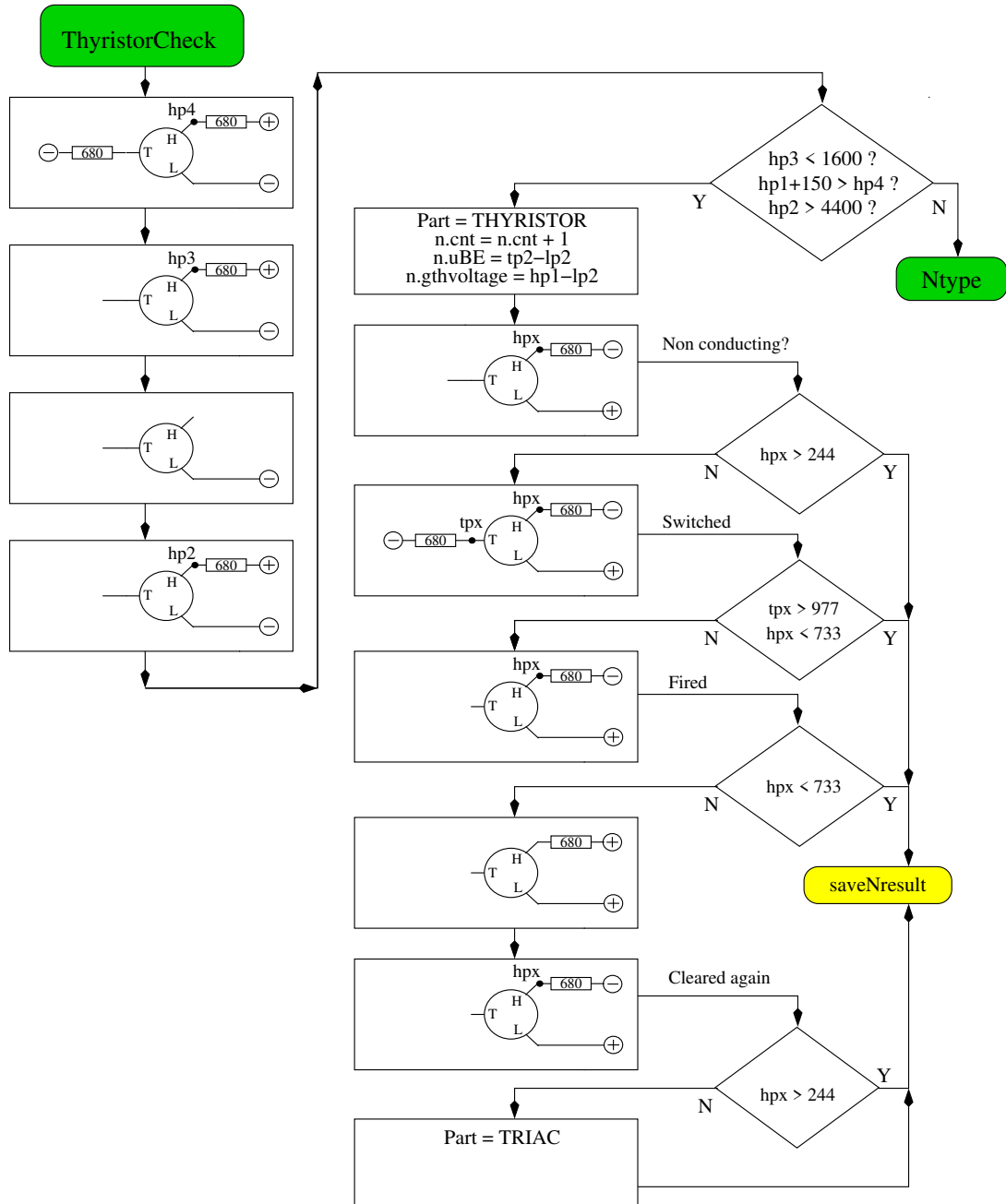


Figure 5.10. Flowchart transistor test Part 3, Thyristor and Triac

5.1.4 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 $680\ \Omega$ resistor must be between $0.15\ V$ and $4.64\ V$. The flux voltage with the $680\ \Omega$ must be greater than 1.125 times the flux voltage with the $470\ k\Omega$ resistor and sixteen times the flux voltage with the $470\ k\Omega$ must be greater than the flux voltage with the $680\ \Omega$ resistor. Additionally the afterward renewed measurement with the $470\ k\Omega$ resistor should not have a higher voltage than the previous measurement with the $680\ \Omega$ resistor. I hope, that this behavior identifies always a diode. The identification of a diode by no current flow in the opposite direction is not possible with a inverse parallel diode. If only a single diode is detected, the residual current in reverse direction is measured with the $470\ k\Omega$ resistor at $5\ V$. The resolution is about $2\ nA$. If the residual current is greater as $5.3\ \mu A$ (voltage at the $470\ k\Omega$ is more than $2.5\ V$), the measurement is done with the $680\ \Omega$ instead. Then the resolution is only about $1\ \mu A$. Furthermore the capacity in reverse direction is also measured for single diodes.

5.1.5 Results of different measurements

The following three tables shows results of different test probes with one ATmega8, a ATmega168 and a ATmega328 processor. The measurement of the inverse capacity value for the double diode MBR4045PT is only possible with cooling. This will be caused by high residual current of this 40A diode. Also the capacity value of the inverse base emitter diode of the germanium transistor AC128 can only be measured with cooling.

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

Table 5.2. measurement results of diode testing

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

Table 5.3. measurement results of bipolar transistor testing

Some results are very different to the earlier results of the software of Markus Frejek. For example a darlington transistor BC517 has been measured by the older software with a hFE of 797 instead of 77200 and a base emitter voltage of 1438 *mV*. This will be caused by the additional measurement of current amplification with the common collector circuit. Also the new version shows the same low hFE result with the common emitter circuit, as you can see in the last column of table 5.3. The base emitter voltage is measured by the older Version as separate diode test with 1438 *mV*. Now the base emitter voltage is measured with the state of current amplification testing (1.20 V). The BUL38D Transistor has a build in protection diode over the anode and cathode of the NPN transistor, by what a parasitical PNP transistor with swapped Base - Collector connection is build. With software revision 1.10k both transistors are detected and marked with a appended p. The right transistor will be found with comparison of the gate - emitter junction capacitance. It is assumed, that the right transistor has the higher junction capacitance. If you hold down the start key during the output of the measurement result, the parameter of the parasitical transistor are shown. With the label PNPn the existance of another transistor will be marked. The parasitical transistor structure is build only by integration of the protection diode nearby the transistor within the same material, not with a external diode.

The following table 5.4 shows the measurement results for germanium transistors, which are extra problematic to measure because of the temperatur dependent and high residual collector current. The results of the original version of Markus F. and the results of the actual 1.10k version are compared together. The 1.10k version for a ATmega328 measures the current amplification factor with common collector and common emitter circuit with respect to the collector residual current, the higher result will be shown. The collector residual current is not respected by earlier versions.

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

Table 5.4. Measurement results of bipolar junction germanium transistors

In the table 5.5 the results of some field-effect transistor measurements are shown. One measured parameters of the E-type MOS types is the gate-source voltage, by which the digital input of the ATmega connected to the $680\ \Omega$ drain resistor changes the state. For very fast change of the gate voltage due to a small gate capacity, the detected voltage is slightly inaccurate. With the BS250 the Voltage changes from $2.6\ V$ to $2.5\ V$, if you connect a additional $10\ nF$ capacitor to the gate-source. Another imeasured parameter is the gate capacity value. The gate capacity is measured by switching the source and drain pin to the GND potential. The available $5V$ gate voltage of the tester is unsufficient to generate some collector current with some IGBTs. This will prevent the correct detection. Only the collector-emitter protection diode is detected then in most cases. A Battery with about $3V$ connected to the gate pin can solve this problem with the detection. The other pole of the battery should be connected to a test pin (TP) of the tester instead of the gate connection. With the right polarity of the battery the tester should detect the IGBT now. The shown gate-emitter switching voltage must be increased by the value of the battery voltage to get the right switching voltage.

For JFET transistors often the characteristic current I_{dss} is specified, the current in the drain when the gate-source voltage is $0\ V$. Here, however, the current is given by a $680\ \Omega$ load resistance at the source side of the JFET. The load resistor generates a reverse voltage V_{gs} , which is also shown. With a $470k\Omega$ load resistor at the source side of the JFET the Source-Drain current will be nearly zero. With this circuit we can get the Gate-Source Cutoff voltage V_{gs_off} exactly enough, if the voltage remain below $5V$. With this two operating points we can estimate the current I_{gss} with the nearly quadratic characteristic curve of the current. If the estimated current I_{dss} stay below $40mA$, a additional measurement is done without a additional resistor at the source pin. With the measured voltage at the source pin we can compute a additional current value. Now we can compute a better estimated current I_{dss} with this higher current valued, the gate-source voltage and with the known quadratic current curve, if the value of $40mA$ is not exeeded. Due to the symmetrical design of the JFET transistors, the drain and source can not be distinguished.

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

Table 5.5. 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 5.11 and figure 5.12 as a example of the six choices of probe combinations.

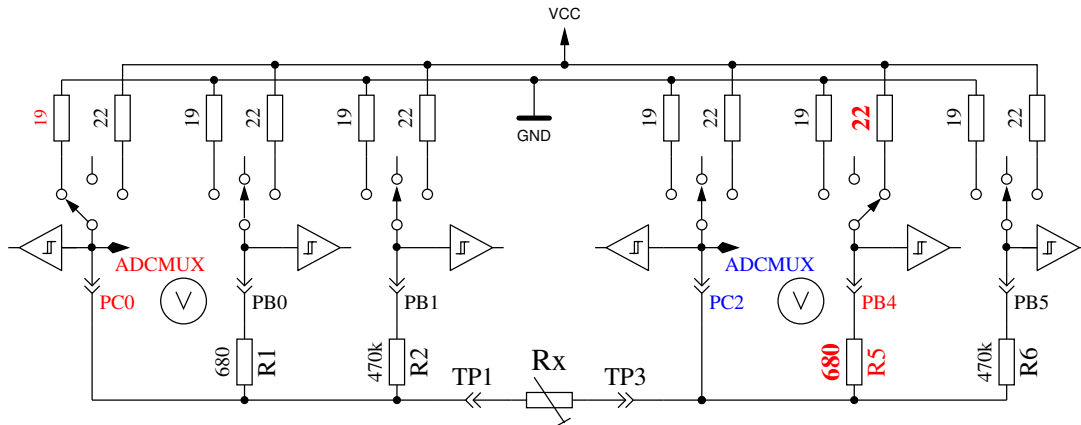


Figure 5.11. Measurement type 1 with 680 Ω

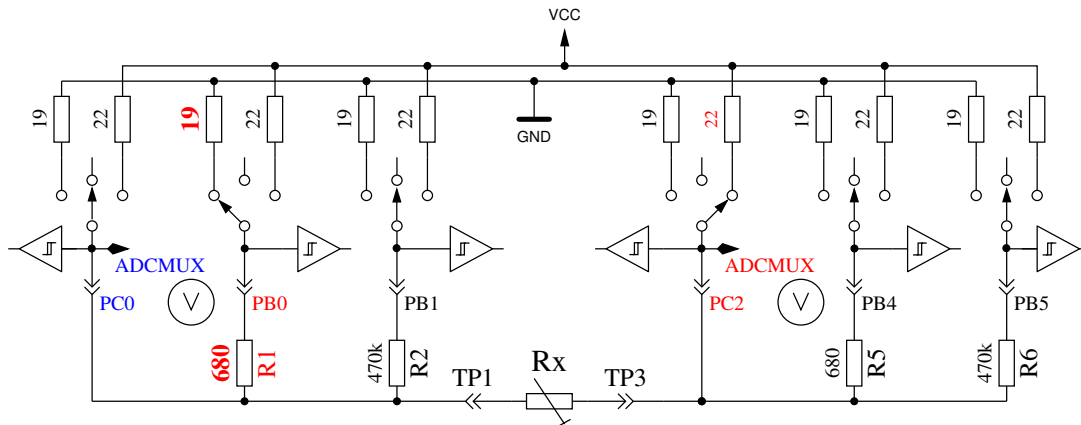


Figure 5.12. Measurement type 2 with 680 Ω

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 R_x can be computed with the known resistor values and the measured voltages,

if the relation of resistor R_x and the 680Ω is not too high. The theoretical voltage gradient is shown in figure 5.13, where resistor values are shown in logarithmic scale.

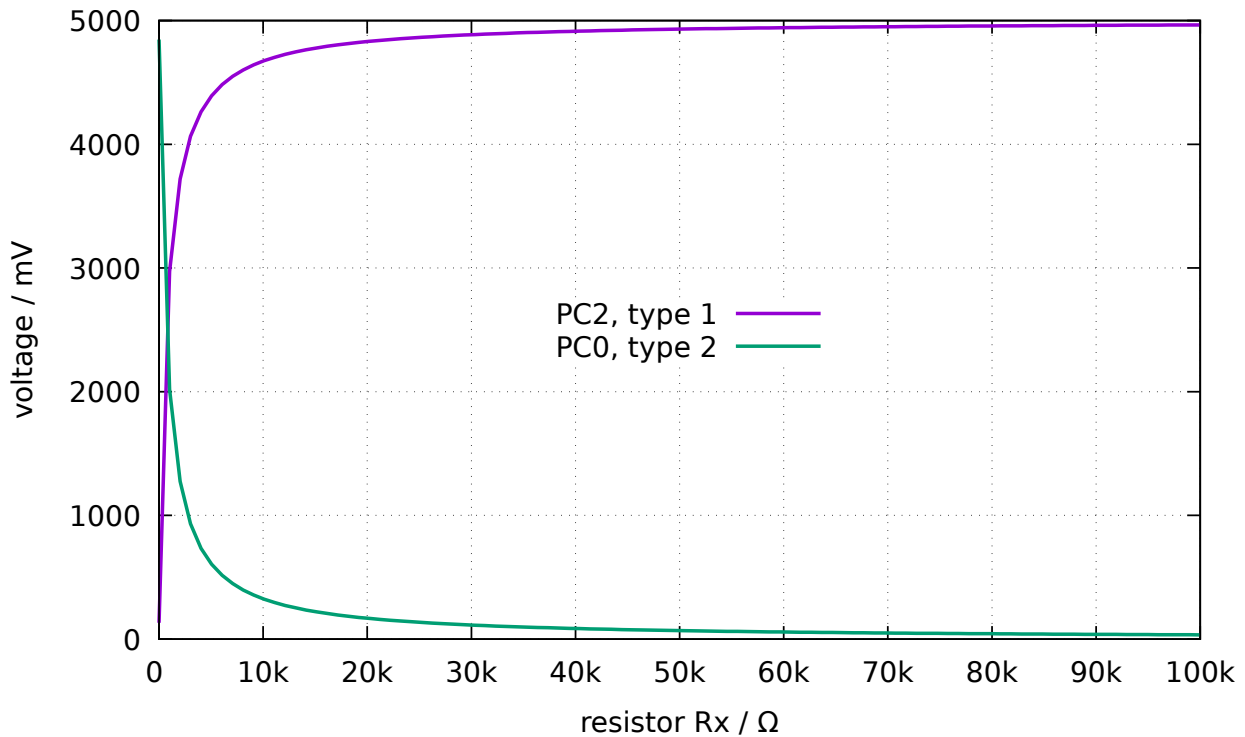
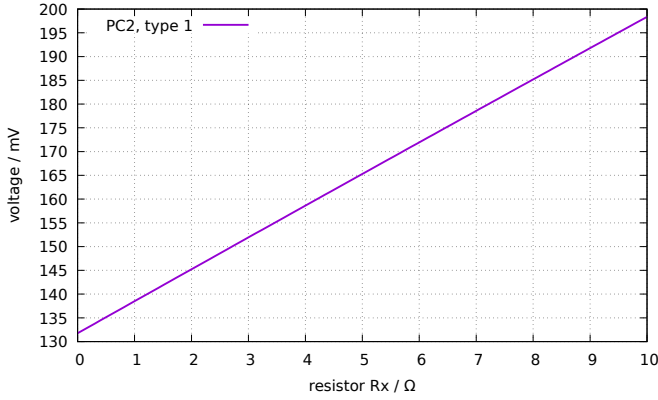


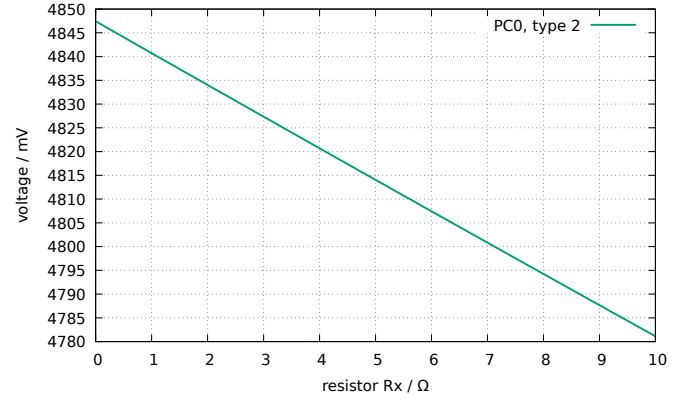
Figure 5.13. Voltages of type 1 and type 2 measurements with 680Ω

The graph of measurement type 1 is shown in figure 5.14a 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 5.14b. 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 of measurement type 2 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 it 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 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 $1.1V$. If the ATmega has more than $8 KByte$ flash memory, the voltage measurement at the resistors will be delayed until no more changes are detected or the time limit is reached. With this method big capacitors are no more detected as resistors by mistake and the DC resistance of big inductors will be measured correctly.



(a) Type 1 measurement



(b) Type 2 measurement

Figure 5.14. Cut-out of theoretical Voltage from 0Ω to 10Ω

5.2.2 Resistor Measurement with 470 kOhm resistors

The next figures 5.15 and 5.16 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 5.17 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.

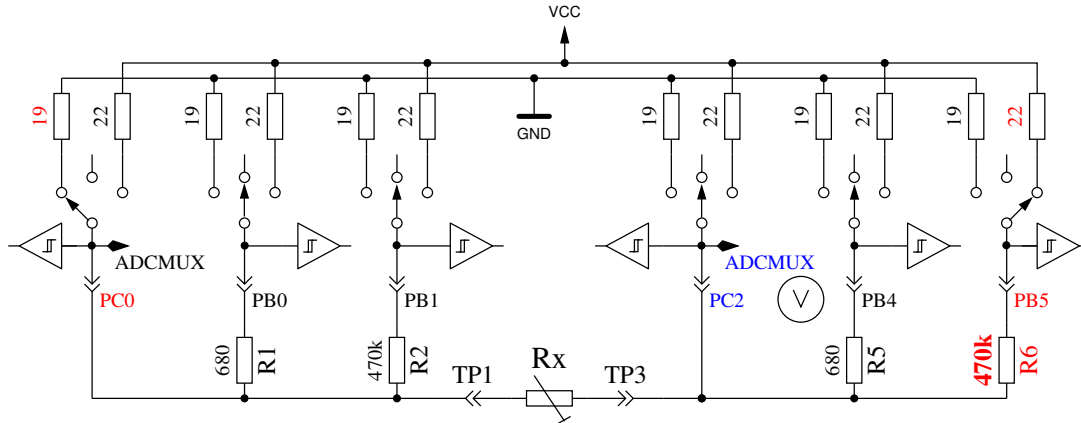


Figure 5.15. Measurement type 3 with $470k\Omega$

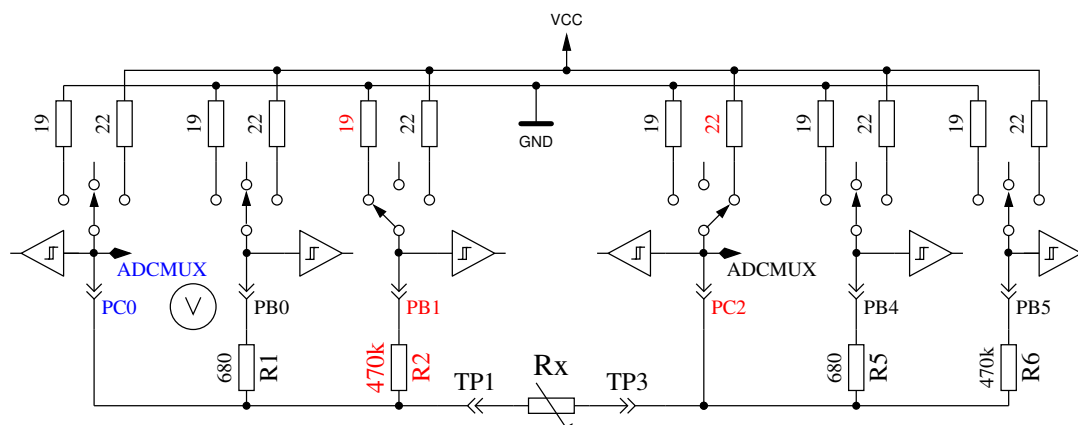


Figure 5.16. Measurement type 4 with 470kΩ

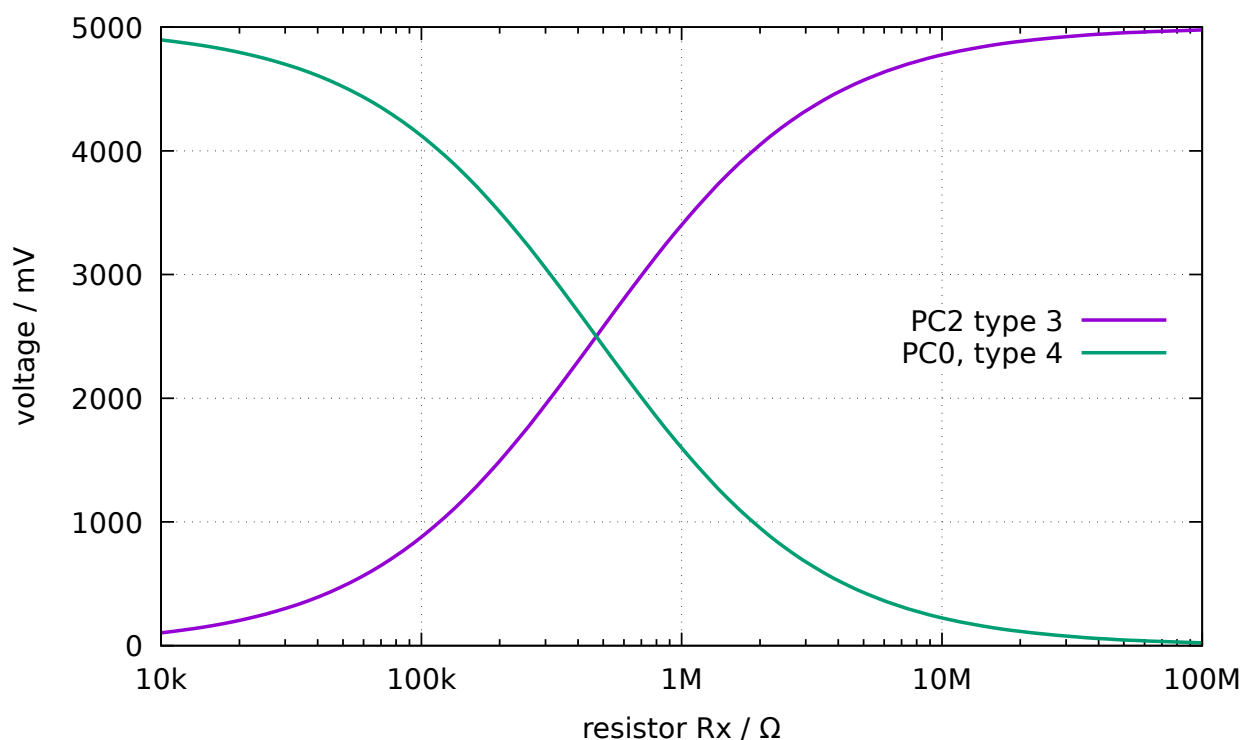


Figure 5.17. Voltages of type 3 and type 4 measurements with 470kΩ

5.2.3 Results of the resistor measurements

Figure 5.18 shows the relative errors of the resistor measurements with three ATmega8 microcontrollers. Additionally some results with the original software of Markus F. with one ATmega8 are shown as “Mega8orig” in this figure. More measurements results with ATmega8A and ATmega8L are shown in figure 5.19a and 5.19b. Figure 5.20 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%!

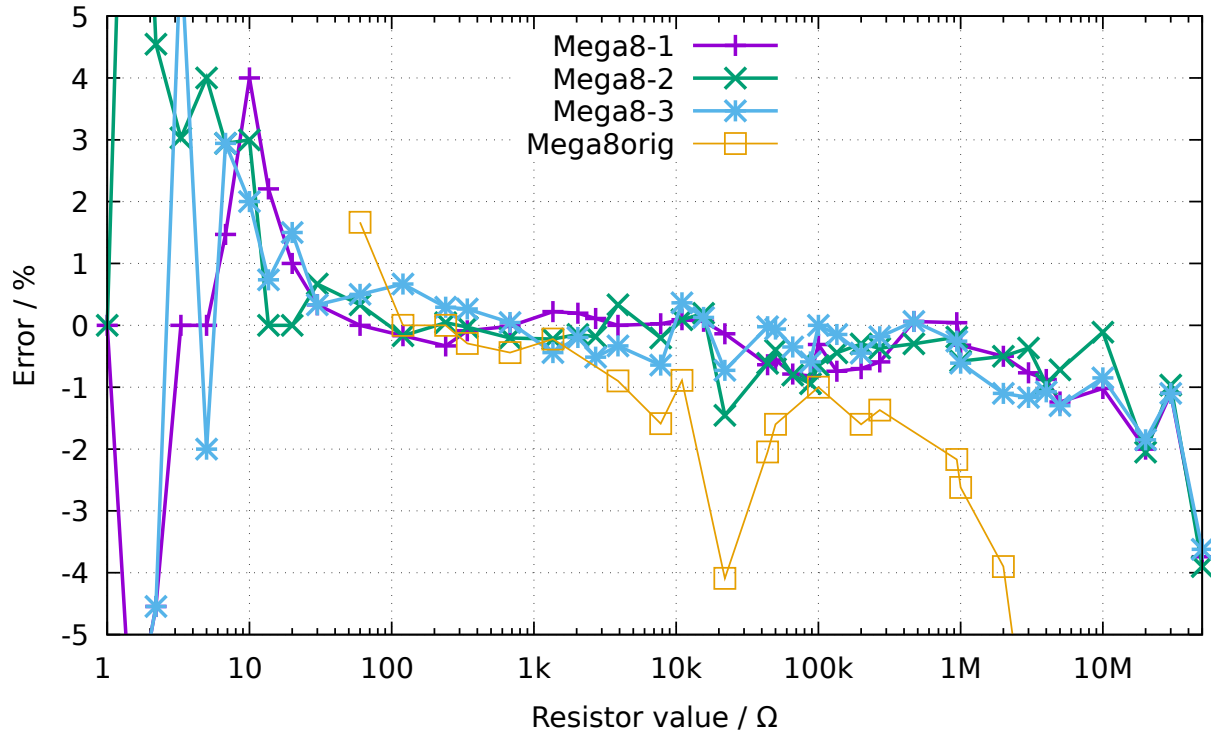
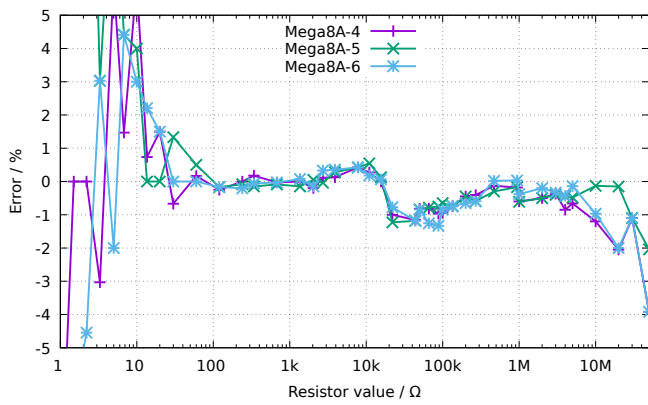
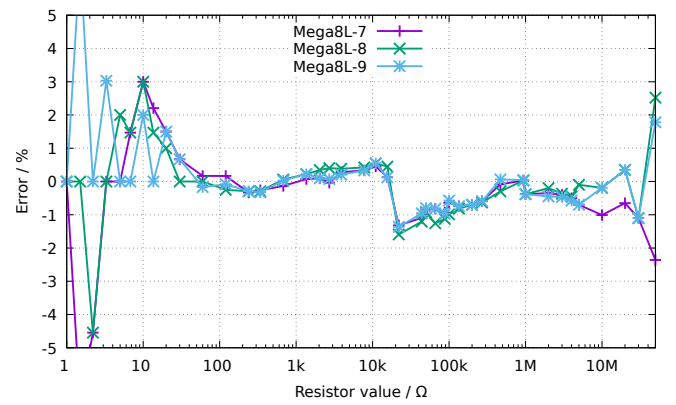


Figure 5.18. Relative error for resistor measurements with ATmega8



(a) with three ATmega8A



(b) with three ATmega8L

Figure 5.19. Relative error for resistor measurements

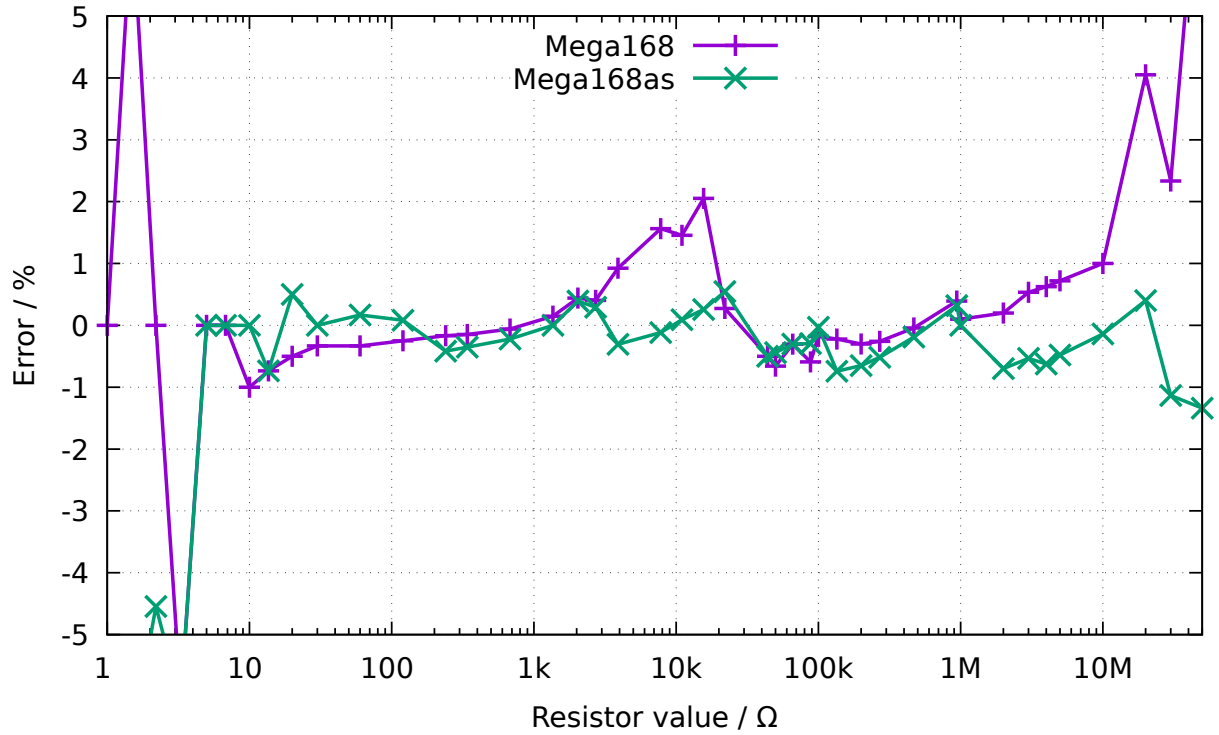
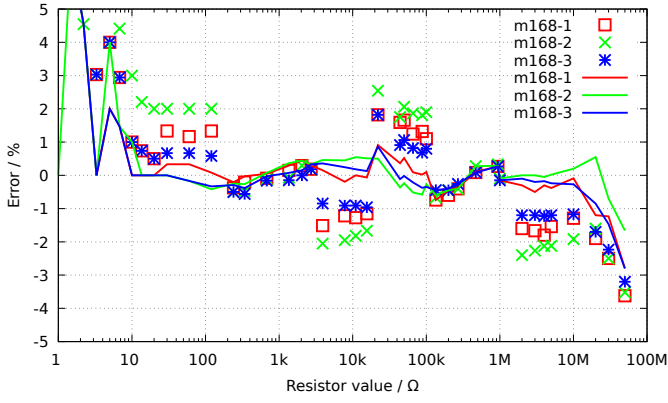
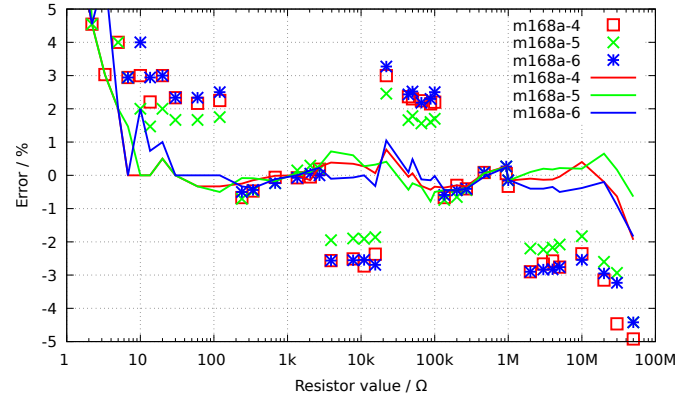


Figure 5.20. Relative error for resistor measurements with ATmega168

The figure 5.21a 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 5.21b and the measurement errors of three ATmega168P processors are shown in figure 5.22. The measurement errors of three ATmega328 processors are shown in figure 5.23a and 5.23b. 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!



(a) with three ATmega168



(b) with three ATmega168A

Figure 5.21. Relative error for resistor measurements

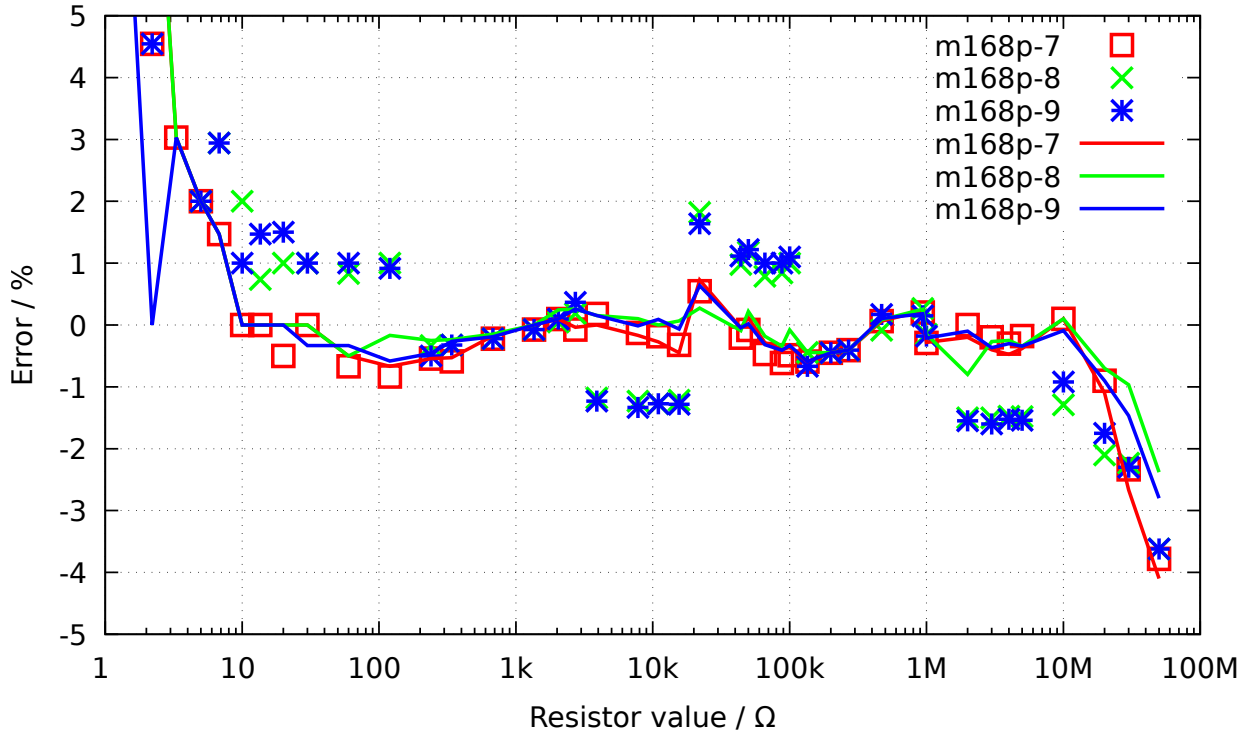
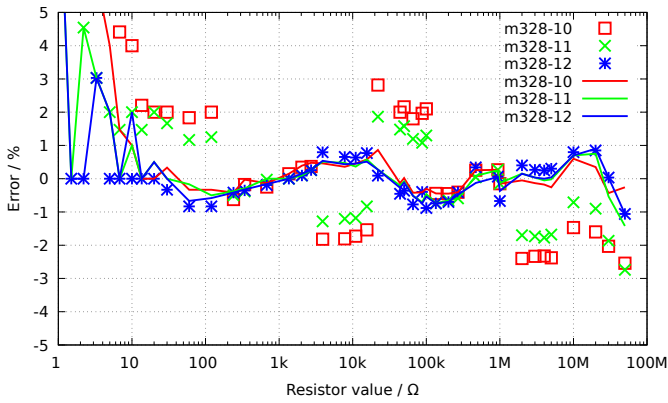
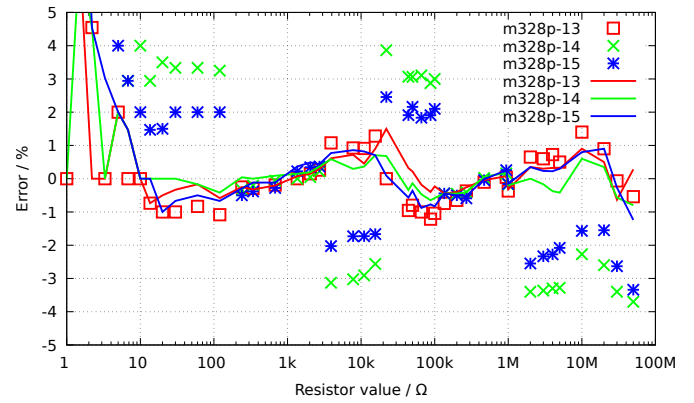


Figure 5.22. Relative error for resistor measurements with three ATmega168P



(a) with three ATmega328



(b) with three ATmega328P

Figure 5.23. Relative error 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) [4] 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 big capacitors of more than $15mF$ 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 $10ms$ 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 5.24 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.

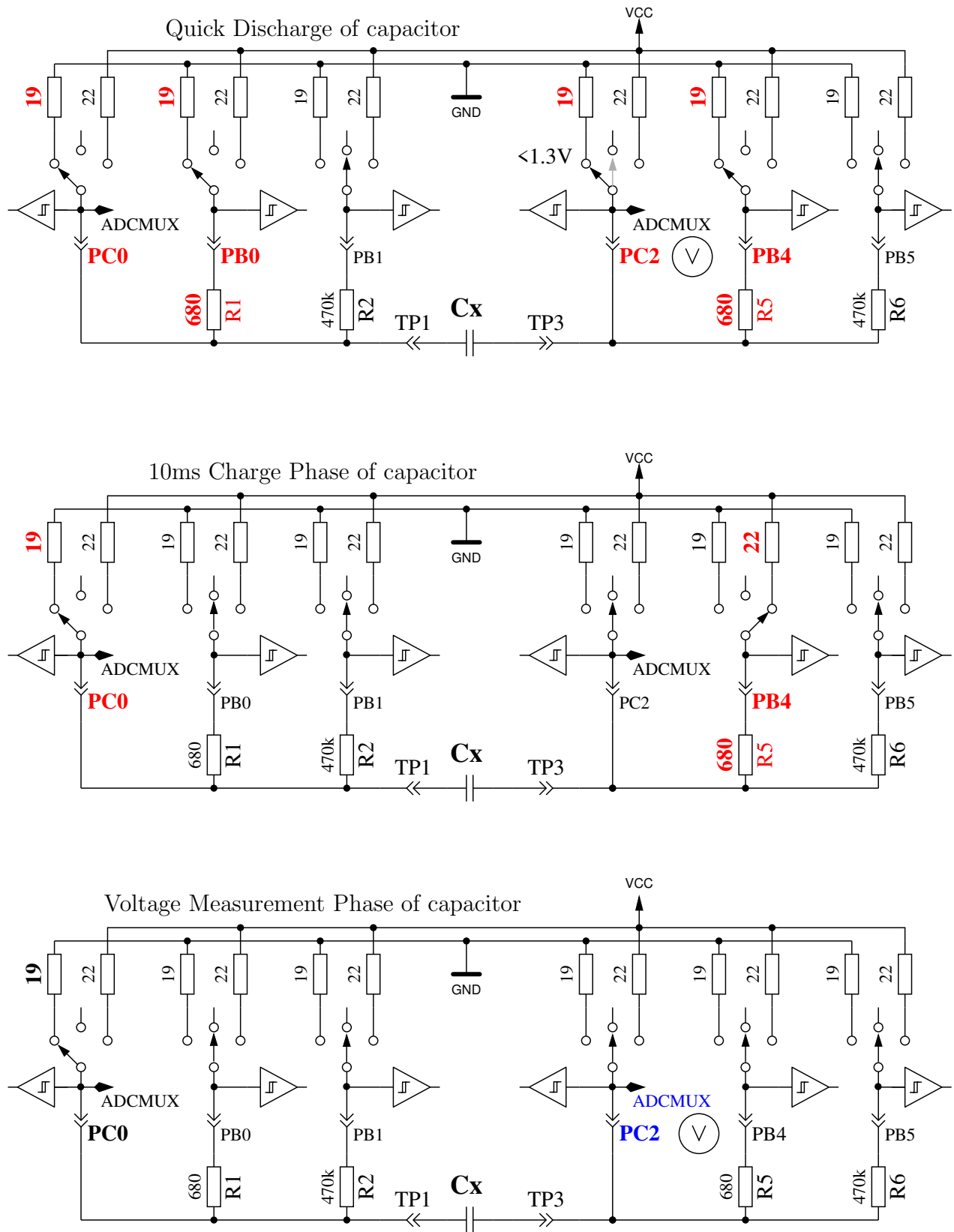


Figure 5.24. discharge a capacitor and load with 10ms load pulses until voltage reach a value of 300mV

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 don't disturb the measurement in most cases, because the flux voltage of most diodes is not reached. Beginning with software version 1.12k a

trickery is used to measure the residual voltage of a capacitor before the capacity measurement. Depending on the previous history of the capacitor the residual voltage can be positive or negative. Negative voltages can not be measured with the ADC. For that reason the voltage of the negative test pin is raised with the 680Ω resistor to about $132mV$ as shown in figure 5.25. With the difference of the voltages measured at both sides of the capacitor the residual voltage can be build with any polarity. The voltage of the positive test pin remains positive in any case, even if the capacitor have a negative residual voltage of some mV.

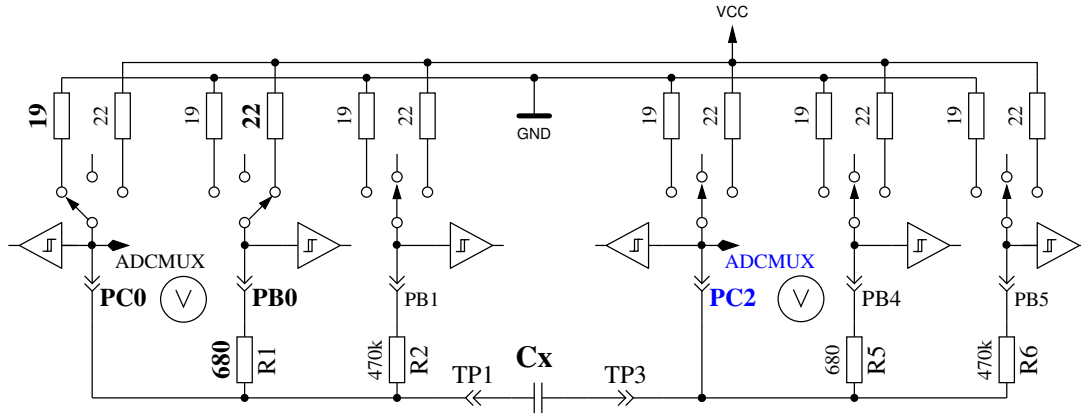
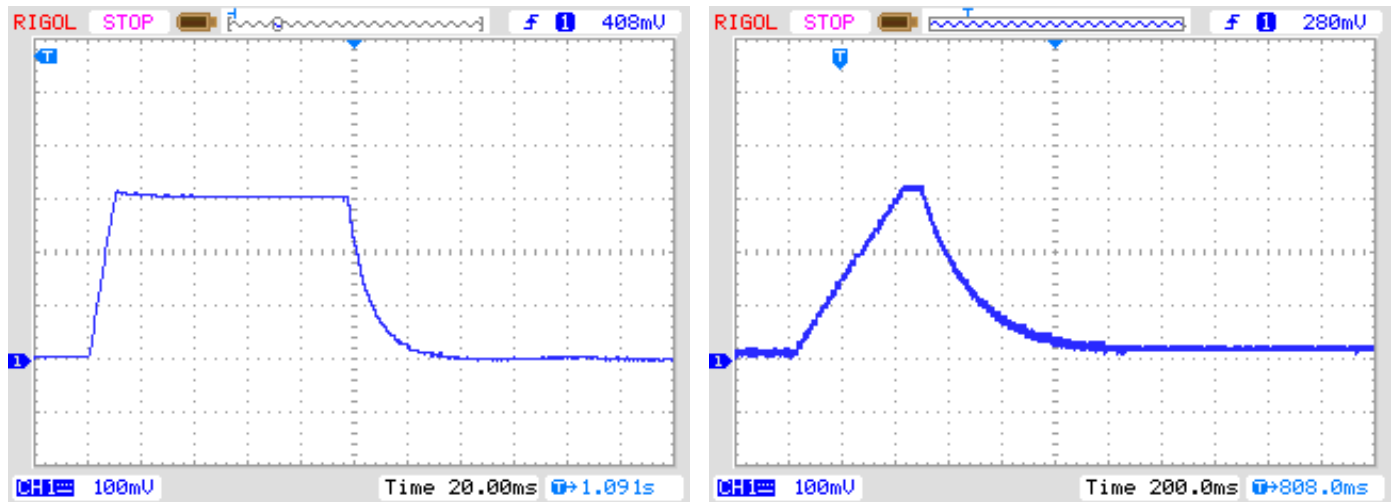


Figure 5.25. Measurement of the residual voltage of a capacitor before loading

Figure 5.26a 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 5.26b 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 5.27



(a) $229\mu F$ Capacitor

(b) $5mF$ Capacitor

Figure 5.26. Charge and discharge of big Capacitors for measuring

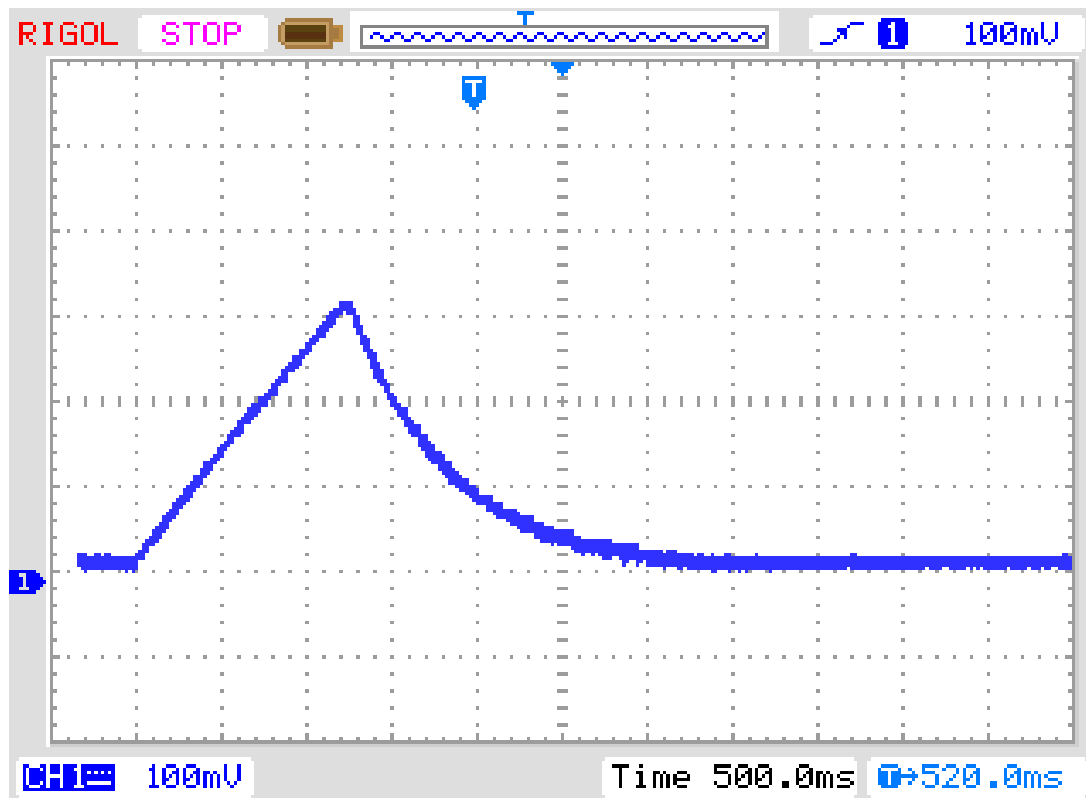


Figure 5.27. Charge and discharge of a $15mF$ Capacitor for measuring

After this capacity measurement the self-discharge of the capacitor will be checked by waiting a proportional period the loading has taken and reading the load voltage again. The measured capacity value is corrected due to this voltage drop. A test with a parallel connection of a $68\mu F$ capacitor and a $2.2k\Omega$ resistor shows the effectivity of this method. The measured capacity value without the resistor is $66.5\mu F$, with the parallel $2.2k\Omega$ resistor results to a capacity value of $66.3\mu F$. For comparison here are the results measured with a Peaktech 3315 multimeter: Without the resistor a capacity value of $68.2\mu F$ is measured, with the parallel $2.2k\Omega$ resistor a value of $192\mu F$ is measured with the multimeter.

5.3.3 Measurement of small Capacitors

If the first $10ms$ 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 5.28 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 we can get the capacity with a factor. 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`. From the build capacity value I subtract a predefined experimental find out constant or a value found by the last selftest with `AUTO_CAL` option to

eliminate the zero offset. The zero offset may vary with printed board type, the used test equipment or processor. The selftest with AUTO_CAL option will find out your zero offset automatically. I noticed that the reference voltage is permanently somewhat too low, so that you can choose an offset with the Makefile option REF_C_KORR. After calibration with the AUTO_CAL option, the REF_C_KORR will only be an offset to the measured difference voltage between loaded capacitor and internal reference. 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, ATmega168 and ATmega328 are applied as noted in the data sheets [4] [5]. A sample measurement of this type is shown in figure 5.29. The measurement time for the $22\mu F$ capacitor is above 2.6s because the $470k\Omega$ is used for charging. But discharging is in this case much faster than charging.

Figure 5.28. measurement little capacity values with comparator

Figure 5.29. 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 Measurement of very small capacity values with the sampling technique

The radio amateur Pieter-Tjerk (PA3FWM) has integrated the measurement capability for very small capacity values ($< 100\text{pF}$) with the sampling technique. The conversion period of the ADC is in fact too long for sampling a fast signal directly. But the voltage of the input signal is hold at a specified time of the conversion cycle, the Sample and Hold (SH) time. The ADC need 13 clock cycles for a total conversion and the ADC clock is build by dividing the processor clock by 128 or 64. The input voltage is fixed at exactly ADC clock number 1.5 for continuous cycles. If the input signal can be generated again and again, we can shift the sample time of the ADC from one to the next signal repetition, so that we get a sample sequence of the fast signal. A normal ADC cycle takes $13 \times 64 = 832$ clock cycles with a 8 MHz processor clock. If we repeat the input signal with a 831 clock cycle, a uninterrupted ADC (free run mode) would sample the signal one processor clock tick later for every following signal repetition. We must make sure with this method, that the first ADC sampling of the signal is done at the requested starting time. The time of the following ADC samples would be shifted by one processor clock tic later for every next signal repetition. If the signal can be repeated exactly, the combined signal of many periods is the same, which would be sampled and converted directly with a ADC running with the processor clock (8 MHz). Figure 5.30 shows the principle of sampling a ten times repeated signal to get 10 samples (SH0 - SH9). In reality the relative time shift of successive samples is much smaller than shown here.

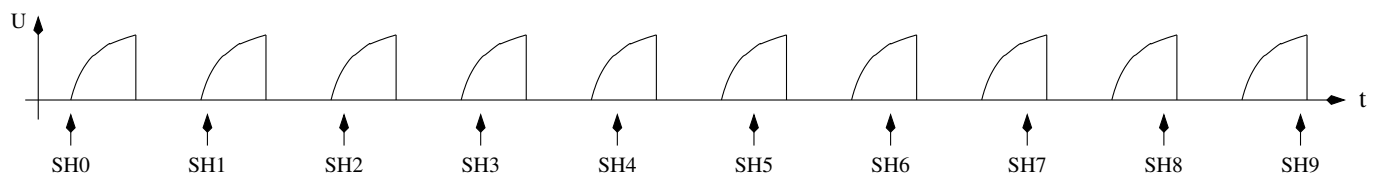


Figure 5.30. Scanning of a voltage curve with the sampling technique

One problem to fix the exactly time of the first sample is given by a continuous running clock precaler. Only a trigger of a external signal can reset the ADC clock divider. The clock divider would resume with dividing, if the ADC is started by program intruction. Only a program written in assembly language can specify the exactly points in time for this sampling technique. Every clock tick is important for building the program loops.

By analysing the voltage characteristic of charging a small capacitor you can see, that the time constant is not continuous during the sampling period. This was shown by Pieter-Tjerk at a presentation at the "60. UKW-Tagung in Weinheim". The internal capacitor of about 10 pF, which hold the input voltage for the conversion, is uncoupled at the SH time and connected again two ADC clock cycles later. Additionally there is a little jamming in the data just a half clock before the reconnection, which is probably caused by the switch of the multiplexor. Both disturbancies are respected with the data processing in the software. The sampling software can handle up to 255 samples. The software can also build the mean value of up to 32 charge sequencies. By building the mean value the effect of noise trouble will be less. The sampling software can monitor and process both, the charge or the discharge of a capacitor. Because both charge directions are used to measure the capacity value of a diode in reverse direction, the calibration task measure the zero

capacity value in both charge direction for all pin combinations. By measuring the capacity value of a diode in both charge directions, the difference between both values can be shown. With the charge direction a capacity value is measured near the voltage 0 and with the discharge direction the capacity is measured near a voltage of 5V. With normal capacitors there is no difference in the capacity values with this little voltage difference detectable. Therefore only the charge direction is used for measuring little capacities ($<100\text{pF}$).

Pieter-Tjerk has optimized his function for a 16 MHz operation. In this configuration you will get a resolution of 0.01 pF. For the 8 MHz operation the ADC will run at the half speed for getting the above-mentioned disturbances at the same data points compared to the 16-MHz operation. The loss of resolution with the 8 MHz operation will be irrelevant for most users and the additional testing time with the slower ADC in this mode is tolerable too.

5.3.5 Measurement of the Equivalent Series Resistance ESR

The series resistance ESR [10] is a good indicator for the aging of electrolytical capacitors for example. The figure 5.31 shows a equivalent circuit of a capacitor. The resistor R_p represents the leakage resistance of the capacitor, ESL the equivalent series inductivity and the resistance ESR represents the equivalent series resistance.

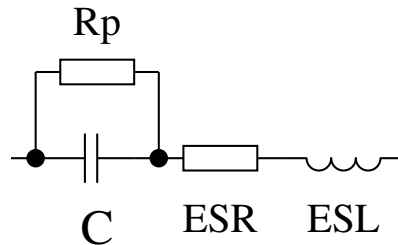


Figure 5.31. Equivalent circuit of a capacitor

Usually the data sheets publish ESR values, which are measured with a frequency of 100 kHz and a temperature of 20°C . The figures 5.32 and 5.33 shows the ESR values of the Panasonic series FC and the “low ESR” series FR. Both series are able to operate up to a temperature of 105°C . The figure 5.34 shows the data of both series with a allowable working stress of 25V. If the series have different types with the same capacity and voltage range, the one with the lowest ESR is taken for the diagram. The values of capacity and ESR of electrolytic capacitors change significant with there operating temperature.

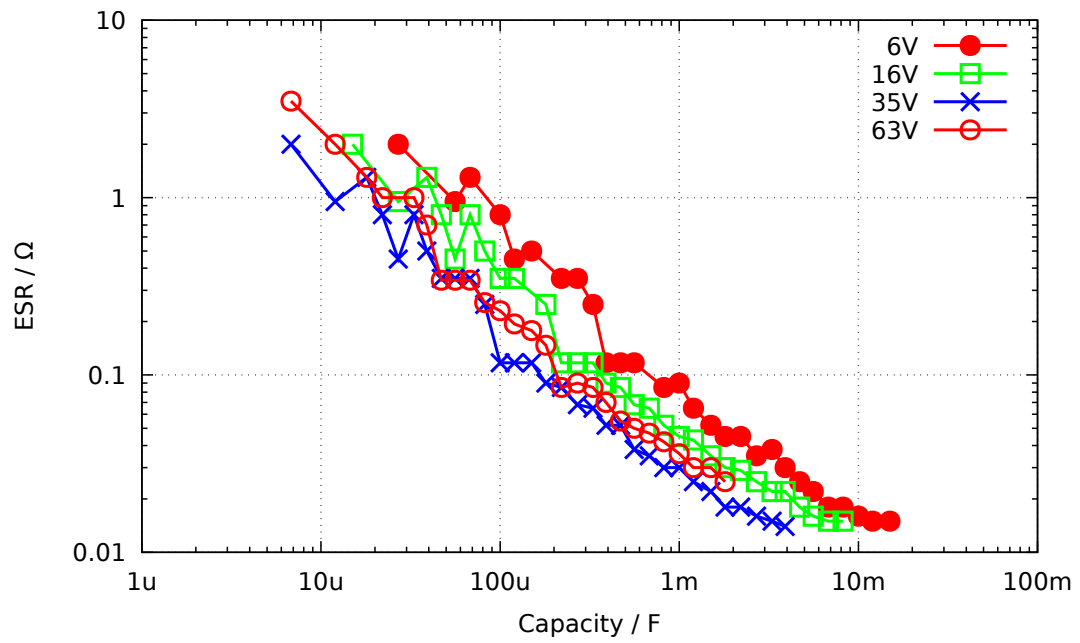


Figure 5.32. ESR data from the Panasonic data sheet of the series FC

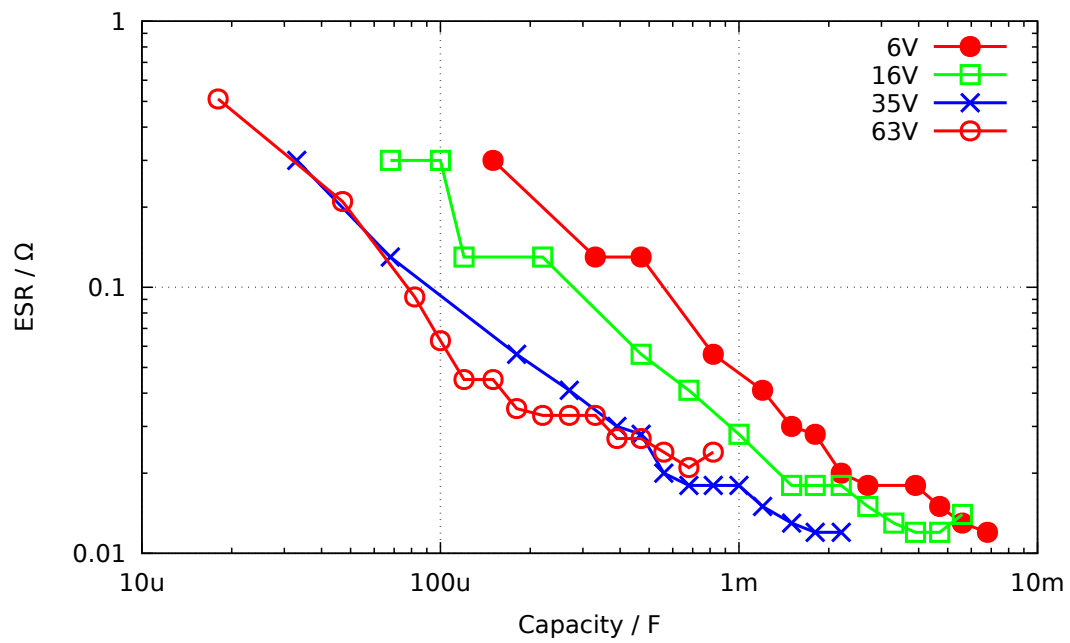


Figure 5.33. ESR data from the Panasonic data sheet of the series FR

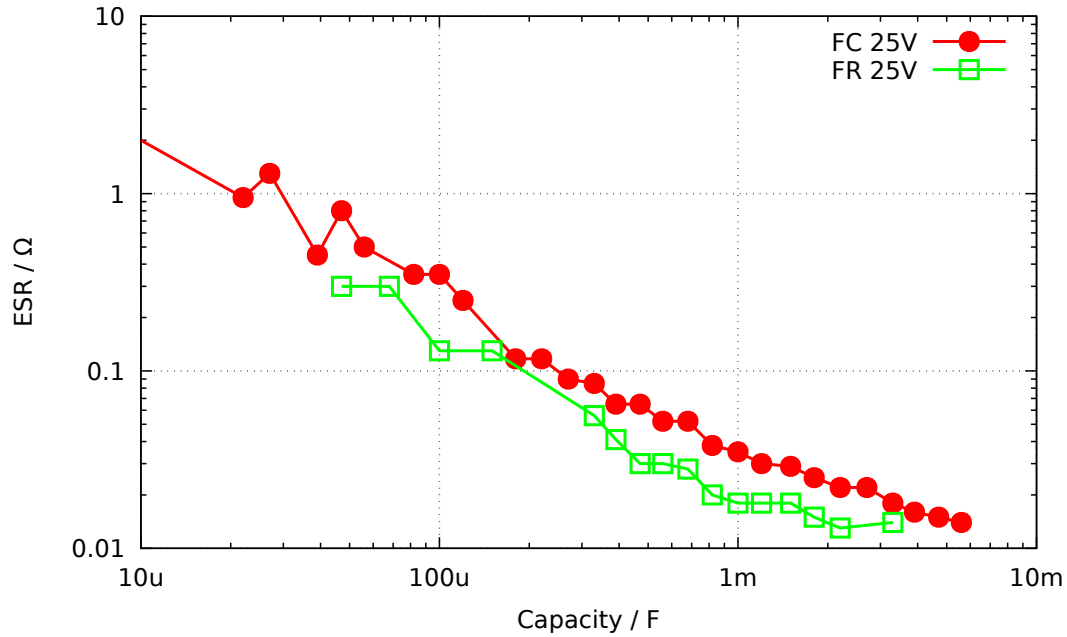


Figure 5.34. Comparison of the ESR data from series FC with series FR

There is no simple way to measure the ESR with a frequency of 100 kHz with the ATmega hardware, because neither the ADC can sample a so high input frequency, nor the existing circuit can support with a $100kHz$ signal. At next there will be introduced two method's for the measurement of the ESR, which both manage on the existing circuit. Both method's use a rectangular signal for the measurement, so that the results will never be the same with the values measured with sinusoidal signal. With the first method the measured values are close to those values, which are measured with a $1kHz$ signal. But the second method has the advantage, that the zero value can be determined with shorted test pads and that additionally the measured ESR is more close to the value measured with $10kHz$ signal. Currently I have no idea for a measurement method, which can produce a ESR value close to the value of a $100kHz$ measurement.

The following table 5.6 should show the dependency of the ESR results from measurement frequency. All capacitors without the $47\mu F$ capacitor are from the same FC series of manufactor Panasonic. The reference values are measured with a Peaktech 2170 LCR meter. All results of the TransistorTester are measured with the method 2 of subchapter 5.3.7 . Capacitors with big capacity values are difficult to measure with higher frequencies like $100kHz$ because the inductance ESL make trouble.

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

Table 5.6. ESR values of different electrolytical capacitors

5.3.6 Measurement of the Equivalent Series Resistance ESR, first way

If the measured capacitor has a capacity of more than $0.45\mu F$, the tester will try to measure the series resistance too. For a capacity of more than $3.6\mu F$ the normal clock rate of $125kHz$ for the Analog-Digital converter is used. For lower capacities the higher clock rate of $500kHz$ is used to accelerate the measurement. The accuracy of the ADC results will be more worth by the higher clock rate, but this could be accepted by the higher ESR values of capacitors with lower capacity. Otherwise the measurement of ESR with this method is not possible for a capacity of less than $1.8\mu F$ at the normal clock rate of $125kHz$.

Strictly speaking the ESR of a capacitor depends on the operating frequency and temperature. Usually the value measured with sine wave-form signal of $100kHz$ is denoted in the data sheets. This measurement can not be done with the ATmega without external equipment. With the subsequent written method the measurement frequency with the standard ADC clock rate will be below 640 Hz with nearly rectangular signal. With $500kHz$ ADC clock rate the measurement frequency will be 2400 Hz . To get the value of the equivalent series resistance, the voltage of both connections will be measured during loading in one direction with the ADC internal reference voltage ($1.1V$). After the measurement the load current will be switched off and the voltage of the capacitor is measured again without the current. If this voltage is below $3mV$, the sequence of measurement is repeated. The figure 5.35 shows the corresponding circuits.

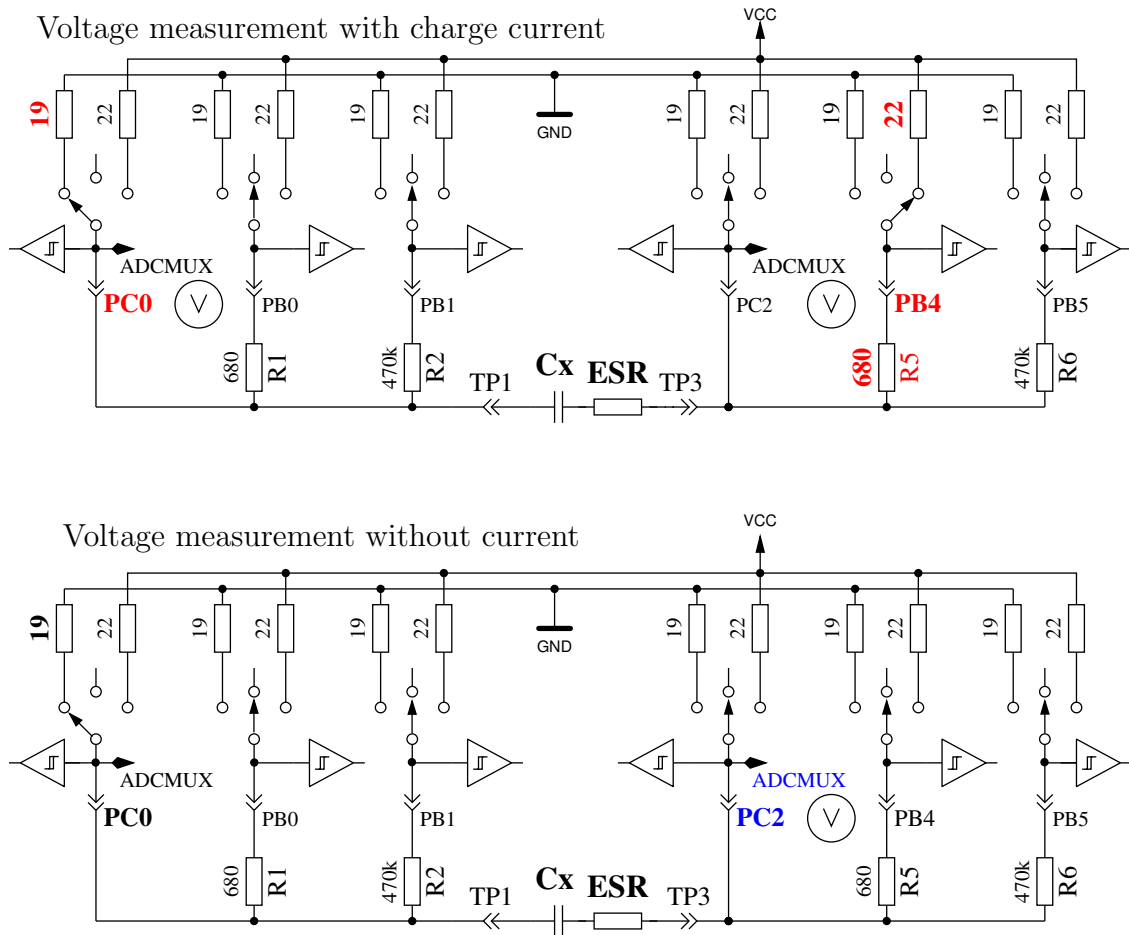
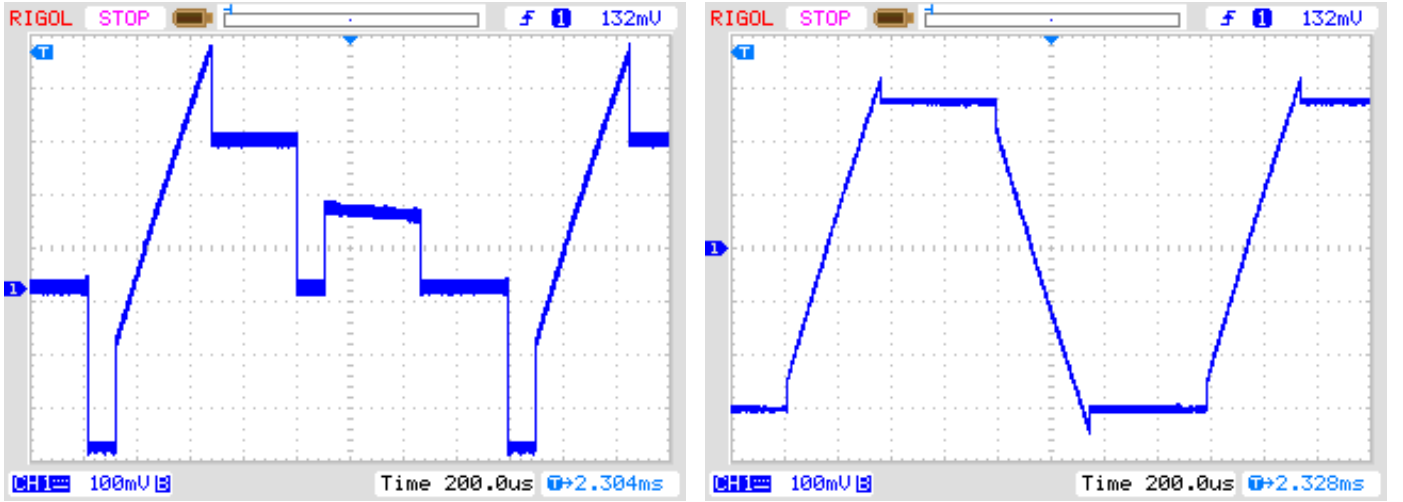


Figure 5.35. Circuit of the ESR measurements of a capacitor

The difference of capacitor voltages with and without current is proportional to the internal resistance of the capacitor. The expected voltage of this difference is so low, that one measurement can not result to a feasible result. Therefore after this the current will be switched to the opposite direction

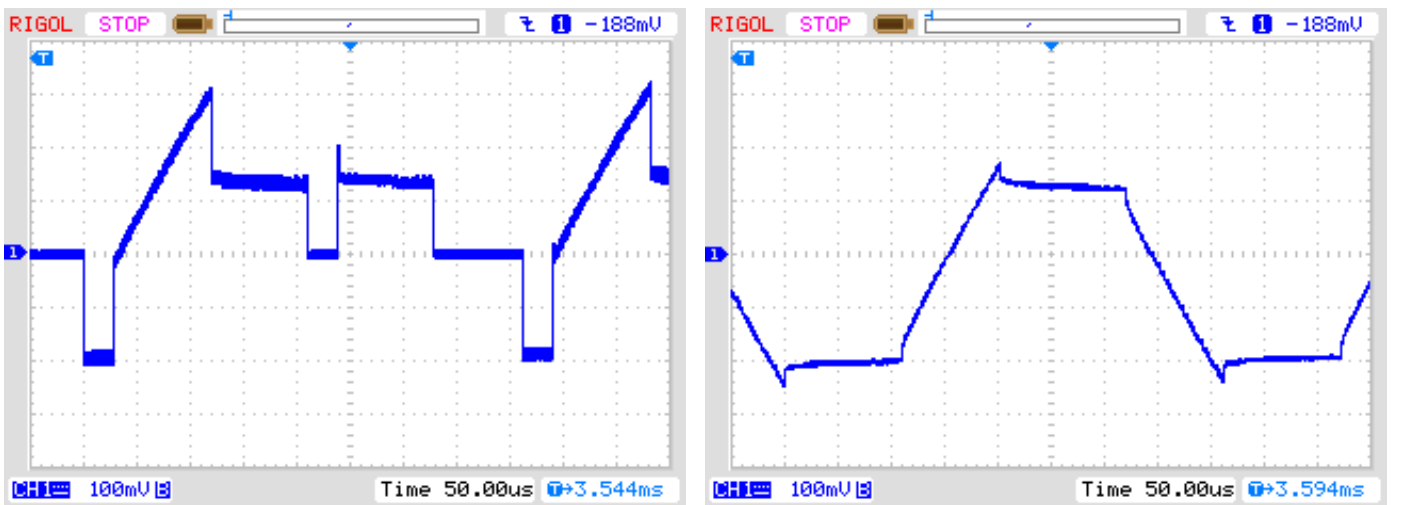
and the same measurement will be repeated. The whole measurement sequence will be done 128 times and the results of the voltage measurements will be added. So we have three sums of voltages, the voltage U_{lp} at the low side of the capacitor with current, the voltage U_{hp} at the high side of the capacitor with current and the voltage U_c of the high side of the capacitor without current. The sum of voltages at the low side of the capacitor represents the potential drop with the mean load current at the port output resistance R_{port} . The voltage difference of the high side and the low side of the capacitor represents the voltage of the capacitor with load current $U_{diff} = U_{hp} - U_{lp}$. The difference $U_{esr} = U_{diff} - U_c$ should represent the voltage drop at the internal resistance of the capacitor with mean load current. We will get the resistance value with the relation of this voltage U_{esr} to the voltage U_{lp} , scaled with the known resistance value of the port output R_{port} . The scale factor is selected to get a resistance resolution of 0.01Ω : $Resr = \frac{U_{esr} \cdot 10 \cdot R_{port}}{U_{lp}}$ The figure 5.36 shows a part of the voltage curve of a $4.2\mu F$ capacitor during the ESR measurement. To explain the influence of the ESR, a series 6.8Ω resistor is added to the capacitor. The little voltage break after loading the capacitor is interpreted by software to get the ESR. The greater voltage drop of the measurement to GND potential is caused by the port output resistance of about 20Ω . For this measurement a total ESR of 7.5Ω is reported by the tester, without the series 6.8Ω resistor a ESR of 0.56Ω is found. The figure 5.37 shows the same measurement with higher measurement frequency of a $2.2\mu F$ electrolytical capacitor with a ESR of 6.5Ω .



(a) measured one pin to GND

(b) measured pin to pin

Figure 5.36. Voltage curve of a $4.2\mu F$ capacitor during the ESR measurement



(a) measured one pin to GND

(b) measured pin to pin

Figure 5.37. Voltage curve of a $2.2\mu F$ capacitor during the ESR measurement

The accuracy of the ESR measurement is not very high by different reasons:

1. The voltage measurement at both pins of the capacitor can not be done at the same time, the only way is to do it in sequence. In the interim time between both measurements the load current has changed due to the charge of capacitor. The program tries to compensate this fact with a capacity dependent correction of the low side voltage.
2. The ADC takes the measurement voltage after 1.5 clock ticks after the start of conversion. The conversion begins with the rising edge of the ADC-clock, if the start bit is set. If the charge current will be switched off to early, the ADC takes the wrong voltage for the measurement with current. If the charge current will be switched off to late, the capacitor will take more electric charge, than that of the corresponding measurement with load current. This will cause a too high voltage of the measurement without current. But it is difficult to switch off the current at the right time by software.
3. The port output resistance is used as a reference value by this measurement method, but this resistance value is not exactly known too.
4. The resolution of the ADC is not sufficient to get a resolution of resistance of 0.01Ω . To get the best available resolution of ADC, the internal reference ($1.1V$) is used for all measurements. The resolution deficit will be attenuated by accumulating a big number of single measurements too.
5. The switching of ports can not be exactly synchronized to the ADC clock with polling of conversion done.

Anyway the results seems to be practical, as shown with the following figure 5.38. The ESR values of the same part measured with the Transistortester vary more than the values measured with the LCR meter. The ESR values from the LCR meter are measured with a frequency of $1kHz$ or are interpolated for little capacities to $2.4kHz$. You must respect the quality of all connection parts. The used cable connections can cause a higher measured resistance value. The plug connectors can also result a higher resistance value. The LCR meter has the advantage of the used Kelvin terminals. Only one capacitor with a capacity below $1\mu F$ was a $500nF$ ceramic type, all others were plastic film capacitors. The only electrolytical capacitor of the test series below $9\mu F$ was a $2.2\mu F$ capacitor.

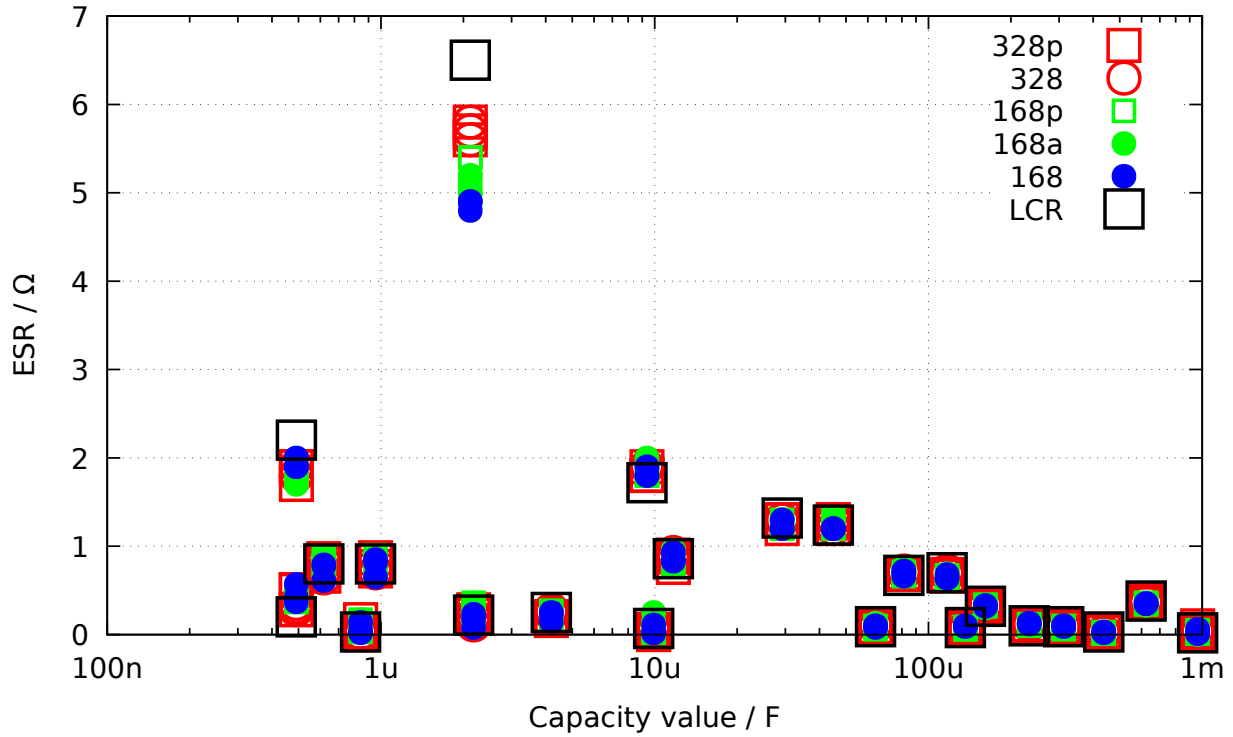


Figure 5.38. ESR measurement results of 15 different ATmega

5.3.7 Measurement of the Equivalent Series Resistance ESR, second way

From beginning with software version 1.07k the ESR measurement way is changed to a new measurement method. The different measurement steps are shown in figure 5.40. The difference to the previous way is that the period of current flow through the capacitor is essential shorter. The capacitor is preloaded with a half pulse to the negative direction and is then loaded in a cyclic way in both direction. The timing of the load pulse is so selected, that the middle of the load pulse at sample 4 and 8 is pointed to the sample and hold time of the ADC (2.5 clock ticks after start of ADC). A complete measurement cycle is shown in figure 5.39. The sums of 255 measurement cycle results is used for getting a result with adequate resolution. A continuing charge of the capacitor in any direction is avoided by the same charge and discharge pulse length and the same circuit. By measuring the reference voltage the capacitor remains currentless. By that this measurement are not time critical. It is only assumed, that the capacitor hold the voltage until the next charge or discharge pulse begins.

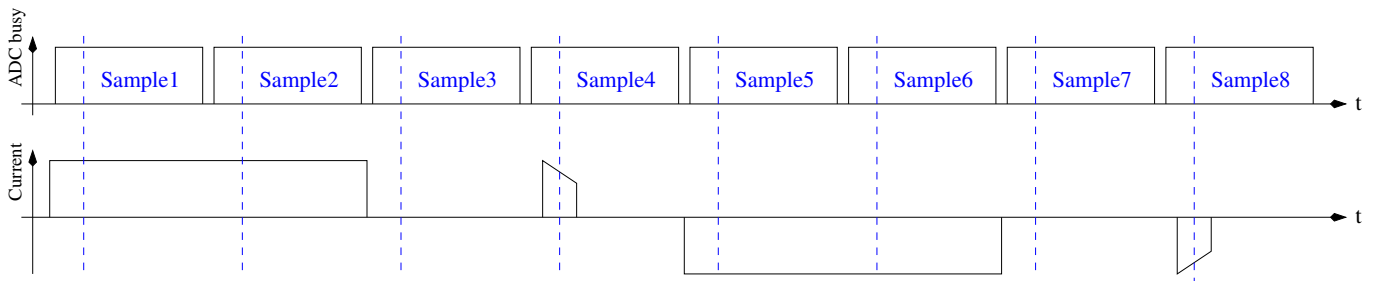


Figure 5.39. Timing of a measurement cycle for the new ESR-measurement way

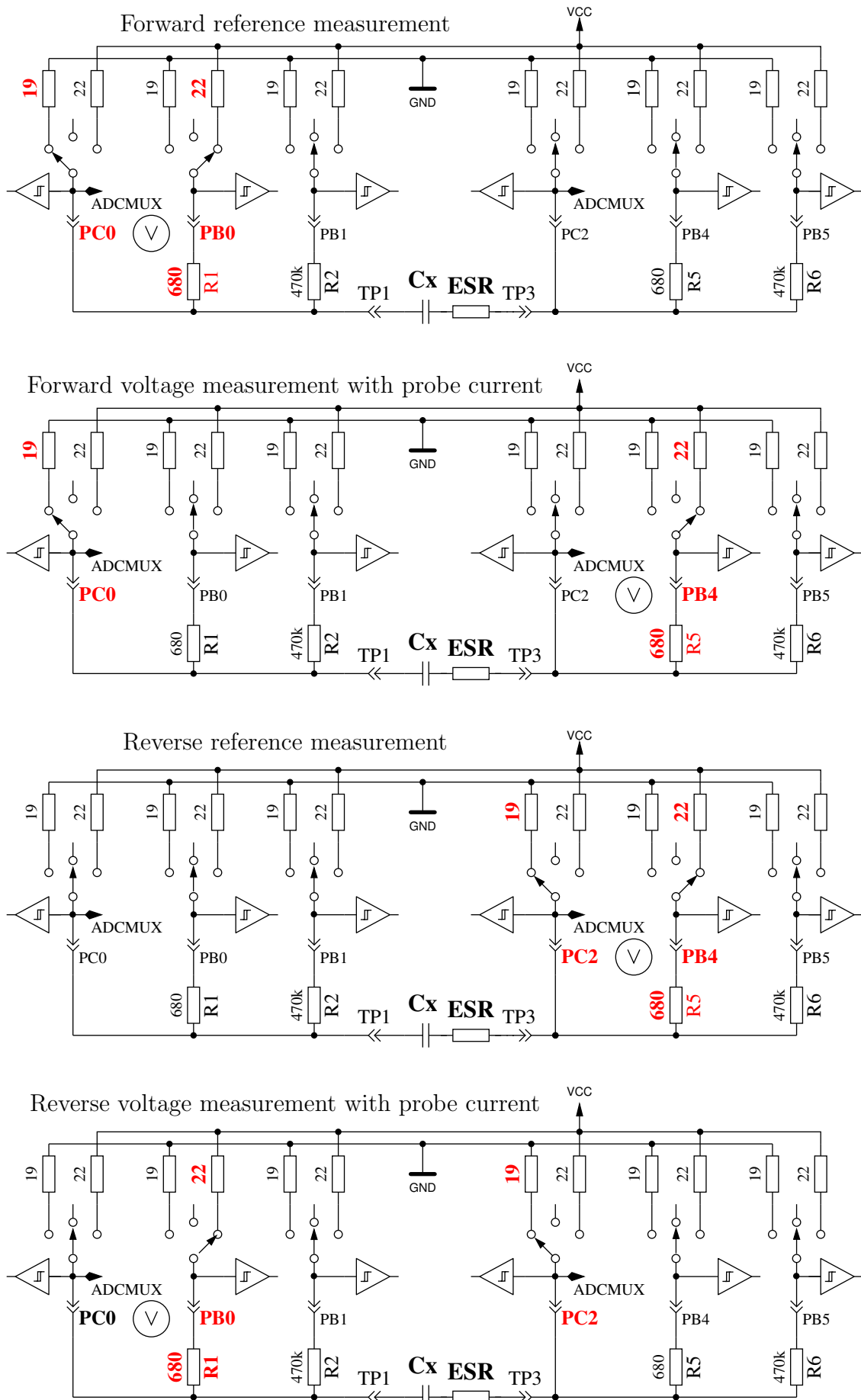
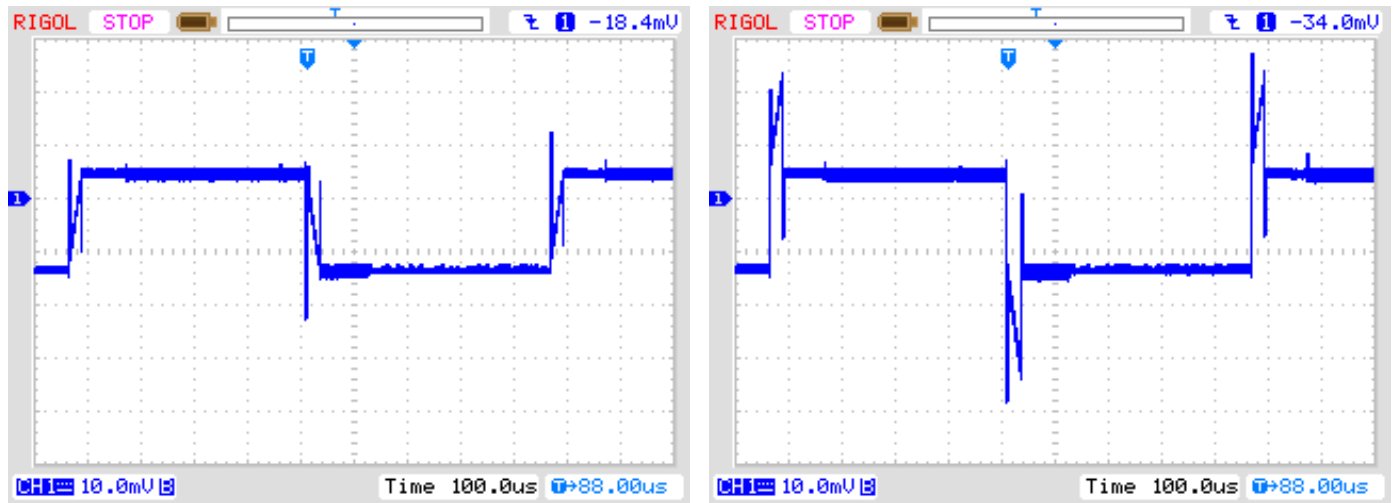


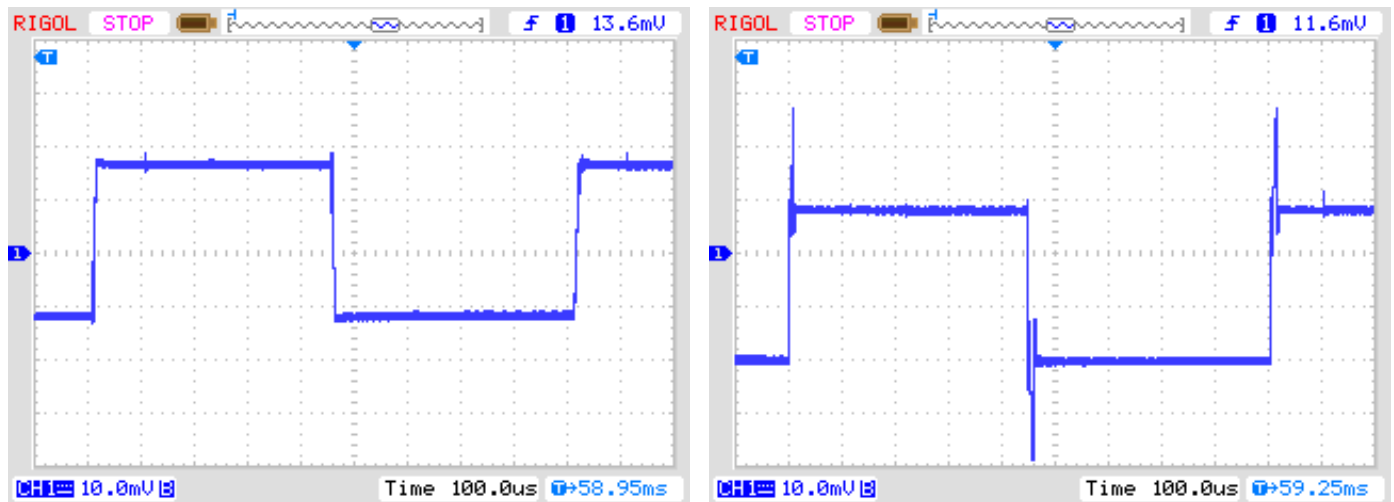
Figure 5.40. More simple ESR measurement of a capacitor

Due to the shorter load puls not only the ESR of capacitors with lower capacity can be measured, but this way of measurement can also be used for the measurement of resistors with little resistance, if they don't have a detectable inductance. By doing that, a resolution of 0.01Ω for this resistors can be achieved. Also the zero resistance can be detected by the calibration part of the selftest for all three test pin combination. You should keep in mind, that stable plug sockets or clamping connectors are essential for stable results. The measurement periode is about $900\mu s$, which results to a frequency of about $1.1kHz$. Because the load pulse is very short, the measurement result is comparable to measurements with $10kHz$. A measurement example with a $10\mu F$ foil capacitor, once measured alone and once measures with a 2.7Ω series resistor is shown in figure 5.41. You can see the effect of the additional resistance by comparing both diagrams. You can see also, why the ADC measurement (SH) should point to the middle of the load pulse. With big capacity values the load current is nearly stable during the total pulse length, so you will get the middle voltage at the middle time of the load pulse. With lower capacity values you will get a significant difference, which can be compensated by the known capacity value.



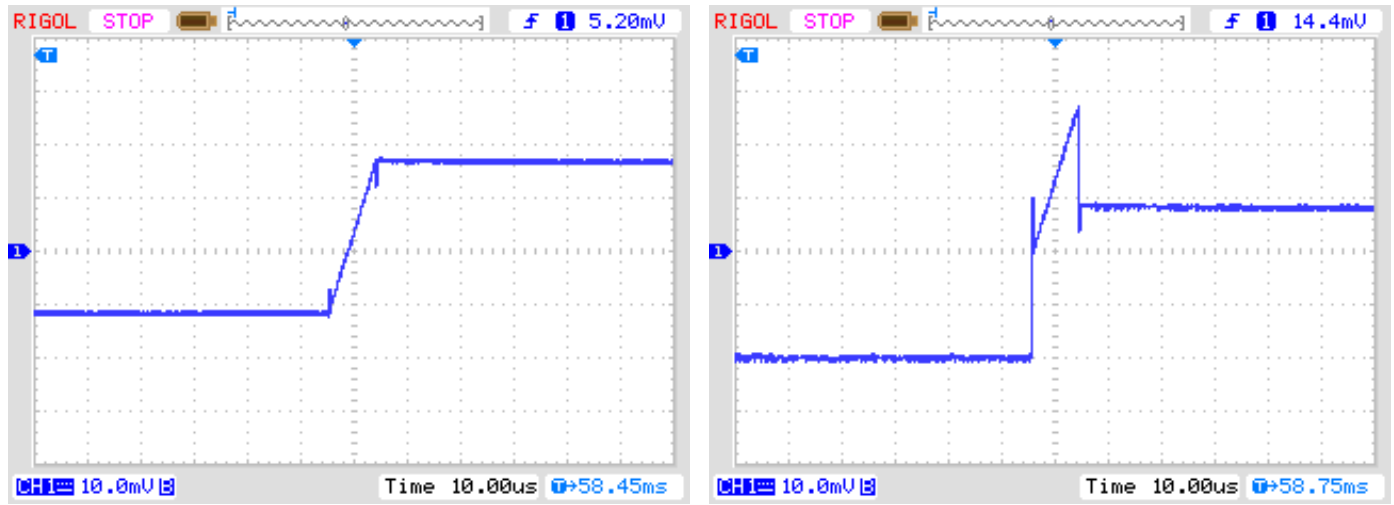
(a) without series resistance (b) with 2.7Ω series resistance
Figure 5.41. Voltage curve of a $10\mu F$ capacitor during new ESR measurement

By using the $27\mu s$ long charge pulses the ESR of capacitors above $180nF$ can be determined. For measuring of capacitors with lower capacity the current pulse is shortened to $8\mu s$ for version 1.11k. The figures 5.42 show the voltage curve of a $2.2\mu F$ capacitor without and with a 2.7Ω series resistor.



(a) without series resistance (b) with 2.7Ω series resistance
Figure 5.42. Voltage curve of a $2.2\mu F$ capacitor during new ESR measurement with $8\mu s$ charge pulses

Because you can not see the sample and hold time of the ADC in the figures 5.42, the voltage curve is shown zoomed in figures 5.43. The sample and hold time is approximately in the middle of the screen picture.



(a) without series resistance

(b) with 2.7Ω series resistance

Figure 5.43. Zoomed voltage curve of a $2.2\mu F$ capacitor during new ESR measurement with $8\mu s$ charge pulses

The measurement results of the new ESR measurement method is shown in figure 5.44. The ESR values are different from the results shown for the previous measurement procedure in figure 5.38 because the ESR is frequency dependence of the ESR. The reference values are determined with a LCR meter at a measurement frequency of $10kHz$.

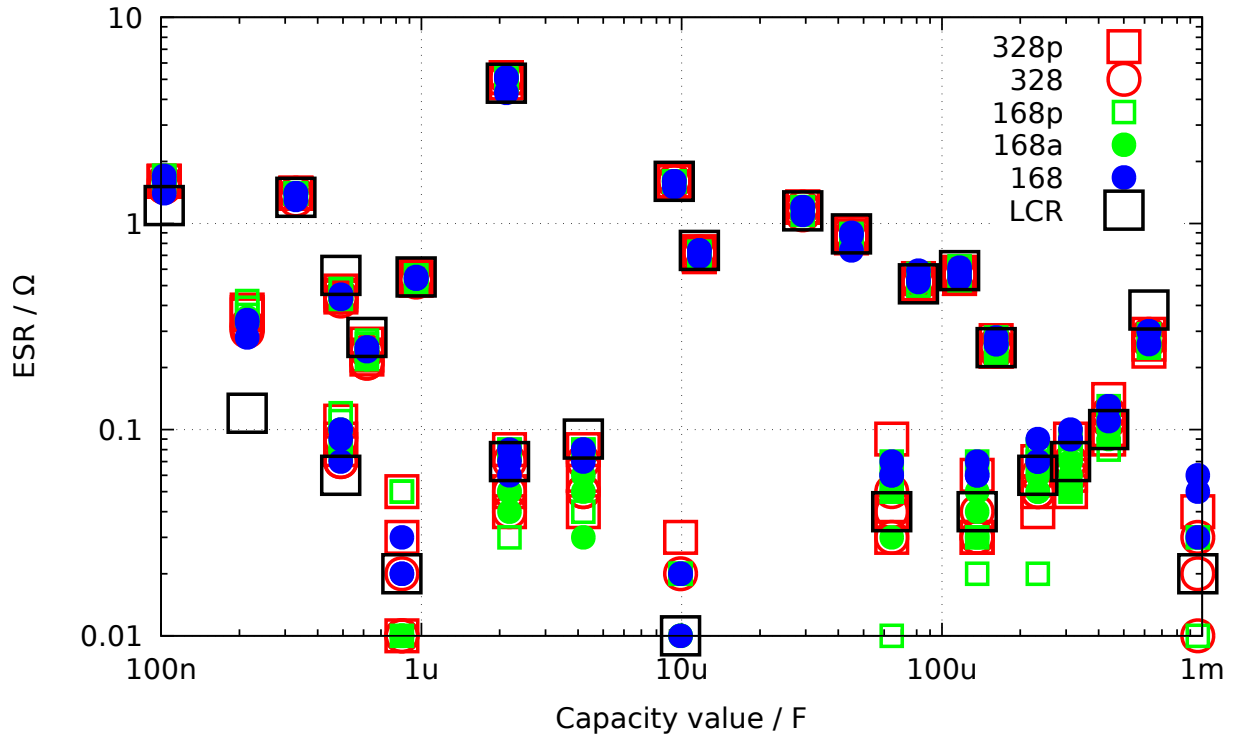


Figure 5.44. ESR results with 15 different ATmega, method 2

A measurement series with different sized electrolytic capacitors are shown in figure 5.45. The results of a PeakTech 3315 LCR meter of measurements with different frequencies and the results of the TransistorTester are shown together. The resistance is illustrated with logarithmic scale in

this diagram. In all cases the results of the TransistorTester is near by the results of the $10kHz$ measurements of the LCR meter. Only the $500\mu F/3V$ capacitor is a older exemplar, all others capacitors are as good as new.

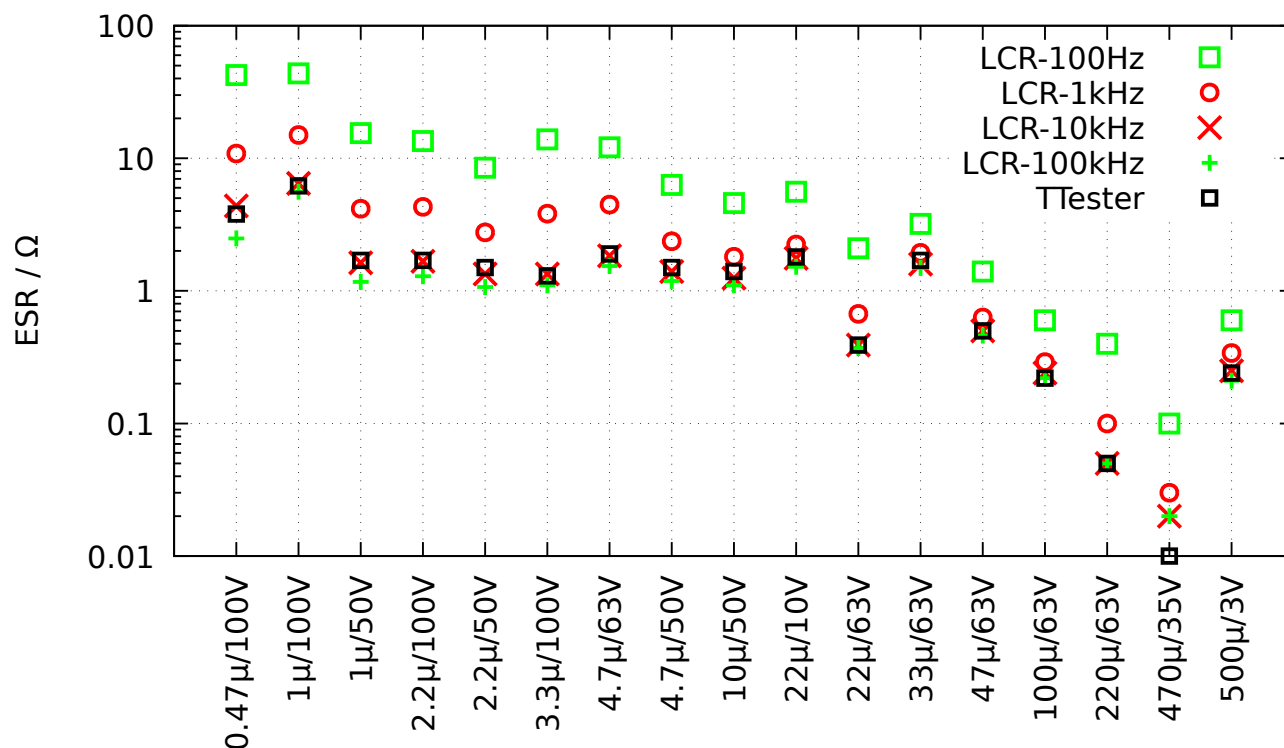


Figure 5.45. Results of the ESR measurements of different electrolytic capacitors

Because the new measurement method can be taken for measuring of resistors with low values, the measurement errors of some resistors below 10Ω with three example of each ATmega type will be shown in figure 5.46.

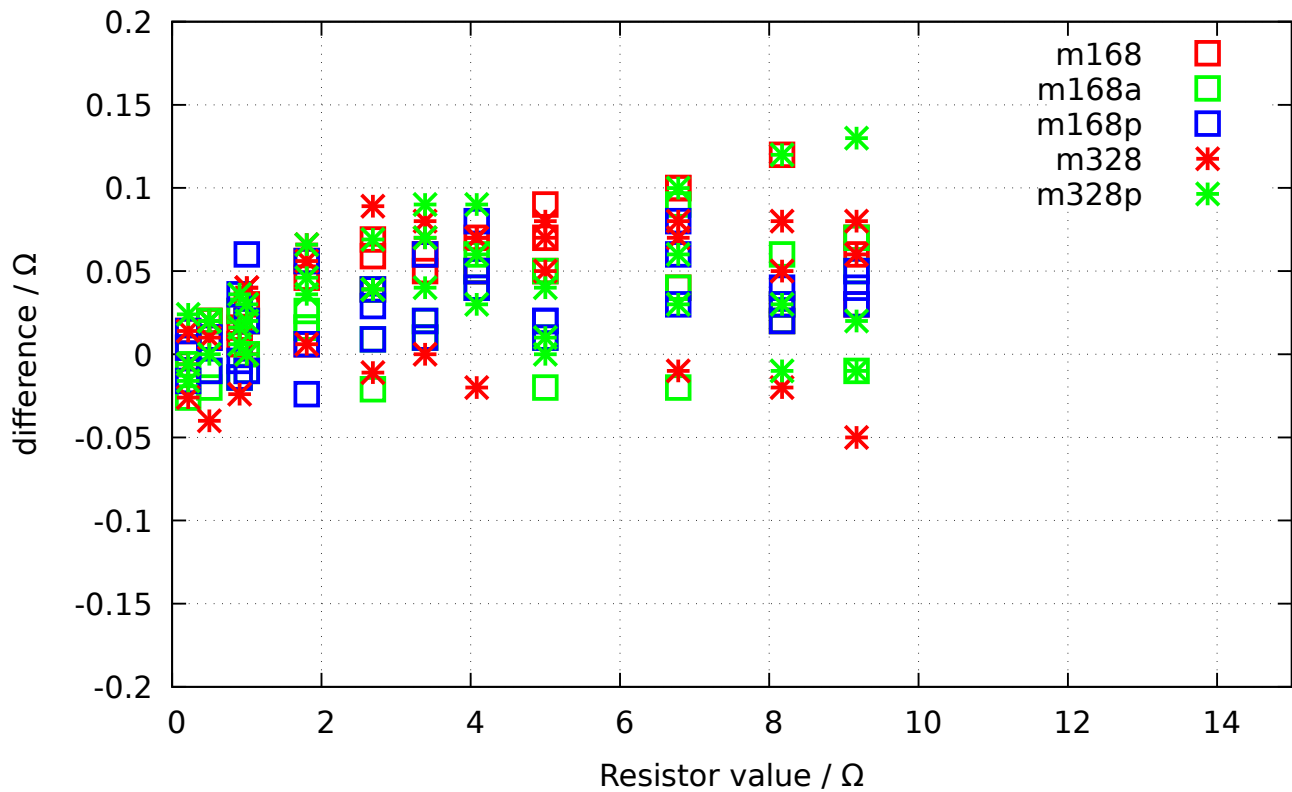
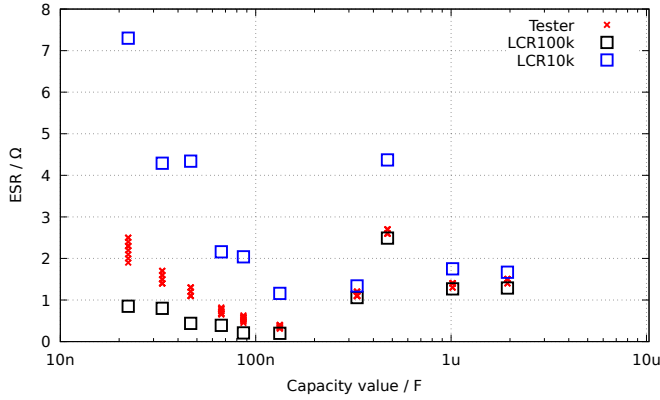
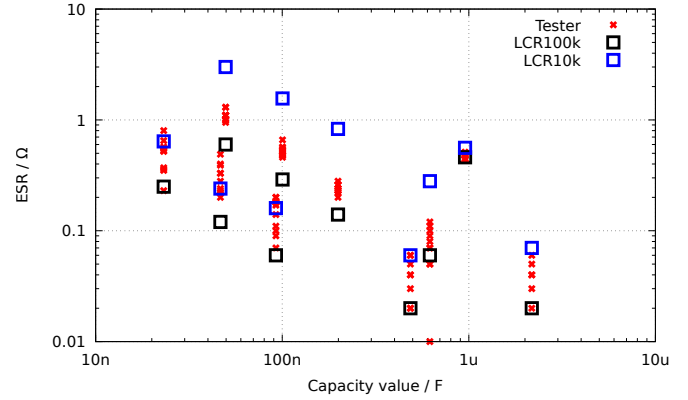


Figure 5.46. Measurement errors of resistors with the ESR method

With software version 1.12k the load pulse length for the capacitors is reduced to $2\mu s$ to accomplish ESR measurements of capacitors with lower capacity values. Now a ESR-value can be measured for capacity values above $20nF$. But the measurement error will grow for lower capacity values. The reason for that is the decrease of the time constant of the RC-circuit, which will be only about $14.4\mu s$ for a capacity value of $20nF$. This will result to a fast change of the capacitor voltage during the $2\mu s$ current pulses. The software can select the sampling instance of the ADC only to the time of any processor clock. But the input filter of the ADC has a time constant of about $0.24\mu s$, which can vary from exemplar to exemplar of the ATmega. This change of the ADC input filter time constant can not be respected with the software. For this purpose the ADC sample time must be selectable to a fraction of the processor clock period. With greater capacity values of the measurement object the time constant will grow and the voltage change during the load pulse will decrease. For this reason the variation of the ADC input filter time constant has lower effect with greater capacity values. The examples in figures 5.47 show the results for some capacitors with 10 different tester examples. The picture at the left side shows the ESR results of some capacitors with higher ESR values. The result is rather similar compared with the results of the Peaktech 2170 LCR meter at $10kHz$ and at $100kHz$. At the right picture you can see the measurement results of some high quality capacitors with low ESR values. Although you can notice the limit of the method especially for low capacity values, the result is still better than no information. In either case you can classify the quality of a capacitor also for lower capacity values.



(a) higher ESR values



(b) lower ESR values

Figure 5.47. ESR-measurements of capacitors with lower capacity

For processors with more than 16K flash memory the $470k\Omega$ resistor is connected parallel to the 680Ω resistor for the half of single measurements beginning with software release 1.12k to vary the measurement current a little bit. Unfortunately the additional current is very low, so that the resulting voltage enhancement can not make shure a change of the ADC result. The voltage enhancement is only about 20% of a ADC bit with the internal 1.1V reference. A add of a little noise voltage at the ADC input would be preferable. With that feature a statistical enhancement of the ADC resolution by averaging can be achieved.

5.3.8 Voltage loss after a load pulse, Vloss

With the measurement of capacitors with big capacity values the voltage loss after the loading is analysed. The reached load voltage is lost with electrolytic capacitors after a short periode. This voltage loss can be caused by a parallel connected resistor. But I assume, that this voltage loss of electrolytic capacitors is caused by a internal load dispersion directly after the load pulse. By loading the capacitors with the $470k\Omega$ resistor, as it is done for little capacity values, this dispersion is already done after switching off the current. No voltage loss is detectable for this case. But if you load the same capacitor with a short current pulse, you can also detect the voltage loss for capacitors with lower capacity. The same effect with lower loss can also be noticed for ceramic type capacitors. I have noticed, that capacitors with more than some % voltage loss are suspect. Especially noticable with respect to the voltage loss are older paper type capacitors, which are for other measurement a problem too. Some measurement examples will be shown in the following table.

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

In this table you will find, that the capacity of all foil type capacitors can be measured by all instruments with good precision. The capacity values and the quality factor Q of the PeakTech LCR meter are minimum and maximum values of the measurements in the frequency range $100Hz$ to $100kHz$. At all examples in the table the voltage loss Vloss of the TransistorTester is big, if the capacitors have a low quality factor. Only in this case the differences of the capacity measurement results are also big. The TransistorTester can only determine the voltage loss, if the measured capacity is more than $5000pF$.

5.3.9 Separate capacity and ESR measurement

The separate capacity measurement and the afterwards measured ESR is only available for ATmega with sufficient memory with the handling dialog. This way of measurement is usefull for measurement of capacitors in the circuit without desoldering. Please take care, that all capacitors of the printed board are discharched before starting any measurement! To realize the measurement in the soldered state, the measurement voltage is hold to a low level of a little above $300mV$ only. In addition to that the measurement is only done with the 680Ω resistor to prevent a big effect of connected components on the printed board. To enable the measurement of capacitors with little capacity value, the first load puls is only $200\mu s$ short. If the loaded voltage let expect, that the $300mV$ would not be reached with a load pulse of $2ms$, the next load pulse is done with $2ms$ length. When the capacity value of the measured capacitor is very high, the voltage grow is still low with the $2ms$ pulse. In this case the next load pulse(s) will be done with $20ms$ length. If the loaded Voltage grow near to $300mV$, the shorter load pulses will be used again. The total time of load pulses is added and after the load voltage has passed over $300mV$, the capacity value is computed from load time and the loaded voltage. With this method capacity values of a little below $2\mu F$ can be measured. The upper limit for the capacity values is given with the restricted load time of $2.5s$ to about $50mF$. If the capacity value is successfully measured, the ESR value of the capacitor is measured with the method already described in section 5.3.7. The result is shown only short and then the next measurement is started immediately. The series of mesurement is stopped after 250 measurements or after pressing the start key. After finishing the measurements the program returns to the handling dialog.

5.3.10 Results of Capacitor measurement

The results of my capacity measurements are shown in figure 5.48 for three ATmega8 processors. Additionally some values of original software are shown with a correction factor of 0.88 (-12%). Other measurement results of different ATmega8 versions are shown in figure 5.49a and 5.49b. The results of the measurement of the same capacitors for a ATmega168 is shown in figure 5.50. 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.

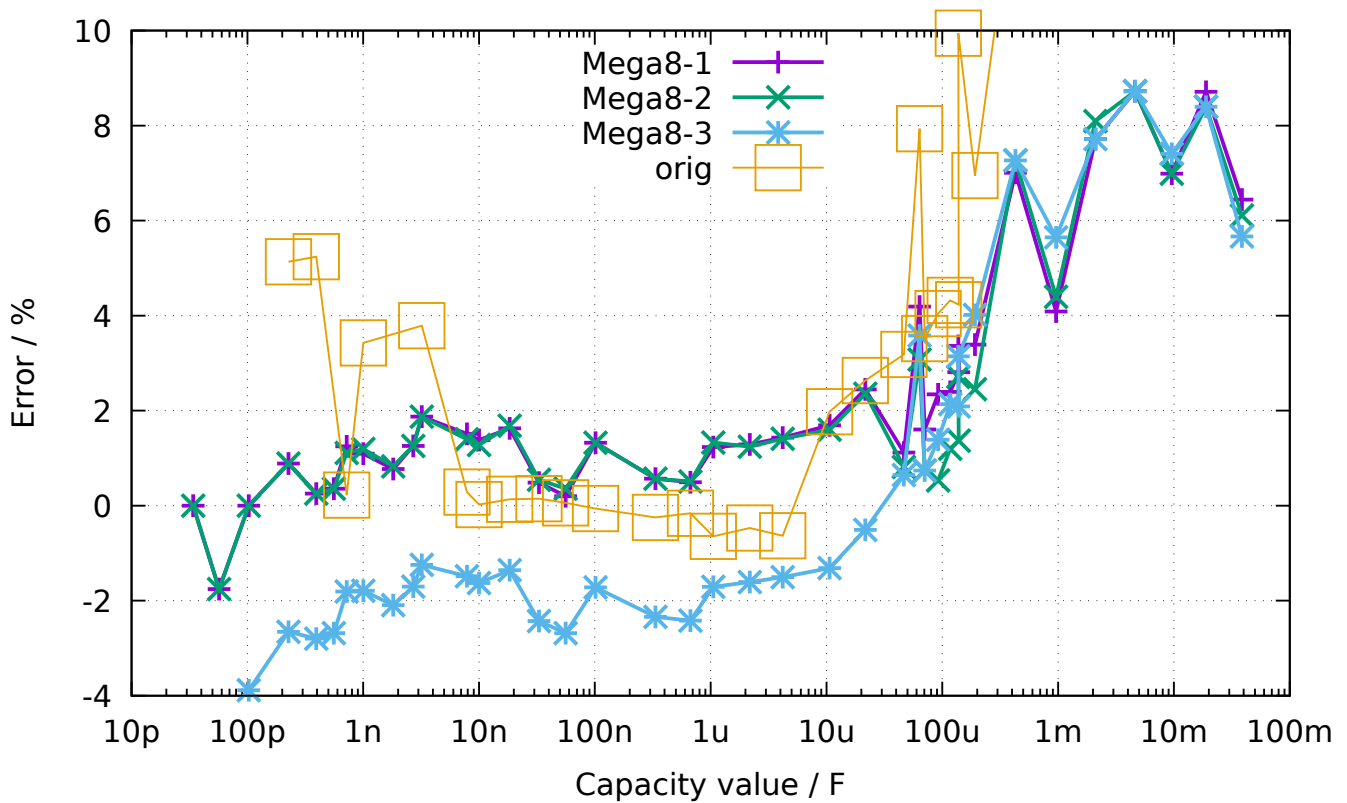
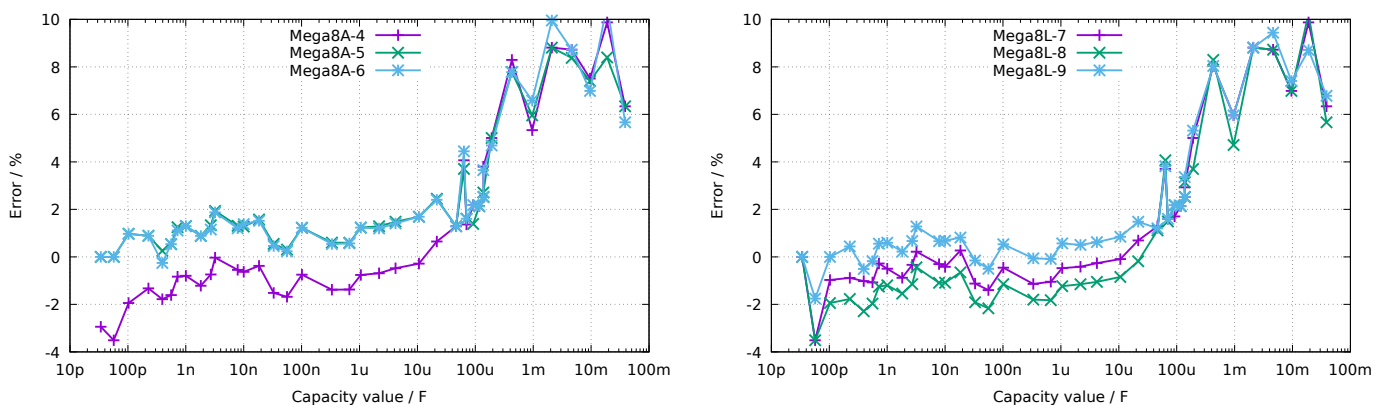


Figure 5.48. Error in % for capacitor measurements with ATmega8



(a) with three ATmega8A

(b) with three ATmega8L

Figure 5.49. Relative error of capacitor measurement

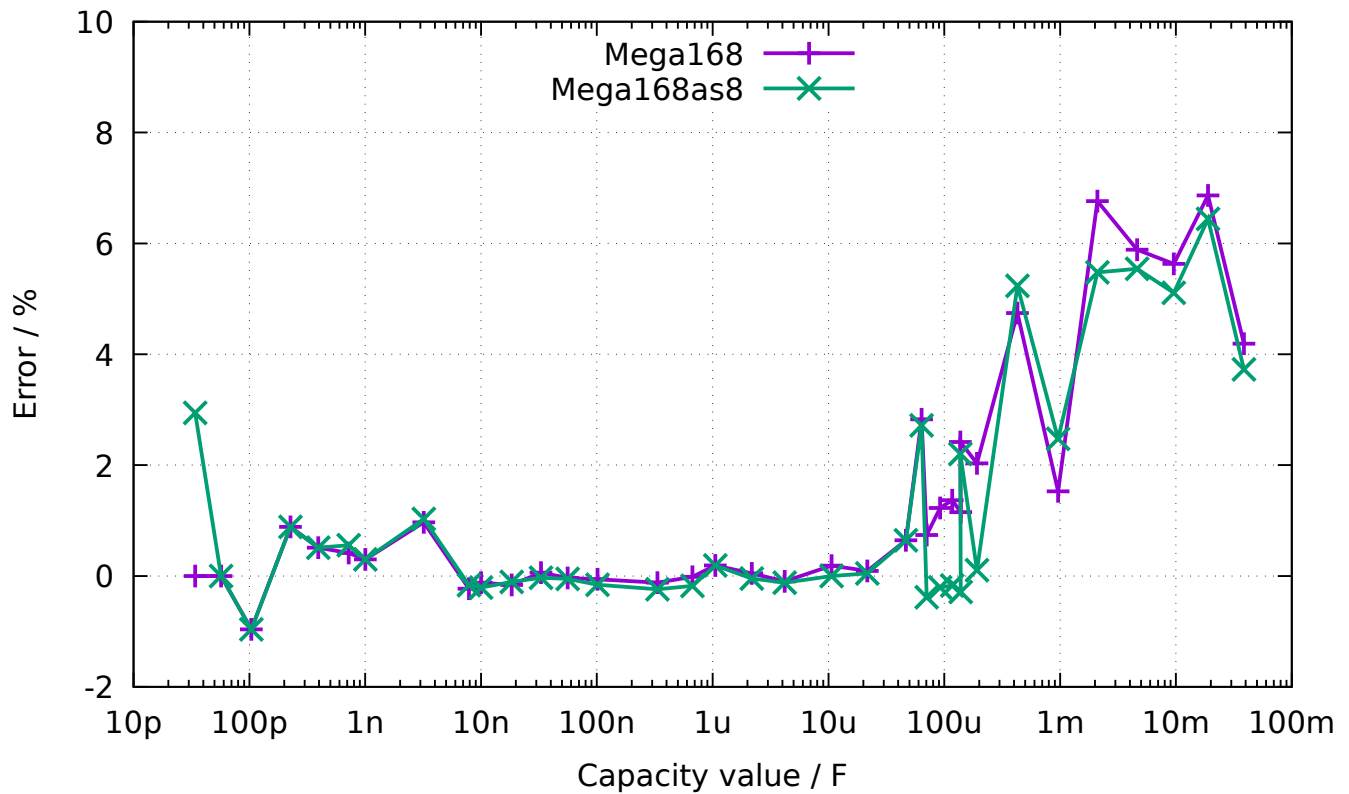


Figure 5.50. Error in % for capacitor measurements with ATmega168

Figure 5.51 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.

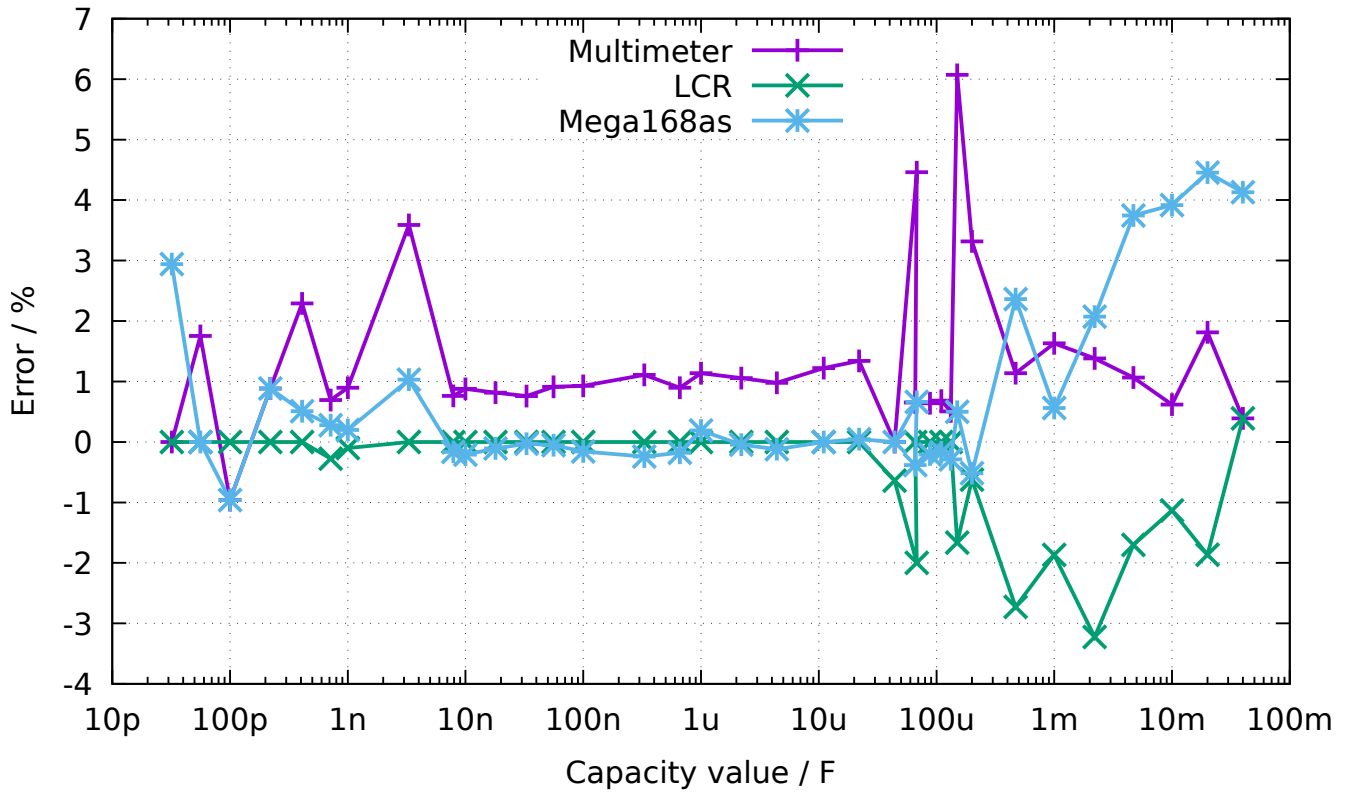
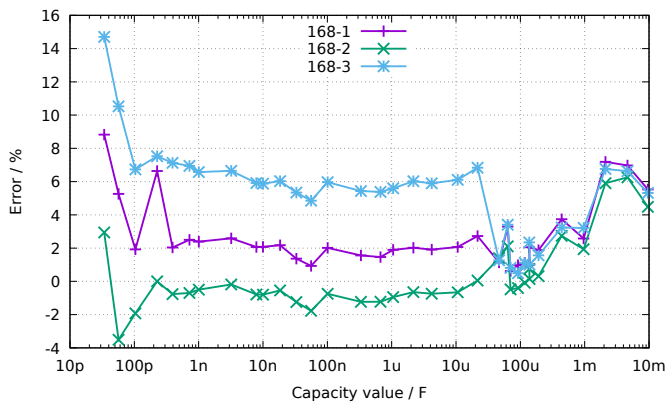
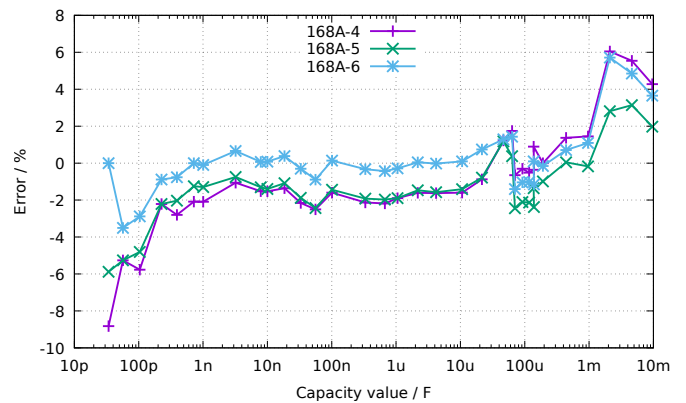


Figure 5.51. Comparison of capacity measurement results of Multimeter, LCR-meter and ATmega168

The differences of measurements of three different ATmega168 processors are shown in figure 5.52a . 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 5.52b and three different ATmega168PA processors are shown in figure 5.53. The results of three ATmega328 are additionally shown in figure 5.54a and the results from three ATmega328P are shown in figure 5.54b. 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 $12cm$ 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 5.52. capacity measurement error, not calibrated

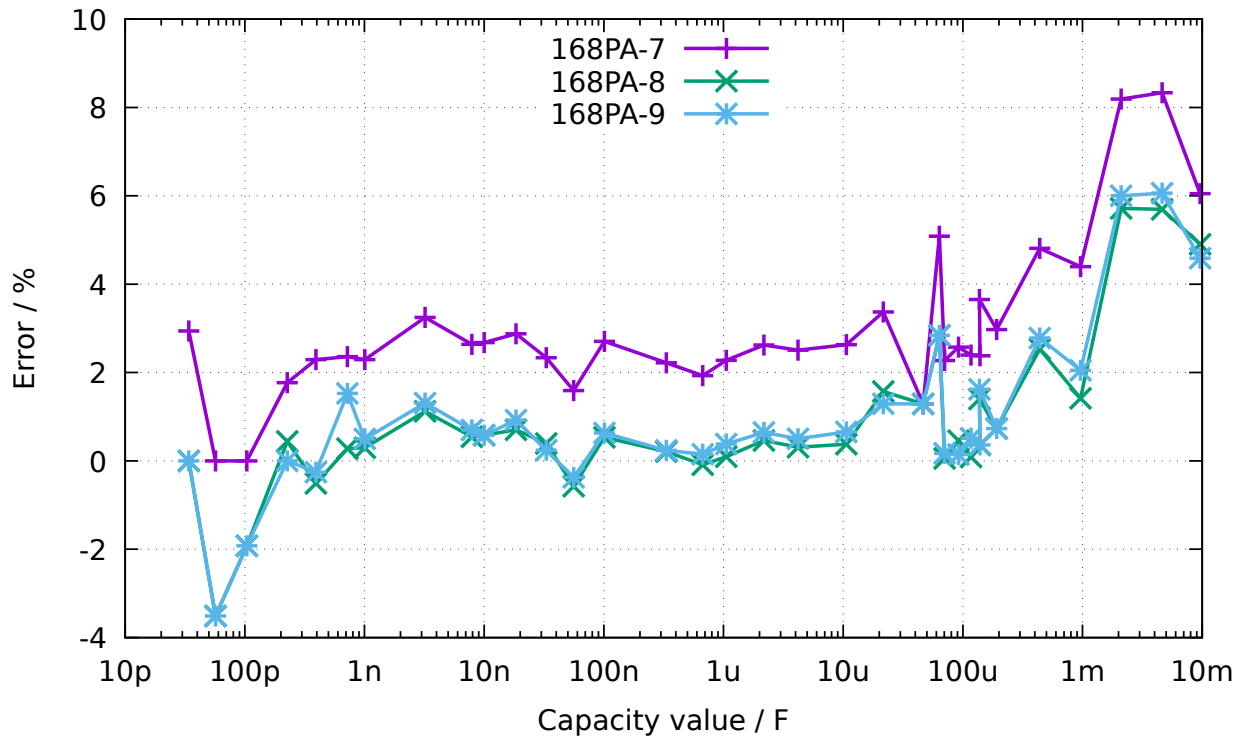
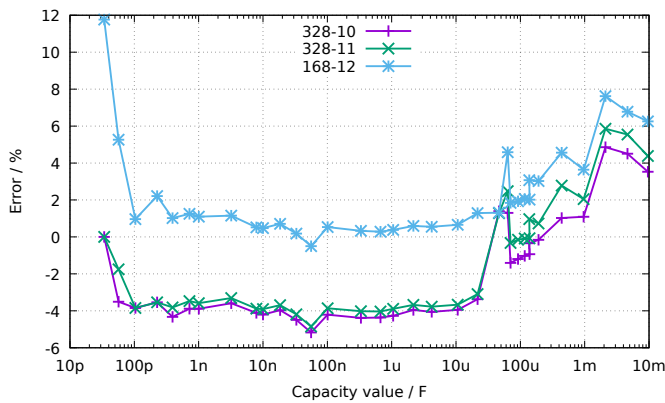
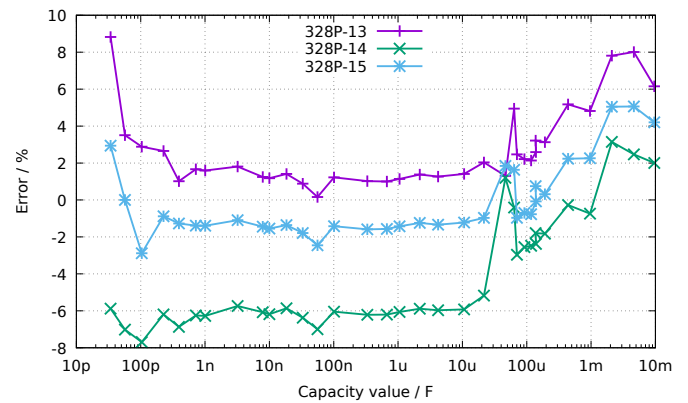


Figure 5.53. capacity measurement error of three ATmega168PA, not calibrated



(a) three ATmega328



(b) three ATmega328P

Figure 5.54. 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 $1mV$ 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 5.55 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 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.

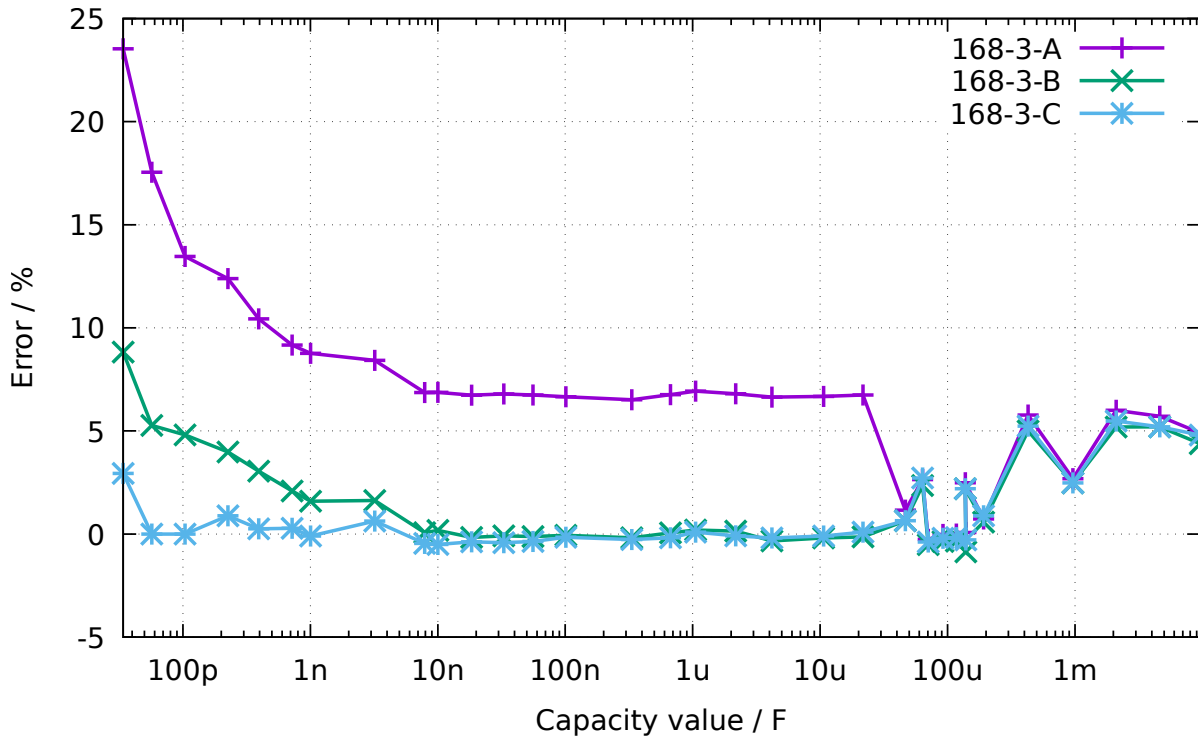


Figure 5.55. Improvement of the capacitor measurement of one ATmega168

5.3.11 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 5.52a, 5.52b and 5.53. 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 5.56 . 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 5.52a, 5.52b and 5.53 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 between $100nF$ and $20\mu F$. It should be a film capacitor, as far as possible not a ceramic capacitor und in no case a electrolytic capacitor. You don’t need to know the exact value of this capacitor.

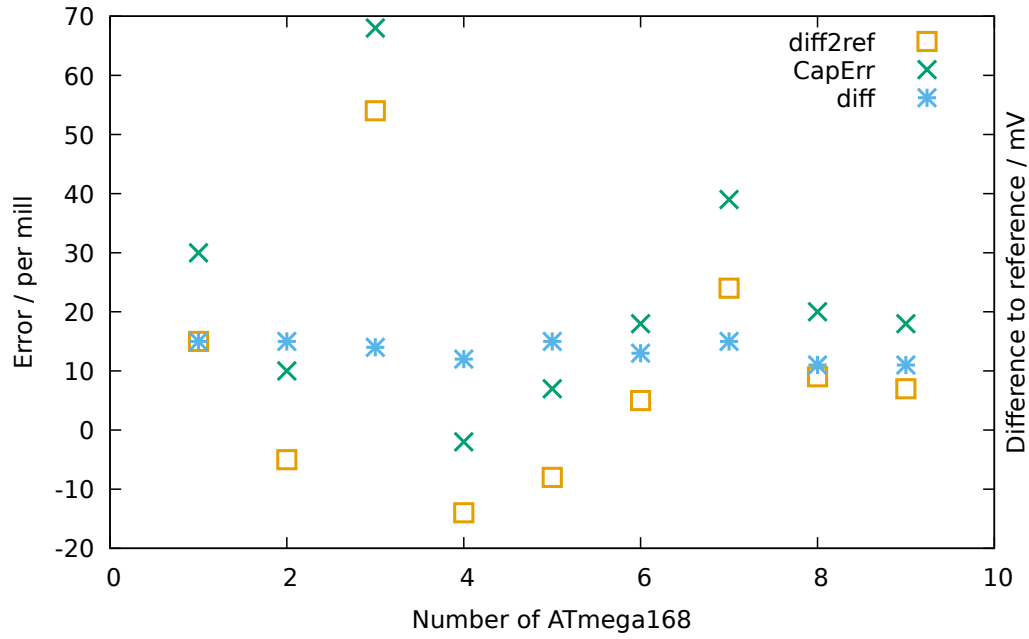
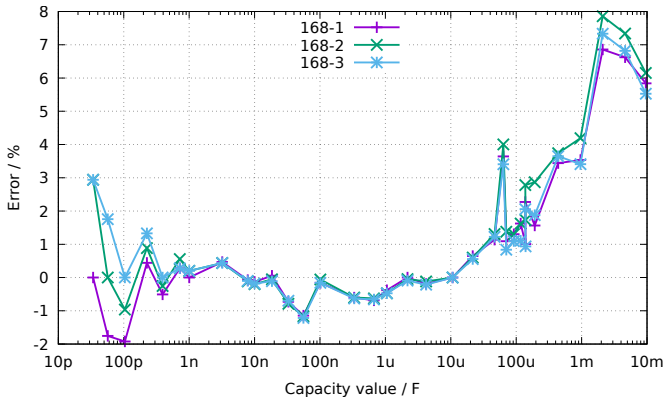
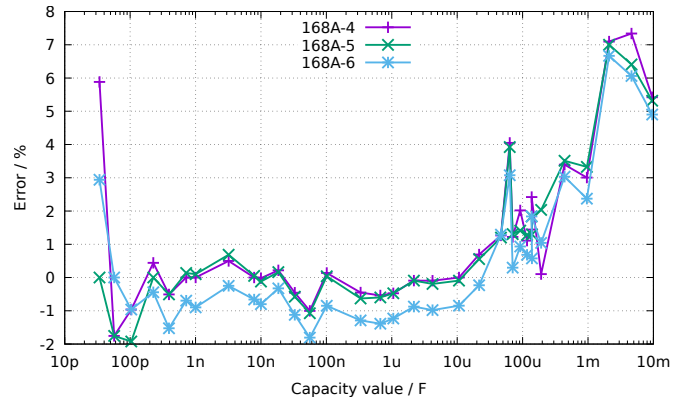


Figure 5.56. Date of nine ATmega168 processors

The figures 5.57a, 5.57b, 5.58, 5.59a and 5.59b 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 the Makefile option “PARTNO = ” must be adapted to the different processor type (“m168”, “m168p”, “m328” or “m328p”) for the avrdude program. 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 5.57. capacity measurement error, calibrated

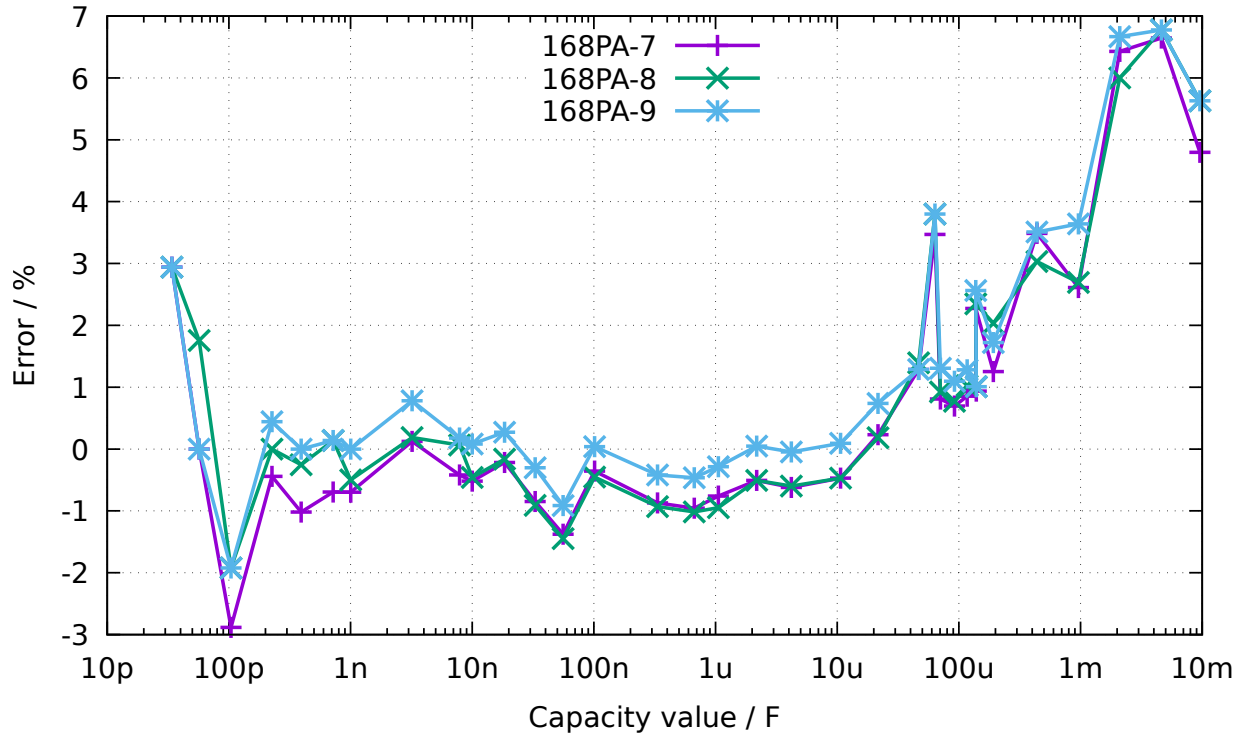
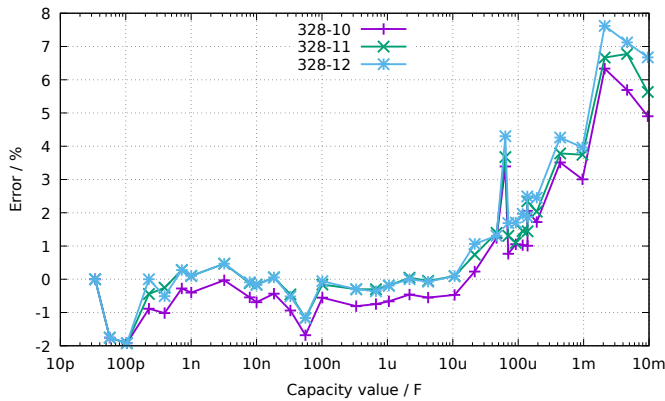
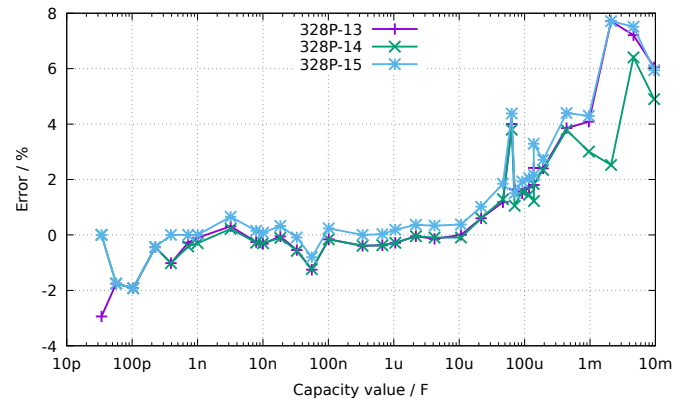


Figure 5.58. capacity measurement error of three ATmega168PA, calibrated



(a) three ATmega328



(b) three ATmega328P

Figure 5.59. 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 5.60 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.

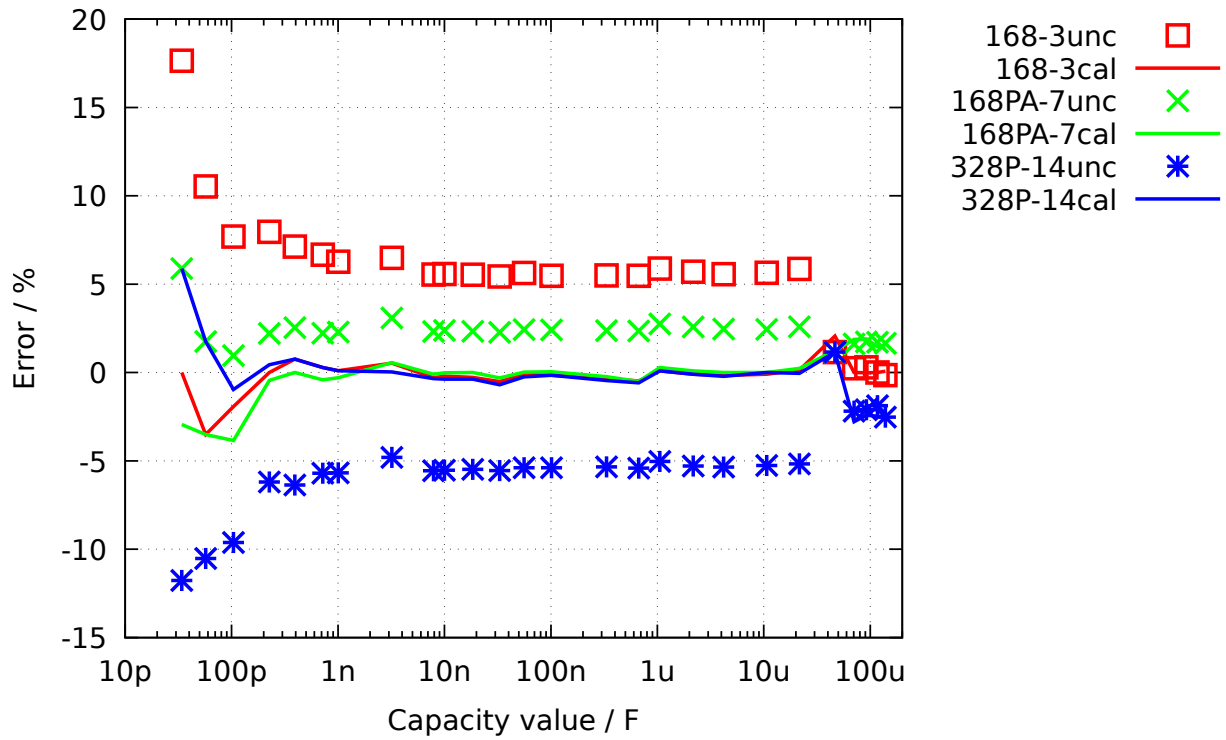


Figure 5.60. Error of capacitor measurement of three ATmega, before and after the calibration

The circuit with a ATmega644 or ATmega1284 provides a capacitor for calibration at the printed board. The figure 5.61 shows the results of the capacitor measurements with a ATmega1284, with the on board $100nF$ ceramic capacitor as well as with a external $220nF$ foil capacitor, compared to the results of a ATmega328 on another printed board.

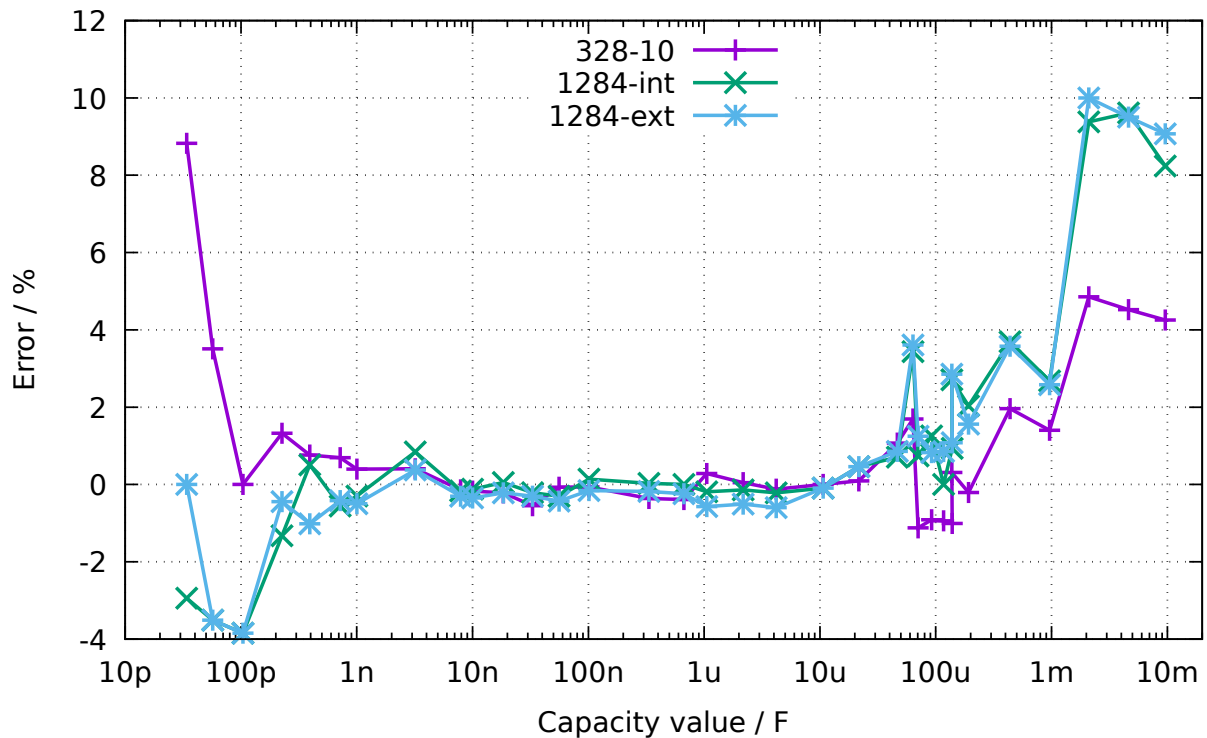


Figure 5.61. Error of capacitor measurements with a ATmega1284 compared to the ATmega328 results

5.4 Measurement of inductance

The measurement of inductance values will be done as separate part with all found resistors with less than 2100Ω . The method of measurement is based on the growing of current by formula $I(t) = I_{max} \cdot (1 - \exp(-\frac{t}{\tau}))$ after switching on the current. The time constant $\tau = \frac{L}{R}$ is proportional to the inductance L , but reverse proportional to the resistor R . The current can only be measured indirectly with the potential drop of a resistor.

Unfortunately the time constant will be reduced additionally by the relative high resistance 680Ω , for that the measurement of little inductance values is additionally made difficult with the $8MHz$ clock. To get the time constant, the voltage at the 680Ω resistor will be monitored by the analog comparator. If the voltage drop at the 680Ω resistor is higher than the voltage of the internal reference, this event will be notified to the 16-bit counter, which is started at the same time of switching current on. The counter will save the state of this event. If the counter will overrun, this will be counted by the program. After the event, the counter will be stopped by the program and the total time will be built with the saved counter stage and the overflow counter. The positive side of the coil will be switched from VCC to GND and hold in this stage until monitoring of the voltages of both pins shows, that no current is detected. The figure 5.62 shown a simplified diagram of the measurement situation.

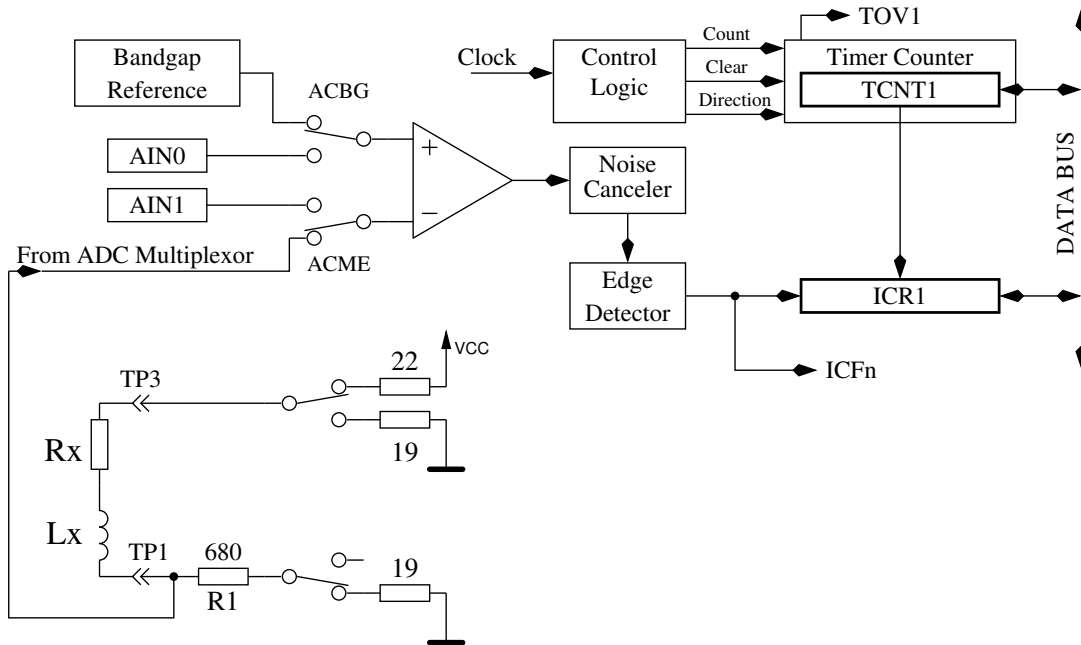


Figure 5.62. Measurement of inductances with the comparator

With the supply voltage VCC and the sum of all resistors in the electric circuit the maximum current I_{max} and from that the percentage of the reference voltage to the maximum voltage at the 680Ω resistor can be calculated $U_{max} = I_{max} \cdot (680 + 19)$. With the formula $L = -\frac{t \cdot R_{ges}}{\log(1 - \frac{U_{ref}}{U_{max}})}$ the inductance can be calculated. The natural logarithm will be taken out of a built-in table. A inductance resolution of $0.1mH$ is taken for this type of measurement.

In order to also measure lower inductance values, the 680Ω resistor will be omitted in the current loop, if the resistance value of the inductor is measured with less than 24Ω . Only the output resistance of the port (19Ω) will be used for measurement of the current. In this special case the peak current will be greater than the value, that the specification of the ATmega allows. Because this will be true only for a very short time, I expect no damage of the ATmega ports. For this type of measurement a resolution of inductance of $0.01mH$ is selected. To avoid a longer time with excessive current, the additional measurement with delayed start of the counter will always be done with the 680Ω resistor.

To get better fitting measurement results, a zero offset of 6 is subtract from the counter reading, if the measurement is done without the 680Ω resistor. Otherwise a zero offset of 7 or 8 is subtracted. With great inductance values the parasitic capacity can cause a quick rise of current, so that the comparator will response immediately. To get the value of the inductance anyway, the measurement will be repeated with a delayed start of the counter. With this methode the voltage grow caused by the current increase of the inductor will be detected by the analog comparator instead of the current peak of the parasitic capacity. The measurements are always done in both current directions. The program will select the higher result of measurement in the same current direction, but the lower result of the different current direction as the displayed result.

5.4.1 Results of the inductance measurements

The figure 5.63 shows the results of the measurement of different inductors. The Inductors above $1H$ are relays or primary sides of power transformers, for which measurements are difficult because the iron core has residual remanence.

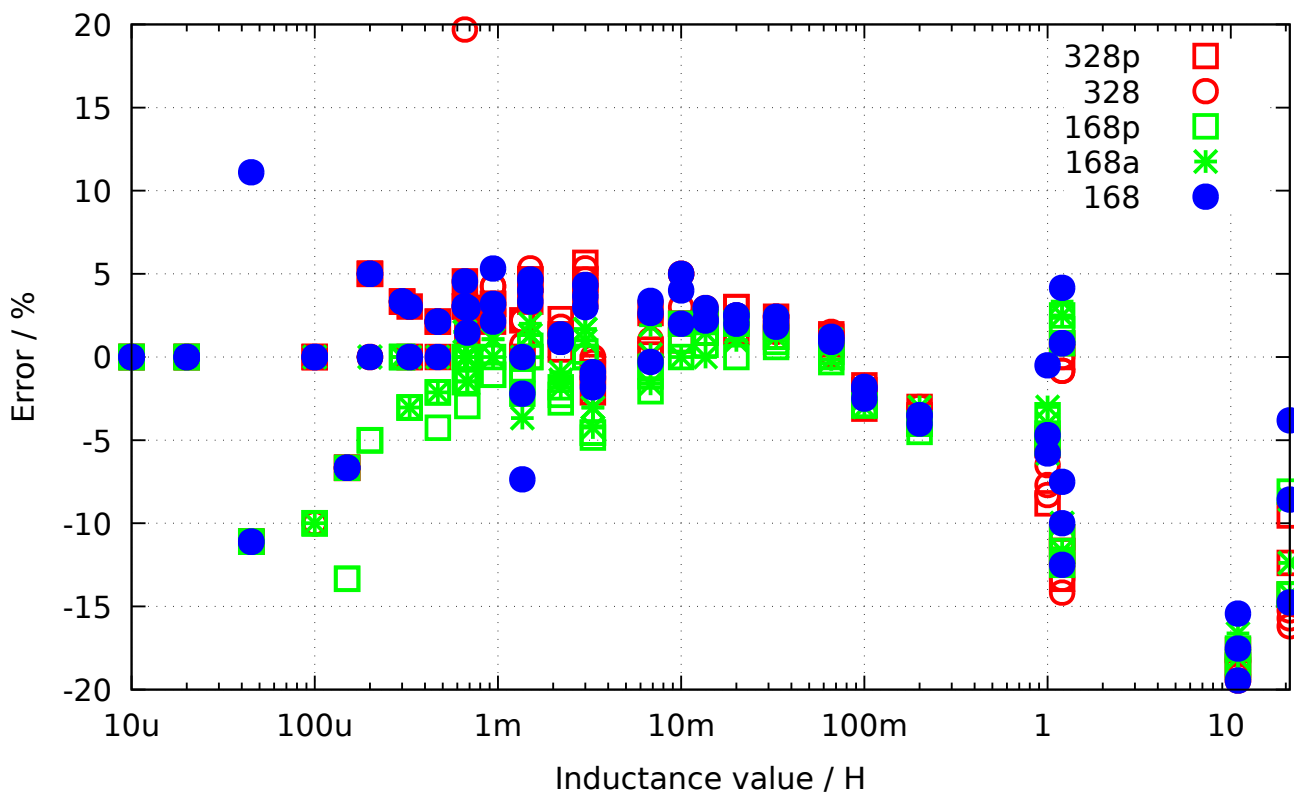


Figure 5.63. Error of inductance measurement of 15 different ATmega

5.4.2 Measurement of small inductances with the sampling method

The smallest detectable inductance is 0.01 mH with the normal measurement technique. For high frequency application the measurement of smaller inductances is required. The normal method use the speed of the current grow to measure the inductance. This method can not be used for the sampling technique, because the measurement circuit uses no additional resistors with the coil. The current will grow to forbidden high values very fast. We can prevent a damage of the ATmega only with a very fast break of the current flow. For the sampling method it is difficult to get this fast current turning-off and additionally this critical process must be done many times in series. For this reason the radio amateur Pieter-Tjerk (PA3FWM) has used another method to get the inductance values. Together with a parallel connected capacitor the inductor build a resonant circuit. With a

short current pulse this circuit begins to oscillate some times without further stimulation. With the sampling method the frequency of this oscillation can be measured. Because one side of the coil must be connected to GND level for this measurement, there are two problems with the measurement. Once the negative voltage of the oscillation is limited by the internal protection diode of the ATmega to about 0.6 V. For this reason the positive part of the oscillation will never reach a higher voltage than 0.6 V too. Additionally the ADC of the ATmega can only measure positive voltage values. Therefore all negative parts of the oscillation is read as zero. Anyway could Pieter-Tjerk find a solution, to measure the resonant frequency with a practical precision. With the resonant frequency the inductance can be computed, if the capacity value is known. For this reason the calibration program is extended with a capacity measurement for later inductance measurement use. The capacitor will be requested with the message "1 —||— 3 10-30nF(L)". A default value of 18 nF is preselected for the uncalibrated tester. The value of the parallel capacitor for inductance measurement should be selected rather high to get a low resonant frequency with small inductance values. A capacitor with high quality like foil-type should be selected because the quality factor of the resonant circuit is additionally measured by monitoring the decrease of the amplitude. With a high quality capacitor the total quality factor of the resonant circuit is build by the quality factor of the coil.

There is no additional operation required with the parallel connected capacitor. The resonant circuit is usually detected automatically. When the resonant circuit is detected, the text "if" and the expected value of the parallel capacity is written behind the inductance value in line 2. In this case the resistance value of the coil is written at the end of line 1. You should check the resistance value with a separate coil measurement without the capacitor, because the resistance measurement of the resonant circuit often fail! A additional output line shows the measured resonant frequency and the quality factor (Q=) of the resonant circuit.

If there is no resonant circuit detected, the resistance value and the inductance is shown in line 2. When a resonant frequency of a single coil is detected, the frequency value and the quality factor of the coil is also shown in a additional line.

For a air coil with 6 turns and a parallel connected capacitor with 18 nF the sampling method get the following result:

```
260nH if 18.1nF
2306kHz Q=38.7
```

This result was measured with a operation frequency of 8 MHz. A similar result was also measured with a 25 cm long copper wire, which was formed to one big circle. The inductance is measured too high in this example, because the parallel capacitor was a wound foil type with build in inductance. The following table 5.7 shows the measurement results of some coils with low inductance, which are all measured with a tester clocked with 16MHz.

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

Table 5.7. Measurement results of some coils with low inductance

The capacitors in this table are all types with low inductance like the WIMA MKS series. With a wounded 18.2nF capacitor you get a inductance of 196nH instead of the 151nH from the table. All inductors without the last one are self made coils. The inductance value in brackets are computed values with the coil parameters. The last 6.3μH coils is industrial build and labled with 6.3μH. But the measurement with a LCR-meter at 100kHz gives a result of 7.12μH! You can find big differences for the quality factor Q with the same coil and different parallel capacitors with nearly the same capacity value. For the coil with 12 turns you see a quality factor 50.2 with the 18.2nF capacitor and a quality factor 20.8 with the 20.3nF capacitor. A program fault may be the reason for this difference. Therefore I show the ADC data for the 12-turn coil with the 18.2nF and 20.3nF capacitors in figure 5.64 for secureness. You can find the difference of the quality factors also in the raw data clearly. Probably the used type of the capacitor is the reason for the difference of the quality factor.

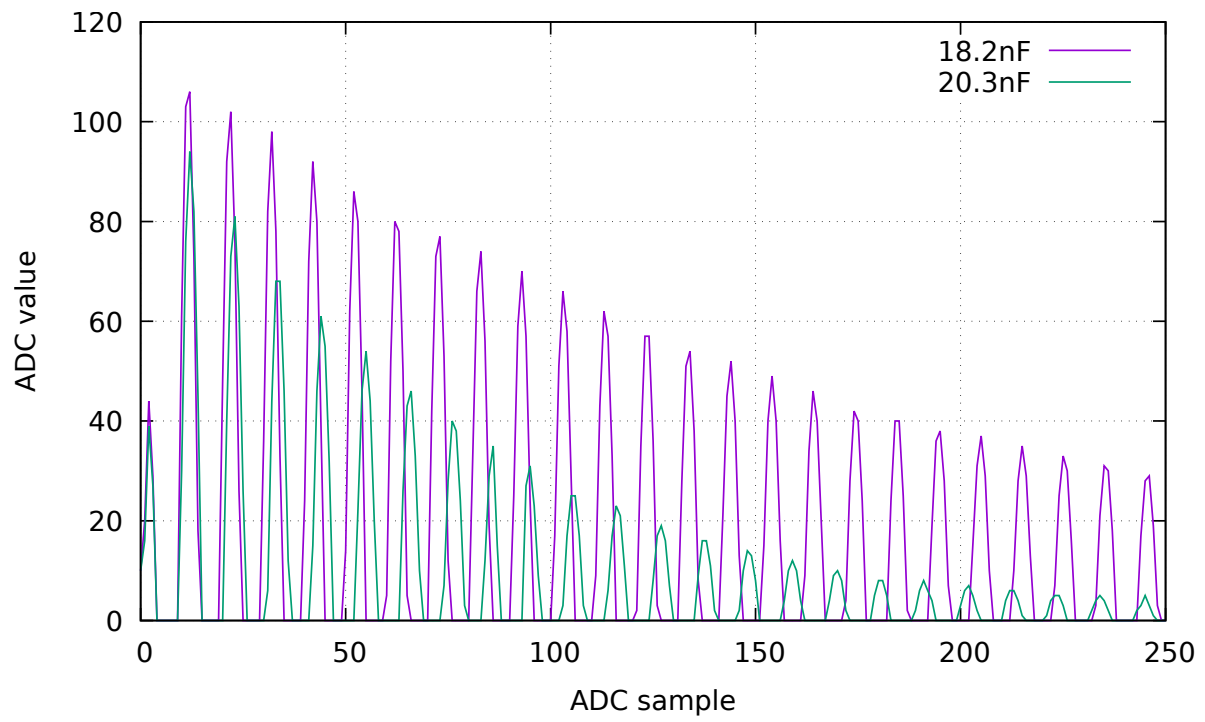


Figure 5.64. ADC data of 2 resonant circuits with the same 12 turn coil

5.5 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, if you confirm within two seconds with pressing the start key. This confirmation is implemented to prevent the tester going automatically to the self test by connecting a defect transistor. 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”. You can configure self test only for a ATmega168 or ATmega328. Before the test steps begin, the zero resistance of the connected probes is determined for all three combinations (T1:T3, T2:T3 and T1:T2). This zero resistances will be subtracted for the future ESR and resistance measurements below 10Ω . Only zero resistance values below 0.90Ω are accepted, because this correction values are not uses for measurement of resistor values above 10Ω . If you use cables for measurement, you should only take cables with low resistance values. If the later measured resistance results fall below the particular zero resistance for more than 0.2Ω , the tester will be resetted to “uncalibrated”. This will be marked by a acticated cursor during the tests. The separate steps of the self test function 1 to 7 is displayed on row 1 of the LCD display with the letter T followed by the step number. Every step is repeated 4 times, before the program continues with the next step. But if you hold the start key pressed, when the test cycle 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 every 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 $470k\Omega$ 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 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 5.65. The theoretical value with respect to the internal resistance of the pins should be:
$$\frac{5001 \cdot (19+680)}{(19+680+680+22)} = 2493 .$$

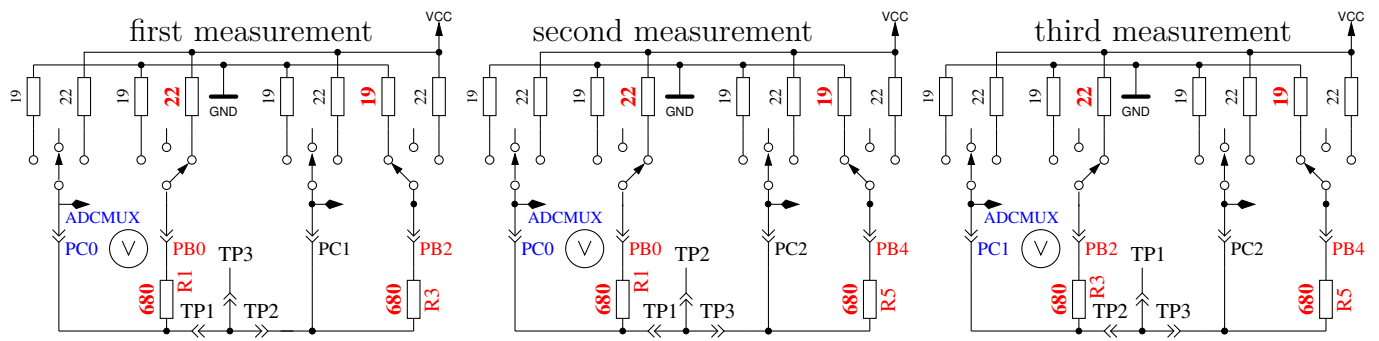


Figure 5.65. Comparison of 680Ω resistors

3. **Comparing of the 470kΩ resistors.** Now the display shows in row 1 “+RH- 12 13 23”. The same procedure as done in step 2 is repeated with the 470kΩ resistors (symbols RH). All results are shown as difference to the theoretical value. The theoretical value is this time $\frac{5001 \cdot (19 + 470000)}{(19 + 470000 + 470000 + 22)} = 2500$ 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). This step will finish only if you release the connections between the probes.
5. **This step tests the capability of GND (-) connected 470kΩ 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 470kΩ resistors (H) to pull the test pins to VCC (+).** Row 1 shows the text “RH+”. The results are shown als difference to VCC and should be near zero. Great differences from the best value for test 5 and 6 are errors such as isolation problem, flux material or damaged port.
7. **This Step tests the voltages of the 470kΩ/680Ω resistor divider.** The voltage difference to the expected voltage of the 470kΩ / 680Ω resistor dividers is shown in row 2 of the LCD for all three terminals. Differences of more than some mV can be caused by the assembly of wrong resistor values.
8. **Measuring of internal resistance of pin output switched to the GND signal.** This test and the follwing tests will only be done, if the option AUTO_CAL is selected. The internal resistance of the port C outputs switched to GND (-) are measured with the current of to VCC (+) switched 680Ω resistors, see Figure 5.66. 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. The resistor value will be shown in the next test.

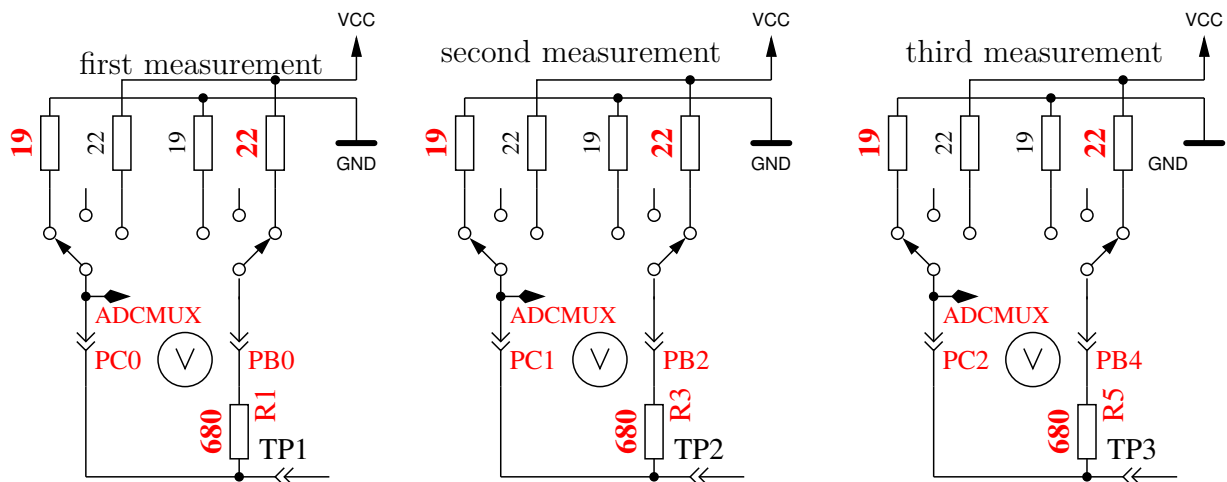


Figure 5.66. Measurement of internal resistance of Port C switched to GND

9. **Measuring of internal resistance of port outputs switched to the VCC (+)signal.**

The needed current is generated with to GND connected 680Ω resistors . It are the same measurements as those in test 8 to the other side as you can see in Figure 5.67. With the following steps the resistance is computed: To get the current, the following is computed: $(VCC - (resultoftest8) - (resultoftest9))/680$. To get both resistor values, the voltage (result of test 8 or 9) is divided by this current. The result for this test will then be notified in row 1 with the text “RI_Hi=”, the resistance value (Ω) to the GND side is displayed in row 2 with the text “RI_Lo=”. Beginning with version 1.06k of the software, the port output resistance values are determined at the beginning of every measurement. The values are only shown by this step.

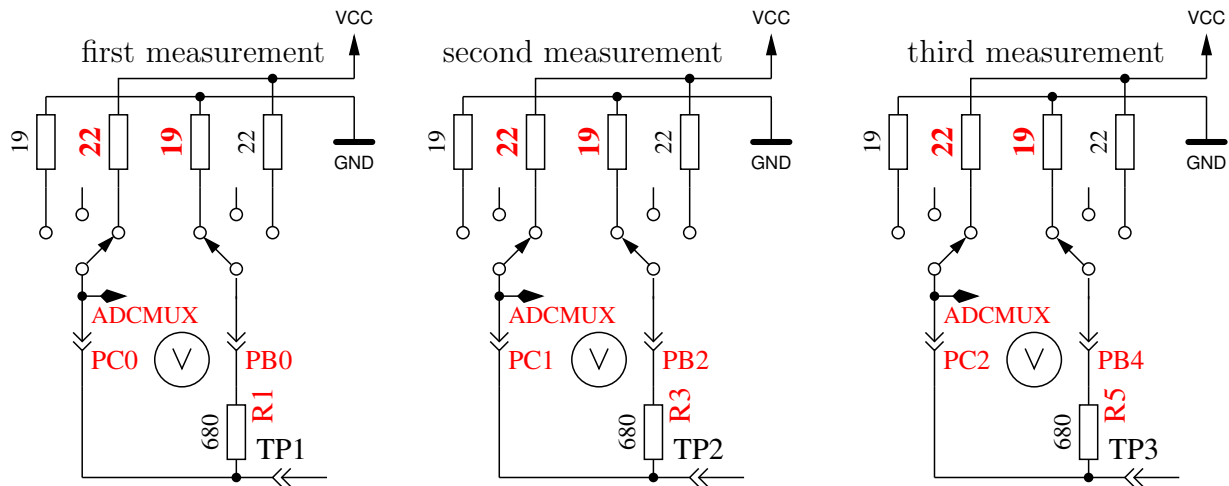


Figure 5.67. Measurement of internal resistance of Port C switched to VCC

10. **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 in display row 1 following the “C0”. All three values are shown in pF units. For this measurements no predefined zero offset is respected. The zero offsets of pin combinations in opposite order is also measured. The results will be written to the EEprom, if all values are less than $190pF$. This will be notified by the output of “OK” in row 2. The found zero offsets are used for further capacity measurements with respect to the pin combination. If there is any measurement found with a capacity value $20pF$ below the particular zero offset, the tester will be resetted to “uncalibrated”. This will be noticed by a activated LCD cursor during further tests. 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 $3pF$ greater compared to a empty socket.

If the tester is configured with the SamplingADC function, the zero capacity values is determined with double the count of all pin combinations. The reason for this is, that the zero capacity is measured for the charge and the discharge of all pin combinations separately.

11. **Wait for the connection of a capacitor to pin 1 and pin 3.** The message “1 —||— 3 >100nF” is shown in row 1 of LCD. To prepare the measurement of the comparator offset voltage, 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 between $100nF$ and $20\mu F$. You should never use electrolytical capacitors, use film capacitors instead.
12. **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 is 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 successful, the correction value is short shown with the text “REF_C=” in row 1 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. The measurement result is shown in row 2 with the text “REF_R=”. Your REF_R_KORR value is a additional offset to this automatic find out difference value.

13. **Wait for a capacitor for measuring of low inductace values** If the tester is configured with the SamplingADC function, a known capacitor is required for the computation of the inductance value from the resonant frequency of a parallel connected coil. Practical capacity values can be choosed from $10nF$ to $27nF$. A suitable capacitor must be connected to pin 1 and pin 3, if the message “1 —||— 3 10-30nF(L)” is shown in line 1 of the display. Exa-
cly this capacitor should be connected later parallel to a little coil, if the calculation of the coils inductance value from resonant frequency is desired. If your tester does not include the SamplingADC function, this step is omitted.

At the end of test function the text “Test End” is shown in row 1 and the version number of software is shown in row 2. If the Makefile option FREQUENCY_50HZ is set, 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. Figure 5.68 shows the oscillograph curves of both 50Hz output pins with crystal operation.

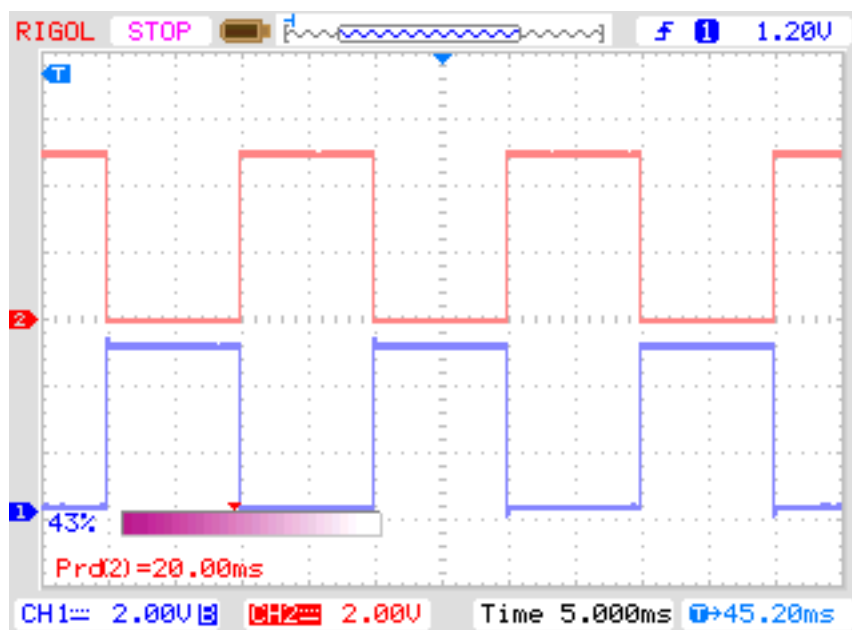


Figure 5.68. Oscillograph curve with the 50Hz outputs of Port 2 and 3

If you don't use the crystal clock version, the result may be inexact. An exact 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.

5.5.1 Some Results of the Selftest Function

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	5.69
Test 2	RL-Mean	0	5.70a
Test 3	RH-Mean	0	5.70b
Test 5	RH-Low	0	5.71a
Test 6	RH-High	0	5.71b
Test 8	R out Lo	131	5.72a
Test 9	R out Hi	151	5.72b
Test 10	Cap zero offset	30	5.73
Test 11	Reference correction	0	5.74

Table 5.8. Table of the selftest figures

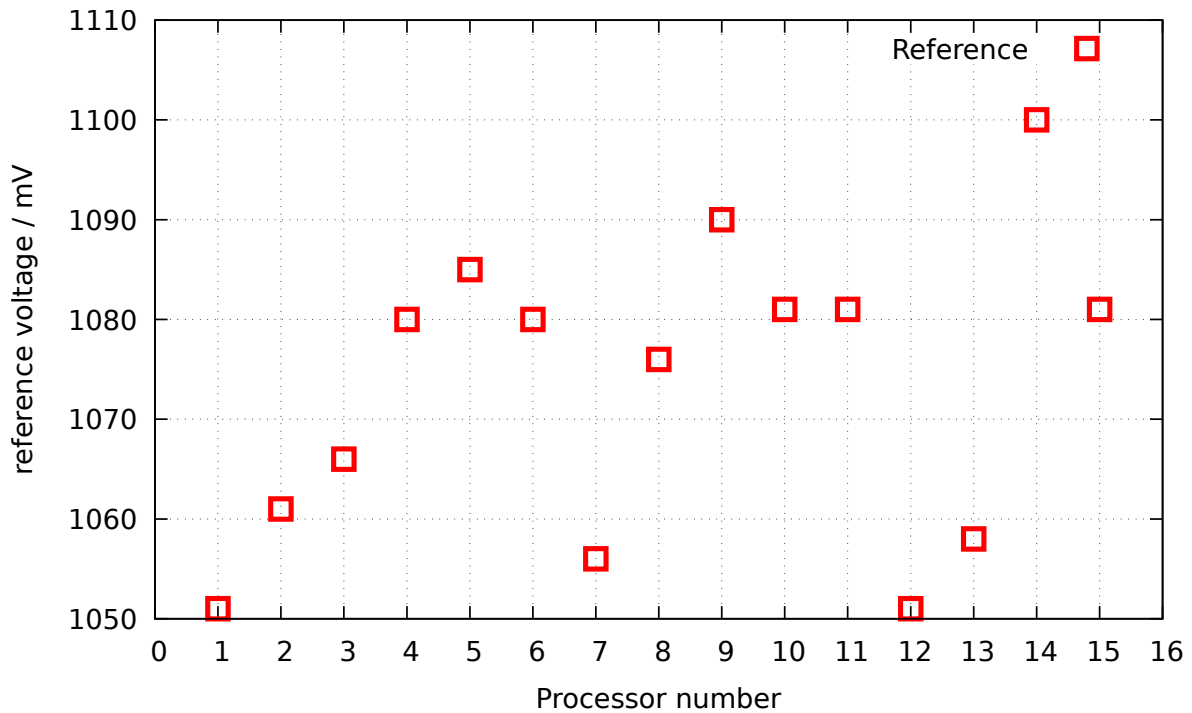
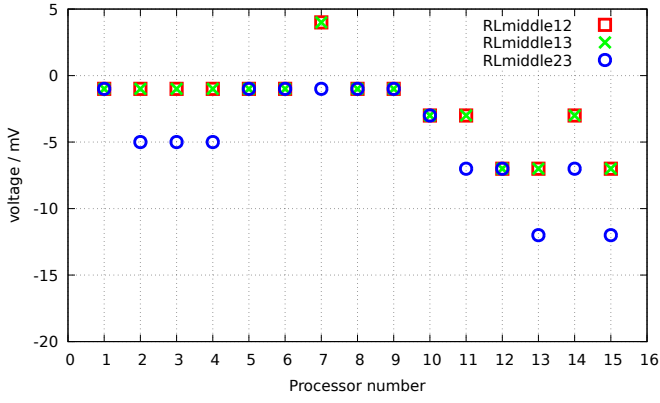
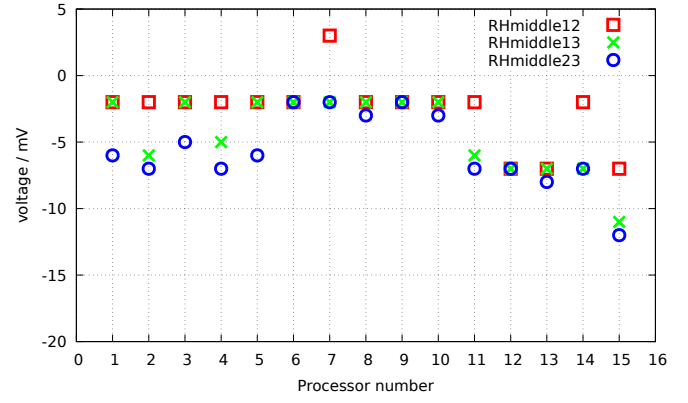


Figure 5.69. Selftest: Reference-Voltages

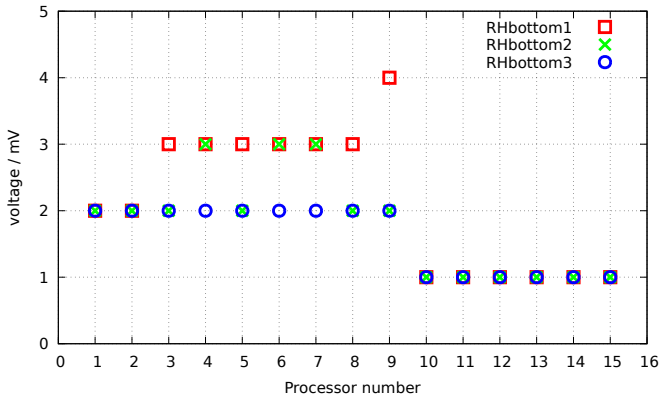


(a) with 680Ω

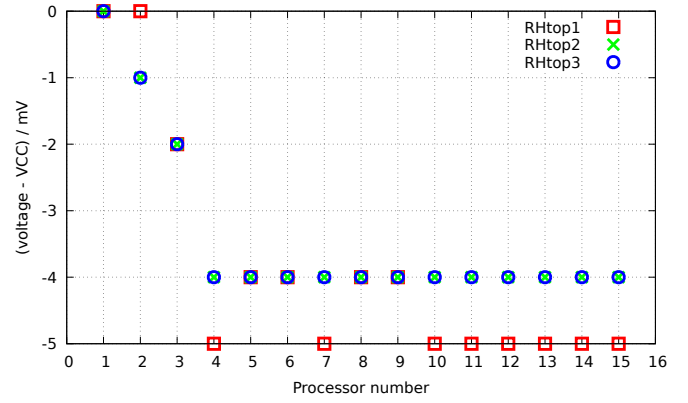


(b) with $470k\Omega$

Figure 5.70. Selftest: difference to ideal mean voltage

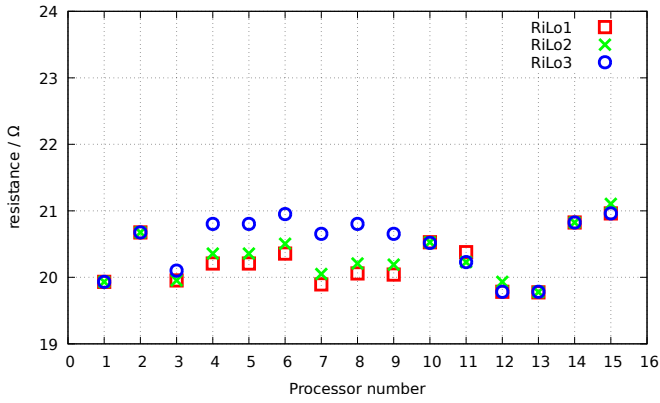


(a) with $470k\Omega$ to $0V$

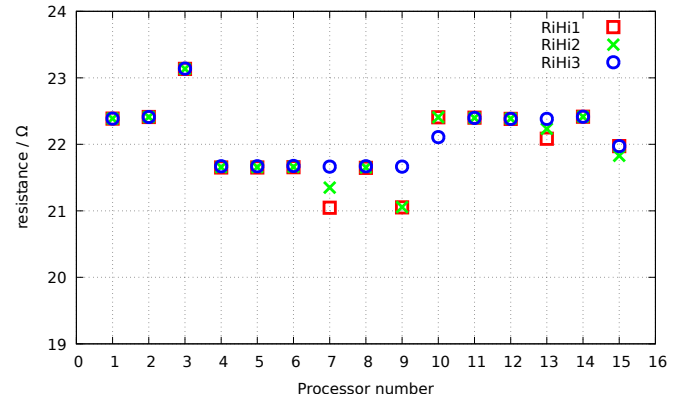


(b) with $470k\Omega$ to $5V$

Figure 5.71. Selftest: Input voltage



(a) with 680Ω to $5V$



(b) with 680Ω to $0V$

Figure 5.72. Selftest: Output resistance

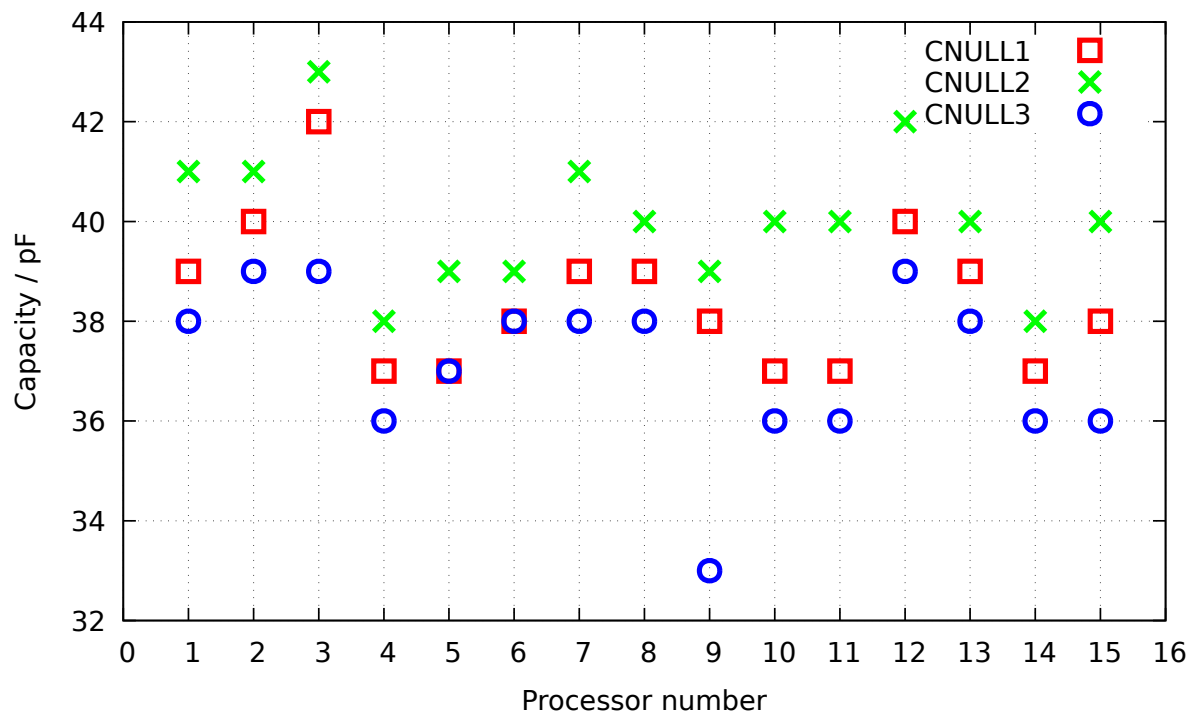


Figure 5.73. Selftest: zero offset of the capacity measurement

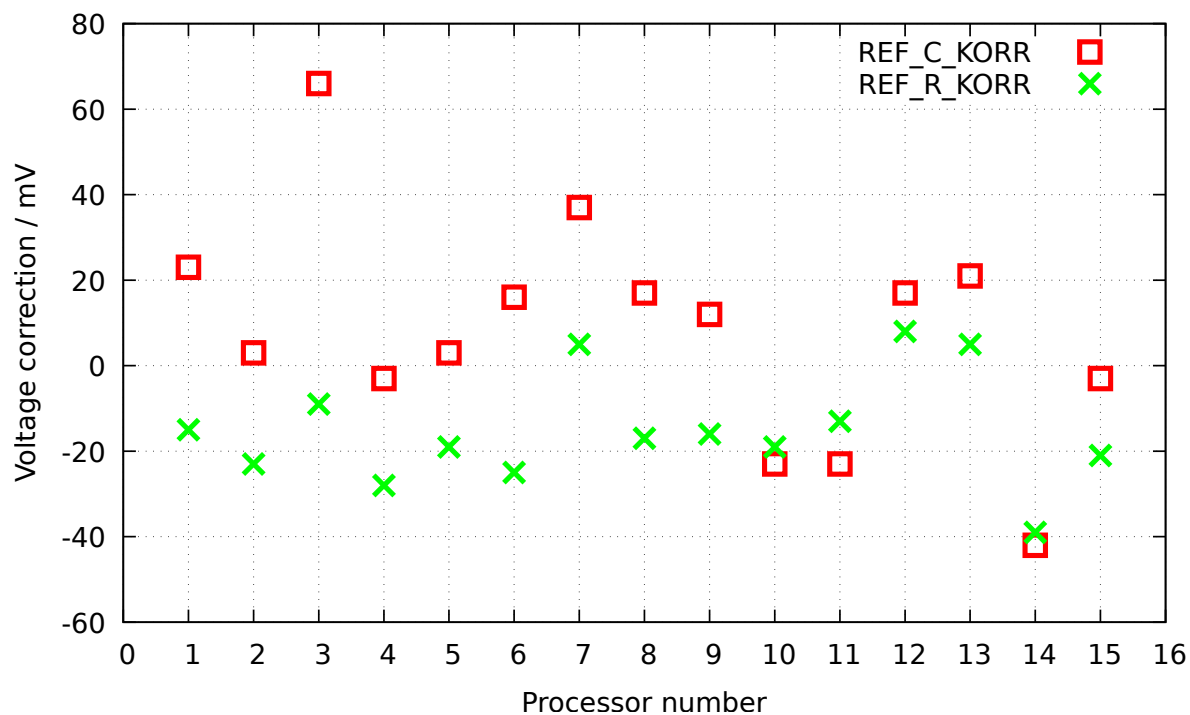


Figure 5.74. Selftest: correction values after automatic calibration

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 precessors and the found correction voltages (REF_R_KORR) after the automatic calibration in figure 5.75. You can see, that the automatic calibration values nearly follow the external measured values.

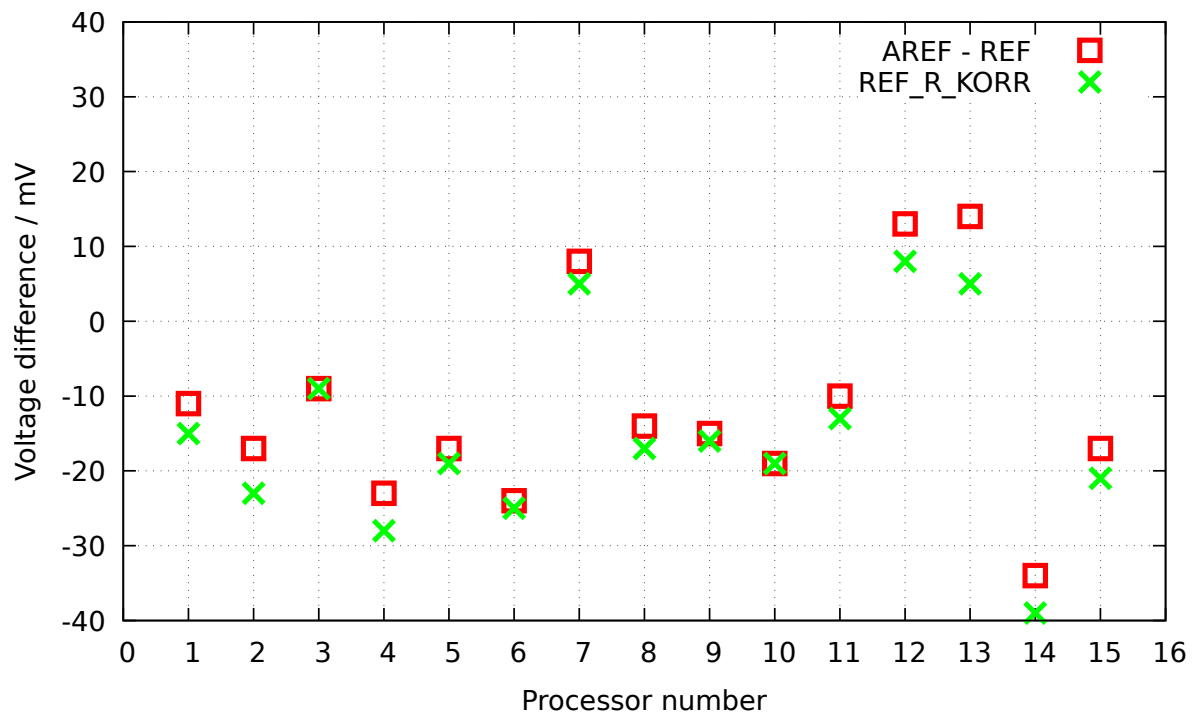


Figure 5.75. Selftest: Voltage difference of the internal reference

5.6 Measurement of frequency

Beginning with Version 1.10k the frequency measurement can be selected with a control menu. The normal frequency measurement is done with counting the falling edges of input signal T0 (PD4) with counter 0 for one second. For maintaining of a accurate second, the counter 1 is used with a 256:1 prescaler of the CPU frequency. The 16 bit counter of the ATmega can be used up to $16MHz$ CPU frequency with the prescaler to serve the second period in one pass. For the start and stop of the counter 0 ist the compare register B and A of the counter 1 used. To prevent a unstable delay by polling the compare event signals, for starting and stopping the Interrupt Service of both counter 1 compare events is used. The time delay of both Interrupt Service Routines is nearly equal. To maintain a accurate second period is a constant delay insignificant. With analysing the assembler code, the difference in time can be adjusted.

For frequencies below $25kHz$ the normal measurement is followed by a measurement of period time. This additional measurement is only followed after a normal frequency measurement. This will be done by measuring the time of a selectable count of the Pin Change interrupts of the PCINT20 (PD4) input with the counter 0. By measuring the period both, the negative puls width and the positive puls width, should be at least $10\mu s$. The counter 0 is used with full clock rate. This results to a resolution of $125ns$ for $8MHz$. With a greater count of measurements periods the resolution can be reduced. By using a measurement period of 125 periods, the middle resolution for one period is $1ns$. To prevent the inexactness of start and stop the counter 0, the start of counter 0 is started within the first pin change interrupt of PCINT20 and will be stopped with the last pin change interrupt with the same interrupt service routine. The count of periodes is choosed, that the measurement time is about 10 million clock tics. The part of error from one clock is only 0.1ppm with this choise. With a $8MHz$ clock the measurement time is about 1.25 seconds. From this mean value of one period a frequency with better resolution is computed too.

For checking the procedure, two testers are measured against each other. First the test frequencies are generated with tester 2 and measured with tester 1. After that the testers are swapped and the measurement is repeated. Figure 5.76 shows the results of both measurement series. The nearly constant errors can be explained with a little frequency difference of both crystals.

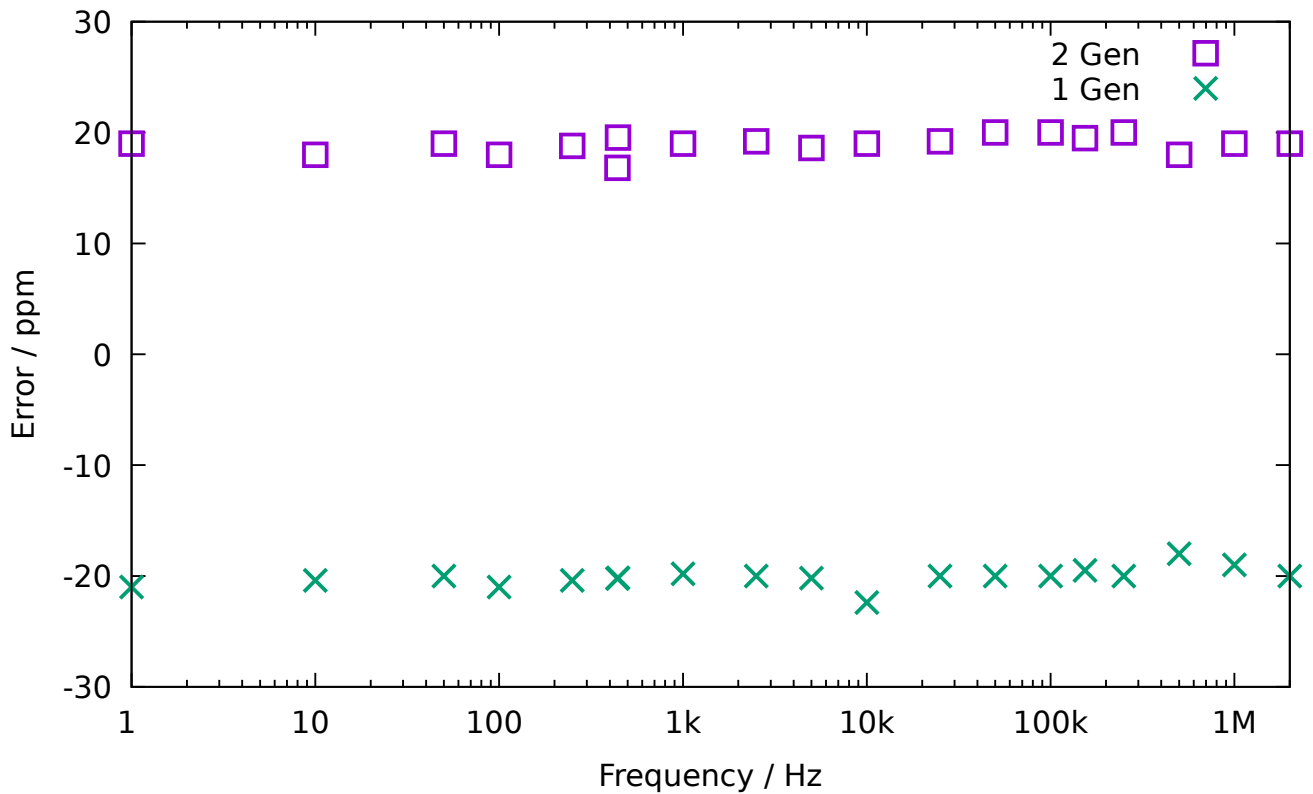


Figure 5.76. Relative errors of frequency measurement

5.6.1 Calibration of frequency with a GPS- or GLONASS-receiver

Tuning of the crystal frequency is possible with a adjustable capacitor (5-25pF) at the crystal. The one pulse per second (1PPS) from the GPS receiver **UP501** from **Fastrax Ltd.** and from the GPS/GLONASS receiver **GNS701** from **Global Navigation Systems GmbH** has been tested successfully for the calibration of the crystal frequency. The measured period could be adjusted to exactly 1000.000ms. Only the last digit can toggle with one unit. Of course the frequency of the crystal is temperature dependent. Therefore you can not expect a very good long time stability. The figure 5.77 shows the used circuits with a UM232 USB-serial converter as connection of the receiver modules to a computer. The UM232 converter support the circuit sowohl with both, the 5V and the 3.3V supply voltage from the USB supply voltage. No connection to a computer is required for operating the receivers. Only the 5V supply voltage must be provided to the USB connector.

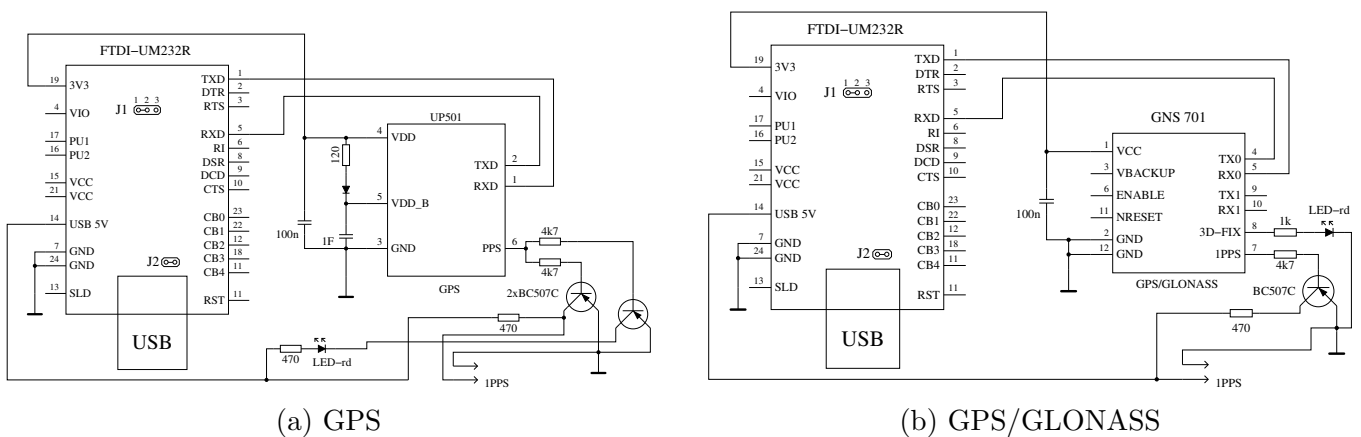


Figure 5.77. Generation of a 1PPS signal with GPS receiver

5.6.2 Calibration of the crystal frequency with a clock modul

To tune the crystal frequency of the transistor tester you must first exchange one of the crystal capacitors against a trimmer with adjustable capacity. The advantage of using watch modules for frequency calibration instead of GPS or GLONASS modules is, that you need no clear view to sky. You can adjust the frequency nearly at any place. I examined watch modules with a DS3231 chip and a printed board labeled with “ZS-042”. The examined modules are probably build in china and are all of the boards are equipped with a DS3231M chip. The DS3231M chip use a MEMS (Micro Electro Mechanical System) resonator instead of the DS3231SN chip, which use a crystal with $32768Hz$. A similar MEMS resonator is also used by the DCP1301 chip. The picture 5.78 shows one of the used modules.

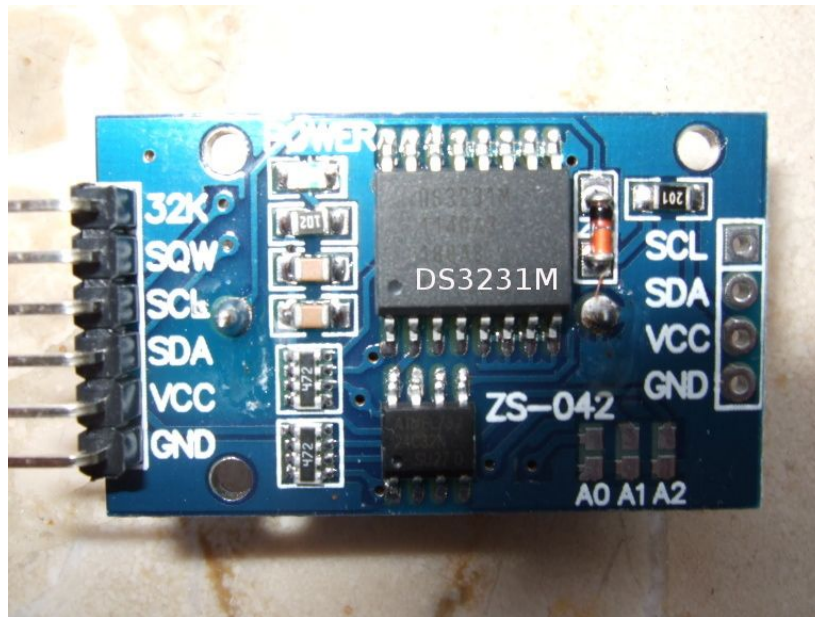


Figure 5.78. One example of the examined DS3231 modules

Both DS3231 chip versions use a internal temperature measurement to control the frequency of the time base in a manner, that the frequency drift with temperature is nearly perfect compensated in a wide range. Unfortunately the provided $32kHz$ Signal of the DS3231M chip can not be used for a frequency calibration. I have measured different frequencies of $32641Hz$, $32710Hz$, $32730Hz$ and $32748Hz$ for all four tested modules. All of these Frequencies are far away from the expected exactly $32768Hz$. If you connect the modules to a Arduino UNO, you can also enable a 1PPS ($1Hz$) output to the SQW pin. This output is so stable and exactly, that it can be used for the calibration. The data sheet of the DS3231M promise for the 1PPS output a accuracy of $\pm 5ppm$ for the full temperature range of $-45^{\circ}C$ to $+85^{\circ}C$, while the accuracy of the $32kHz$ output is only specified to $\pm 2.5\%$ ($25000ppm$).

The data sheet of the DS3231SN chip promise a accuracy of $\pm 3.5ppm$ for the full temperature range from $-40^{\circ}C$ to $+85^{\circ}C$ and a accuracy of $\pm 2ppm$ for the temperature between $0^{\circ}C$ to $+40^{\circ}C$. The DS3231SN chip use a internal clock crystal with a frequency of $32768Hz$, which is controlled with switchable capacitors over a wide temperature range. With the known temperature drift of the crystal and a build in temperature measurement the frequency of the controlled crystal is nearly constant. To examine these chips too, I have replaced the DS3231M chips by DS3231SN chips for all four modules. With a fresh calibrated transistor tester ($16MHz$ model) I have measured the output frequency of all modules with the same result of $32.76800kHz$. Very rarely a difference of $0.03Hz$ was shown during the measurements. This is only a difference of $1ppm$. A fractional amount of $1Hz$ is only shown, if the frequency is calculated of a period measurement. The limit for period measurement is changed from $25kHz$ to $33kHz$ to enable the period measurement for the $32768Hz$

signal.

Chapter 6

Signal generation

The different signal generation modes are only available with a ATmega328 processor. You must also enable the dialog function with the Makefile option `WITH_MENU`. The menu is called by a long key press. The available functions are shown in the second row of the LCD. You can select the shown function by a long key press. The next function is automatically shown after 5 seconds or after a short key press.

6.1 Frequency Generation

The frequency generation is started, if you select the “f-Generator” function by a long key press. The frequency output is done with the 680Ω resistor at measurement port TP2. The measurement port TP1 is switched to GND. The frequencies are build with the 16 bit counter from the CPU clock frequency ($8MHz$ or $16MHz$). The frequency can be selected for each digit position beginning with the 1Hz position with the digits 0-9. The highest selectable position is the 100kHz position. For this position numerals between 0 and 20 can be selected. Therefore frequencies up to 2Mhz are selectable. Without the rotary encoder the position value can be increased with a short key press ($< 0.8s$). With a rotary encoder you can change the position value in any direction. With a longer key press you can change the position of digit change. With a symbol in row 1 of the frequency line the direction of the position change is shown. A $>$ symbol in row 1 indicate, that the next higher position will be selected. With a $<$ symbol in row 1 the next position will be the lower position (down to 1Hz). If the highest digit position (100kHz) is selected, the $>$ symbol is replaced by the character R. In this condition the longer key press will reset the frequency value to the initial value (1Hz). Because not every selected frequency can be generated correctly with the counter, the difference to the real generated frequency is shown in line 3 or behind the frequency value. If you hold the key pressed for a long time ($> 2s$), you can return to the dialog function and select the same or another function.

6.2 Puls width generation

The Puls Width generator is started, if you select the “10-Bit PWM” function by a long key press. The frequency output is done with the 680Ω resistor at measurement port TP2. The measurement port TP1 is switched to GND. The frequency of the output signal is always the CPU clock divided by 1024. This gives a result of $7812.5Hz$ for the 8 MHz CPU clock. Only the positive pulse width can be changed by a key press. With a short key press you can increase the positive puls width up to 99% in 1% increments. With a longer key press you can increase the pulse width in 10% increments. The pulse width reaches a value above 99% 100 is subtract from the result. The pulse width 0% generates a very small positive puls width.

Chapter 7

Known errors and unsolved problems

Software Version 1.12k

1. Germanium Diodes (AC128) are not detected in all cases. This is probably caused by the residual current. Cooling of the diode can help to reduce the residual current.
2. With bipolar transistors a collector - emitter protection diode was not detected, if the collector residual current $ICE0$ is high. The problem was only detected with germanium transistors, which are complemented with a external diode.
3. The current amplification factor of germanium transistors can be measured too high because of the high residual current. In this case the basis emitter voltage will be very low. Cooling of the transistor can help to get a more correct current amplification factor.
4. Capacity value in reverse direction for Power Schottky Diodes such as MBR3045PT can not be measured, if only one diode is connected. The reason is a too big residual current of this diode. Sometimes the measurement is possible by cooling down the device (with cooling spray for example).
5. Here and there a wrong detection of the 2.5V precision reference is reported, when the PC4 pin (27) is unconnected. You can avoid this behaviour with a additional pull up resistor connected to VCC.
6. The diode function of a triac gate can not be examined.
7. Sometimes a problem with the Brown Out level of 4.3V is reported for ATmega168 or ATmega328 processors. This will cause a reset during capacity measurement. A reason is not known. The Resets will disappear, if the Brown Out level is set to 2.7V.
8. With the using of the sleep state of the processor, current of VCC power is changing more than using previous software versions. You should check the blocking capacitors, if you notice any problems. Ceramic capacitors with $100nF$ should be placed near the power pins of the ATmega. The using of sleep state can be deselected by the Makefile option `INHIBIT_SLEEP_MODE`.
9. The measurement of tantalum based electrolytical capacitors often make trouble. They can be detected as diode or can also be not detected as known part. Sometimes the measurement with swapped connection can help.
10. The Source and Drain pins can not be detected correctly with JFET's. The reason is the symmetrical structure of this semiconductors. You can notice this problem with the effect, that the display shows the same layout with the same parameters, if the Source and Drain pins

are swapped. I see no way to detect the Source and Drain pin correctly. But interchanging of Source and Drain in any circuit should usually not cause any problem.

Chapter 8

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_testpin(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 9

To Do List and new ideas

1. Add more and better documentation.
2. Think about how we can get the real internal resistance of port B output (resistor switching port) instead of assuming, that ports are equal.
3. Can discharging of capacitors be made more quickly, if the minus pin is additionally raised with the 680Ω resistor to VCC (+)?
4. 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.
5. Write User's guide for configuring the tester with the Makefile options and description of the build chain.
6. 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)?
7. Check the Port afterwards with self test function!
8. Idea for a New Projekt: USB version without LCD-Display, Power from USB, Communication to PC over a USB-Serial bridge.
9. Replacement of samplingADC function, which make use of the counter1 to manage the shift of the ADC S&H time.
10. Check the accuracy of measurement little capacitors with the samplingADC method.
11. Check the accuracy of measurement little coils with the samplingADC method.

Bibliography

- [1] <https://github.com/mikrocontroller-net/transistortester/> *Quellen des Transistortesters*. 02-2021
- [2] Markus Frejek *AVR-Transistortester*,. Embedded Projects Journal, 11. Ausgabe, 2011
- [3] <https://github.com/kubi48/> *Transistortester Dokumentation der Autoren*.
- [4] Atmel Corporation *8-bit AVR with 8KBytes In-System Programmable Flash - ATmega8(L)*,. Manual, 2486Z-AVR-02/11, 2011
- [5] Atmel Corporation *8-bit AVR with 4/8/16/32KBytes In-System Programmable Flash - ATmega48 - ATmega328*,. Manual, 8271D-AVR-05/11, 2011
- [6] Atmel Corporation *Atmel AVR126: ADC of megaAVR in Single Ended Mode*,. Application Note, 8444A-AVR-10/11, 2011
- [7] Atmel Corporation *Atmel AVR121: Enhancing ADC resolution by oversampling*,. Application Note, 8003A-AVR-09/05, 2005
- [8] <http://en.wikibooks.org/wiki/LaTeX> *LaTeX documentation*,. Guide to the LaTeX markup language, 2012
- [9] <http://en.wikibooks.org/wiki/Gnuplot> *Gnuplot documentation*,. Documentation for the plotting tool gnuplot, 2012
- [10] Wikipedia http://de.wikipedia.org/wiki/Equivalent_Series_Resistance *Explanation for ESR in german language*. Standardization and equivalent circuit of a capacitor, 2012
- [11] <http://www.xfig.org/userman> *Xfig documentation*,. Documentation of the interactive drawing tool xfig, 2009
- [12] <http://docs.gimp.org/2.6/de> *gimp documentation*. Documentation of the GNU Image Manipulation Program, 2010
- [13] <http://www.mikrocontroller.net/articles/AVR-Transistortester> *Online documentation of the Transistortester*, Online Article, 2009-2011
- [14] <http://www.mikrocontroller.net/articles/AVRDUDE> *Online documentation of avrdude programmer interface*, Online Article, 2004-2011
- [15] <http://www.mikrocontroller.net/topic/131804> *Thread from Markus*, Forum thread, 2009
- [16] http://www.mikrocontroller.net/articles/AVR_Transistortester *Short description of new features of the TransistorTester von Karl-Heinz K.*, Online Article, 2012

- [17] <http://www.mikrocontroller.net/topic/248078> *Thread from Karl-Heinz, Thread and new software versions*, 2012
- [18] <http://www.mikrocontroller.net/articles/WinAVR> *Information about WinAVR in german language*, Online Article, 2012
- [19] <http://sourceforge.net/projects/winavr/files> *Source for WinAVR package*, Download source, 2012
- [20] <http://www.mikrocontroller.net/topic/248078?page=5#2922341> *Patch for WinAVR, Setting of Fuses with avrdude*, Download source, 2012
- [21] <http://www.orientdisplay.com/pdf/ST7565.pdf> *Datasheet of the ST7565 graphical controller*, Download source, 2014
- [22] Maxim Integrated Products, Inc. <http://maximintegrated.com> *DS3231: Extremely Accurate I²C-Integrated RTC/TCXO/Crystal*, Data Sheet, 19-5170;Rev 10; 3/15, 2015
- [23] Maxim Integrated Products, Inc. <http://maximintegrated.com> *DS3231M: 5ppm I²C Real-Time Clock*, Data Sheet, 19-5312;Rev 7; 3/15, 2015