AVR Transistortester

From the Mikrocontroller.net collection of articles, with contributions by various authors (see version history)

table of contents

Introduction (German)

Original design by Markus Frejek . Further developed by Karl-Heinz Kübbeler, see **this discussion thread** .

I continued Markus Frejek's transistor tester project and specially developed the software. Due to the improved properties, the name component tester was suggested. I myself still see the outstanding

property in the automatic determination of transistor type and property, as it was developed by Markus Frejek.

The main **features** :

- Works with ATmega8, ATmega168, ATmega328 or also ATmega644 and ATmega1284 processors.
- The measurement results are displayed on an **LCD with 2x16 or 4x20** characters.
- Instead of the 2x16 character LCD, a **graphic display** with ST7565, NT7108 or ST7920 controller can also be used. A connection of an **OLED** display with SSD1306 controller is also possible with SPI or I2C interface. Color displays with ILI9341 or ILI9163 controllers can also be used.
- One-button operation with automatic switch-off.
- The device has three universal measuring ports (test pin).
- Automatic detection of **NPN** , **PNP** , N- and P-channel **MOSFET** , **JFET** , **diodes** and small signal **thyristor** and **TRIAC** .
- Automatic detection of the pin assignment of the components, the components can be connected as required.
- Measurement of the current amplification factor (hfe) and the base-emitter voltage for bipolar transistors, also for Darlington transistors.
- Automatic detection of a protection diode for bipolar transistors and MOSFETs.
- In the case of bipolar transistors with a protective diode, a parasitic transistor is detected in some cases (NPNp = NPN + parasitic PNP).
- Up to two **resistances** are measured in one measurement with a resolution of up to 0.1 ohm, the measuring range extending to over 50 MOhm. Resistance values below 10 Ohm are displayed for the ATmega168 / 328 with the ESR measuring method with a resolution of 0.01 Ohm.
- A connected **capacitor** can be measured in the range 35pF to 100mF with a resolution of up to 1 pF.
- If 32K flash memory is available, the Pieter-Tjerk SamplingADC method can measure capacitors below 100pF with a resolution of up to 0.01 pF.
- Resistors and capacitors are shown with their symbols, surrounded by the pin numbers found.
- The resistor and capacitor values are displayed with up to four decimal places in the correct dimension.
- Up to two diodes are also displayed with their symbols in the correct direction of flow, surrounded by the connection pin numbers and the additional information on the flow voltage.
- In the case of individual diodes, the capacitance value and, from version 1.08k, the current in reverse direction is also measured.
- For the ATmega168 / 328, it is possible to calibrate the zero capacity, zero resistance and other parameters in the self-test branch.
- For the ATmega168 / 328, **inductances** from around 0.01mH to over 20H can also be detected and measured.

- With at least 32K Flash memory, even small inductances can be measured with the SamplingADC method through a capacitor of known capacity connected in parallel. In addition to the oscillation frequency, the calculated inductance value and the quality are output.
- For the ATmega168 / 328, an ESR measurement (Equivalent Series Resistance) for capacitors over 20 nF with a resolution of 0.01 Ohm is integrated. With small capacitance values, however, the accuracy of the measurement becomes worse.
- for ATmega168 / 328, the voltage loss Vloss after a charge pulse is examined for capacitors over 5 nF. This allows the quality of the capacitors to be estimated.
- For the ATmega328 , further functions from a list are possible with a **menu** function that can be called up with a long **press of** the button (> 0.5 s). A short press of the button shows the next function. A longer press of the button starts the displayed function. Below is the list of the additional functions installed so far:
    - **Frequency** measurement on the PD4 pin, which is also used for the LCD connection. The pin is switched to input for the measurement. The applied frequency is first counted for 1 second. If the frequency is below 25 kHz, an average period is also measured and a frequency is calculated from this with a resolution of up to 0.001 mHz.
    - **Voltage measurement** at the PC3 pin, if this is not used for the serial output. With the ATmega328 with 32 pins (PLCC) the ADC6 or ADC7 pin can also be used. Since a 10: 1 divider is used at the input, voltages up to 50V can be measured. With an extension of the circuit (DC-DC converter), Zener diodes can also be measured.
    - **Frequency** generation at the TP2 port. A signal with an adjustable frequency of 1 Hz to 2 MHz can be output at the TP2 port via the 680 ohm resistor connected to the PB2 pin. The TP1 port is connected to ground.
    - **Pulse width modulation** with fixed frequency and adjustable pulse width on the TP2 port. Counter 1 is used as a 10-bit counter for this function. The TP1 port is connected to ground. The pulse width can be increased by 1% by briefly pressing the button and by 10% by pressing the button longer.
    - With a separate capacitance and ESR measurement, capacitors connected to TP1 and TP3 with a capacitance of around 2µF to 50mF can usually also be measured in the circuit. In doing so, however, particular care should be taken to ensure that the capacitors no longer have any residual charge.

The additional functions are time-limited, like the dialog function itself, if the POWER_OFF option is activated in the configuration file (Makefile). More detailed information with measurement examples can be found in the PDF documentation in German and English. Russian translations of the documentation are also available.

Introduction (English)

Original design by Markus Frejek

Refined design by Karl-Heinz Kübbeler, see this thread, most people there will also understand and answer in English.

I (Karl-Heinz Kübbeler) have carried on the *transistor tester* from Markus Frejek and mainly refined the software. Because of its improved performance the name component tester was suggested, but I myself see its purpose mainly in determination of the transistor type and its parameters.

These are the characteristics:

- Works with ATmega8, ATmega168, ATmega328 or ATmega644 and ATmega1284 processors.
- Shows results in a LCD of **2x16 or 4x20** characters.
- **Also a graphical display** with the ST7565, NT7108 or ST7920 controller is possible. Also a OLED display with the SSD1306 controller and communication via SPI or I2C interface is possible. You can also connect color displays with ILI9341 or ILI9163 controller.
- One-key-operation with automatic power off.
- Three test pins for universal use.
- Automated detection of **NPN**, **PNP**, N- and P-channel **MOSFET**, **JFET**, **diodes** und small **thyristors**, **TRIAC**.
- Automated detection of pin assignment, this means the device-under-test can be connected to the tester in any order.
- Measurement of hFE and base-emitter-voltage for bipolar junction transistors, also for Darlingtons.
- Automated detection of protection diodes in bipolar junction transistors and MOSFETs.
- Bipolar junction transistors are detected as a transistor with a parasitic transistor (NPNp = NPN + parasitic PNP).
- Up to two **resistors** will be measured with a resolution down to 0.1 ohm. The measurement range is up to 50 Mohm (Megaohm). Resistors below 10 ohm will be measured with the ESR approach and a resolution of 0.01 ohm if a ATmega168/328 is used. Beware: resolution is not accuracy.
- **Capacitors** in the range 35pF (picofarad) to 100mF (millifarad) can be measured with a resolution down to 1 pF.
- If the processor has at least 32K flash memory, you can use the samplingADC method from Pieter-Tjerk to get a resolution of up to 0.01 pF for capacitors with lower capacity than 100 pF.
- Resistors and capacitors will be displayed with their respective symbol, pin number and value.
- Up to two diodes will also be displayed with their correctly aligned symbol, pin number and voltage drop.
- If it's a single diode, the parasitic capacitance and reverse current will also be measured.
- For ATmega168/328 a self calibration of zero-capacitance, zero-resistance and other parameters is possible.
- For ATmega168/328 also **inductances** of 0.01 mH to 20 H can be detected and measured.
- If your processor has at least 32K flash, you can use the samplingADC method to measure lesser inductances with a parallel capacitor of known capacity. The resonant frequency and the computed inductance value is shown and additionally the quality factor.

- for ATmega168/328 a measurement of ESR (Equivalent Series Resistance) of capacitors greater than 20 nF is built in. The resolution is 0.01 Ohm. For lower capacity values the accuracy of ESR result becomes worse.
- For ATmega168/328 Vloss of capacitors greater 5 nF is examined. With this it is possible to estimate its Q-factor.
- For ATmega328 a **menu function** can be reached with a long key press (> 0.5 s). A short key press switches to the next function. A long key press starts the function. The list of built-in functions until now:
    - Frequency measurement at pin PD4. This pin is also used for the LCD and will be switched to input (High-Z) for the measurement. The frequency is measured for 1 second. If it is below 25 kHz, the period will be measured to improve accuracy. Resolution goes down to 0.001 mHz.
    - **Voltage measurement** at pin PC3, if it is not used for serial output. Since ATmega328 has 32 pins (PLCC), also ADC6 or ADC7 can be used. A 10:1 divider is used, so voltages up to 50 V can be measured. With an additional DC-DC converter, Zener diodes can also be measured.
    - **Frequency generation** at port TP2. A 680 ohm resistor connected to pin PB2 can be used to generate a signal with 1 Hz to 2 MHz at port TP2. Port TP1 is ground.
    - Variable PWM (pulse width modulation) with fixed frequency at port TP2. 10-Bit counter. Port TP1 is ground. Short press increases pulse width by 1 %, long press by 10 %.
    - There is a separate capacitance and ESR measurement available. Capacitors of 2 µF to 50 mF can usually be measured in-circuit. You have to ensure beforehand that the capacitor is not holding a charge anymore.

You can read detailed information with measurement examples in the PDF-documentation in English and German. A Russian translation is also available. The PDFs are linked in the download sections of this page.

Introduction (Danish)

(The original (former) design can be accessed via this link:
http://www.mikrocontroller.net/articles/AVR-Transistortester)

Further developed design by Karl-Heinz Kübbeler, see this forum thread , most forum users can also understand and answer in English.

I (Karl-Heinz Kübbeler) have continued the project *transistor* tests from Markus Frejek and mainly further developed the software. Due to its improved properties, the name *component testing was* suggested. I see for myself that its main purpose is to determine the type of transistor and its parameters, as developed by Markus Frejek.

The most important features are:

- Works with the microcontrollers ATmega8, ATmega168, ATmega328 or also with ATmega644, ATmega1284.
- Displays results on an output device (LCD) with 2x16 or 4x20 characters.
- It is also possible to use the graphics output devices with the controllers ST7565, NT7108 or ST7920. It is also possible to use OLED reading devices with controller SSD1306 and communication via the data bus interfaces SPI or I2C. It is also possible to use color graphics output devices with the controllers ILI9341 or ILI9163.
- One-key operation with automatic shut-off.
- The device has three measuring ports (test connections, (measuring) pins).
- Automatic detection of **NPN, PNP, N-channel and P-channel MOSFET, JFET, diodes** and small **thyristors, TRIAC** .
- Automatic detection of component legs, which means that the legs of the component can be connected to the measuring ports arbitrarily.
- Measurement of hFE (beta) and base-emitter voltage drop for bipolar transistors (BJT), incl. for Darlington transistors.
- Automatic detection of protection diodes in bipolar transistors and MOSFETs.
- Bipolar transistors are detected as a transistor with a parasitic transistor (NPNp = NPN + parasitic PNP).
- Up to two resistors can be measured with a resolution down to 0.1 ohm. The measuring range covers up to 50 Mohm (Megaohm). Resistors below 10 ohms are measured in the same way as an ESR measurement and with a resolution of 0.01 ohms if an ATmega168 / 328 is used. Note: Resolution is not accuracy .
- **Capacitors** capacitance in the range of 35pF (picofarads) to 100mF (millifarad) can be measured with a resolution down to 1 pF. Make sure that the capacitor is discharged before connecting to the measuring ports.
- If the processor has at least 32K flash memory, Pieter-Tjerk's sampling ADC method can be used to get a resolution down to 0.01 pF for capacitors with a capacitance lower than 100 pF.
- Resistors and capacitors will be displayed with their respective symbols, measuring ports and values.
- Up to two diodes will also be displayed with their correctly inverted symbols, measuring gates and voltage drops.
- If the component is a single diode, its parasitic capacitance will be measured - and from version 1.08k, its leakage current will also be measured.
- With ATmega168 / 328, self-calibration is possible for zero capacitance, zero resistance and other parameters.
- With ATmega168 / 328, coils can be detected and their **inductances** measured if in the range 0.01 mH to 20 H.
- If the processor has at least 32K flash memory, the samplingADC method can be used to measure smaller inductances with a parallel capacitor with known capacitance value. The resonant frequency, the calculated inductance value are displayed and the goodness.
- With ATmega168 / 328, a capacitor's ESR (*Equivalent Series Resistance*) can be measured for capacitances greater than 20 nF. The resolution is 0.01 Ohm. For lower capacitances, the

ESR accuracy deteriorates.
- With ATmega168 / 328 and capacitors above 5 nF, Vtab can be examined for charging pulses. Via this method, the goodness of the capacitor can be estimated.
- With ATmega328, a **menu function** can be reached with a long keystroke (> 0.5 second). A short key press switches to the next function. A long keystroke starts the function. Here is the list of built-in features so far:
    - Frequency measurement on port PD4. This port is also used for the reading device (LCD) and will be changed to input (high-Z) during the measurement. The frequency is measured over 1 second. If the frequency is below 25 kHz, mean time periods are measured instead of increasing the accuracy. The resolution drops to 0.001 mHz (milliHertz).
    - Voltage measurement on port PC3 if not used for serial output. Ds ATmega328 has 32 legs (PLCC), ADC6 or ADC7 can also be used. A 10: 1 voltage divider is used so that voltages up to 50 V can be measured. With an additional DC-DC converter, zener diodes can also be measured.
    - Frequency generation on port TP2. Above the 680 ohm resistor connected to port PB2, a signal with a selected frequency from 1 Hz to 2 MHz can be obtained from port TP2. Port TP1 is ground. (?: The German and English text sections could not be understood)
    - Variable PWM (*pulse width modulation*) with fixed frequency on port TP2. 10-bit counter. Port TP1 is ground. Short keystrokes increase the pulse width by 1%, long keystrokes by 10%.
    - There is an alternative possible method of measuring capacitance and ESR. Capacitances of 2 µF to 50 mF can usually be measured while the capacitor is in the circuit. Make sure that the capacitor is discharged before connecting to the measuring ports.

You can read detailed information with measurement examples in the PDF documentation in English and German. A Russian translation is also available. The links to the PDFs are in the download section of this page.

Introduction (French)

Original project: http://www.mikrocontroller.net/articles/AVR-Transistortester

Perfected by author Karl-Heinz Kübbeler, see the **accompanying discussion forum** .

I continued to develop Markus Frejek's project and more specifically the software. Based on the improved characteristics some have proposed to call it a component tester. Personally, however, I consider as an eminent property the automatic determination of the type and characteristics of transistors, as developed by Markus Frejek.

I would like to cite the most important **features** here :

- Use of ATmega8, ATmega168, ATmega328 or ATmega644 and ATmega1284 processors.

- Display of the results measured by a **2x16 or 4x20** character **LCD** display .
- Instead of a 2x16 character LCD display, you can also use a **graphic display** based on an ST7565, NT7108 or ST7920 controller. Connection of an **OLED** display to SSD1306 controller via SPI or I2C interface is possible. Color displays with ILI9341 or ILI9163 controller can also be used.
- One-touch operation with automatic timed shutdown.
- The device has three universal test ports (Pins Test TP1, TP2 and TP3).
- Automatic type determination of **NPN** and **PNP bipolar** transistors , N or P channel **MOSFETs** , **JFETs** , diodes as well as low power **thyristors and TRIACs** .
- Automatic determination of the component connection diagram, the components can be connected in any way.
- Measurement of current amplification factor (hfe) and base-emitter voltage of bipolar transistors, including Darlington transistors.
- Automatic detection of a protective diode integrated into bipolar transistors and MOSFETs.
- In some cases a parasitic transistor can be detected when testing transistors with protective diode (NPNp = NPN + parasitic PNP).
- Up to two **resistances** can be measured simultaneously with a resolution of 0.1 Ohm. The measuring range exceeds 50 MOhm. When using an ATmega168 / 328 processor resistances below 10 Ohm are measured by the ESR method (series resistance) with a resolution of 0.01 Ohm.
- The **capacitors** are measured in a range of 35 pF to 100 mF with a resolution of 1 pF.
- When the flash memory size is 32K, capacitors below 100 pF can be measured by Pieter-Tjerk's SamplingADC method with a resolution down to 0.01 pF.
- Resistors and capacitors are displayed with their symbol, surrounded by the number of connection terminals.
- The values of resistors and capacitors are displayed with 4 decimal digits in the correct dimension.
- Up to two diodes are also displayed with their symbol by observing the direction of current flow. The symbols are surrounded by the numbers of the connection terminals. The voltage threshold value is also displayed.
- In the case of detached diodes, the device also measures the values of the capacitance and (from version 1.08k) of the leakage current in the reverse direction.
- The ATmega168 / 328 processor provides a "self test" mode allowing a calibration of the capacitance respect of the no-load resistance as well as other parameters.
- The ATmega168 / 328 processor also allows the detection and measurement **of inductivities** greater than 0.01 mH up to more than 20 H.
- With a minimum Flash memory of 32 K it is possible, by means of the parallel connection of a capacitor of known capacity, to measure inductivities of low value by the SamplingADC method. In addition to the resonance frequency, the calculated value of the inductivity and the loss factor are displayed.
- The ATmega168 / 328 processor enables measurement by the ESR method (Equivalent Series Resistance) of capacitors of at least 20 nF with a resolution of 0.01 Ohm. Note, however, that

the accuracy of the results is lower for low capacitance values.

- The ATmega168 / 328 processor measures the voltage loss Vloss of capacitors greater than 5 nF by analyzing the voltage withstand after a load pulse. This makes it possible to estimate the loss factor of the capacitors.
- Additional functions are available with an ATmega328 processor. A **menu** can be activated by pressing the button for longer than 0.5 s. Special functions can then be chosen from a list. A short press displays the next function in the list. A long press starts the displayed function. Below are the additional functions implemented now:
    - **Frequency measurement** at pin PD4, used at the same time for the connection of the LCD display. For measurement, the pin is configured as an input. The applied frequency is first counted for one second. If the frequency is less than 25 kHz, the average period is measured. Based on the period, the frequency is calculated with a resolution of up to 0.001 mHz.
    - **Measurement of an** external **voltage** via the PC3 pin, on condition that this is not used as a serial output port. When using an ATmega328 in a 32-pin PLCC package, one of the ADC6 or ADC7 pins can be assigned to the voltage measurement. Since a 10: 1 voltage divider is provided, voltages from 0 to 50 V can be measured. An extension of the circuit (DC-DC converter) then makes it possible to measure Zener diodes.
    - **Frequency generator** at Port TP2. Via the 680 Ohm resistor connected to pin PB2, a signal with a variable frequency between 1 Hz and 2 MHz can be output via the TP2 port. The TP1 port is then connected to ground.
    - **Pulse** converter at port TP2 with fixed frequency and variable pulse width ratio. In this function, counter 1 is used as a 10-bit counter. The TP1 port is connected to ground. The pulse width can be increased by 1% with a short press of the key, and by 10% with a long press.
    - A variant of the capacitance and ESR measurement function enables capacitors from 2 µF to 50 mF to be measured in their circuit. To this end, they will be connected to the Test pins TP1 and TP3. It is particularly important that the capacitors thus measured no longer have any residual charge.

When the POWER_OFF option was activated at the level of the configuration file (Makefile), the additional functions as well as the dialog function itself are limited in time. For more detailed information see the documentation in pdf format (ttester.pdf), in German or English. A Russian translation of the documentation is also available.

Software (german)

The software was further developed based on the work of Markus F. The part for the capacitor measurement has been completely rewritten and the resistance measurement has also been significantly revised. In the event of difficulties or problems, I should be notified via e-mail or via the discussion section (thread). Hopefully only when I know about problems can I find a solution.

I have described further details as well as a description of the individual measurement methods and sample results in the PDF documentation (German and English version). Here you will also find information on configuring the software with Makefile parameters and options. The comments in the source code are in English. A self-test function is newly built into the software, in which the function of the tester is measured. A calibration part is also integrated in this self-test.

Software (english)

The software was developed based on the work of Mark F. The capacitor measurement was completely rewritten, and the resistance measurement substantially revised. If you have difficulties or problems, notify me via e-mail or the discussion section (thread); I can only help if I know about the problems.

For further details, descriptions of the measurement methods, and sample results, see the PDF documentation (German and English versions). It also contains information about configuring the software with Makefile parameters and options. The source code comments are in English.

The software has a new self-test function, which also does calibration.

Software (French)

The software was developed on the basis of the work of Markus F. The part concerning the measurement of capacitors has been completely rewritten and the measurement of resistances has been revised considerably. In case of difficulties or problems, please contact me by email or via the discussion forum. To be able to solve the problems I must first know them.

In my documentation in pdf format, in German or English language, I described additional details, the different measurement procedures as well as examples of results. There are also instructions for configuring the software using the parameters and options of the "Makefile" file. Comments in the source codes are in English.

I integrated in the software a new autotest function verifying the correct functioning of the tester. The automatic self-test also includes a calibration routine.

Hardware (German)

In principle, the new software has to be configured in such a way that it runs on the hardware already presented by Markus F. without any changes.

However, some changes are useful:

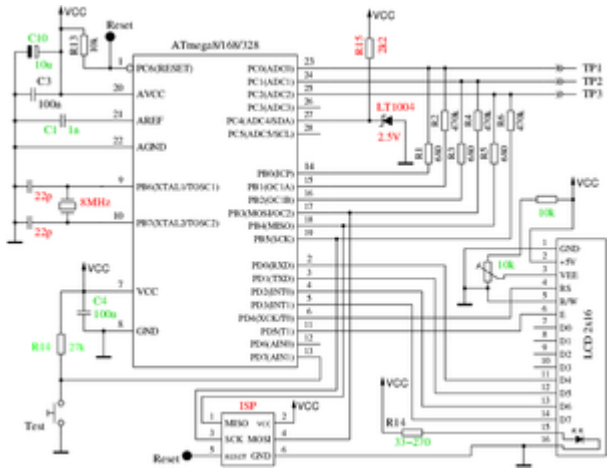
- The processor should be switched to an 8 MHz clock frequency, preferably with an external crystal. To do this, the fuses of the ATmega must be changed. A 16 MHz crystal can also be

used if the software in the Makefile has been adapted.

- A "pull up" resistor of about 27 kΩ should be retrofitted from pin 13 (PD7) of the ATmega to VCC.
- The 100 nF capacitor at pin 21 (AREF) can either be removed completely or, better, replaced by a 1 nF capacitor.
- If the electronic activation of the tester causes problems, at least the C2 capacitor at the base of transistor T1 should be reduced to 10 nF and, if necessary, the resistor R7 should also be reduced to 3.3 kΩ. The complete circuit diagram and details can be found in the PDF documentation.

The reasons and the details for these changes as well as further information for a rebuild are described in the hardware chapter of my pdf documentation. An ATmega168 processor or an ATmega328 processor is recommended because the ADC with the autoscale function can be switched from the 5V reference (VCC) to the internal reference voltage if necessary. The internal reference has a voltage of 2.56V for the ATmega8, but 1.1 volts for the other processors. With 1.1 V, a better resolution of the ADC can be achieved for measured voltages below 1 volt. You can exchange the ATmega8 for an ATmega168 or ATmega328 without changing the hardware! Here is the part of the circuit that is required for the measurement.

The electronics for the battery supply and the automatic switch-off are missing in this circuit diagram.



Circuit diagram without power supply

The components marked in red are not absolutely necessary, but can help improve the measurement accuracy. The components marked in green have been changed compared to the first draft by Markus F. The Eagle files from Asko B. for three variants can be found in the thread at the address: http://www.mikrocontroller.net/topic/248078?page=4#2891344

Here you can find the article of the 1st transistor tester version by Markus F.: AVR transistor tester

Hardware (English)

The new software can be configured to run without any changes on the hardware developed by Markus F.

But a few modifications still make sense:

- The processor clock should run with 8 MHz, preferably with a external quartz. To this purpose the fuses have to be set. A 16 MHz quartz may also be used if the software is adapted through the Makefile option.
- A pull up resistor of 27 kΩ should be added between pin 13 (PD7) of the ATmega and VCC.
- The 100 nF capacitor at pin 21 (AREF) should be removed or even better be replaced with a 1 nF one.
- If the tester turns on unreliably, the capacitor C2 at the base of transistor T1 should be decreased to 10 nF. Where necessary resisitor R7 should be decreased to 3.3 kΩ. The circuit diagram and further detail is to be found in the PDF documentation.

The reasons and details concerning these changes as well as further hints about new implementations are explained in the hardware section of my PDF documentation. ATmega168 or ATmega328 processors are recommended, because the ADC auto-scale function allows to switch from the 5V reference to the 1.1V internal reference. The ATmega8 has a 2.56V internal reference which is inferior for measurements below 1V. The ATmega8 can be replaced by a ATmega168/328 without changes to the hardware. Here is the part from the circuit diagram that is responsible for the measurements.

The circuits for the battery supply and the automatic shutdown are not shown by this circuit diagram.

You could go without the components marked in red, but they may enhance the precision of the measurements. Those marked in green are modifications to the original design by Markus F. The Eagle CAD files by Asko B. for three variants can be found in the discussion thread at http://www.mikrocontroller.net/topic/248078?page=4#2891344

This is the article about the first version of the transistor tester by Markus F.: AVR-Transistortester

Hardware (French)

In principle the software can be configured to run without modifications on the hardware presented by Markus F. (see below).

Some modifications are however useful:

- The processor should be driven by an 8 MHz clock, preferably with an external crystal. To this end, the fuses ("fuses") of the ATmega processor must be modified. A 16 MHz crystal can be used provided that the software is configured accordingly via the Makefile.
- A pull up resistor of about 27 kΩ should be added between pin 13 (PD7) of the ATmega and the VCC power supply.

- The 100 nF capacitor at pin 21 (AREF) can be removed or, better, replaced with a 1 nF capacitor.
- If the electronic start of the tester causes a problem, it is necessary to at least reduce the value of the capacitor C2 at the base of the transistor T1 to 10 nF and, if necessary, reduce the value of the resistor R7 to 3.3 kΩ. The complete diagram and details in this regard can be found in the pdf documentation.

The reasons for these modifications as well as additional indications are detailed in the "Hardware" chapter of my pdf documentation. The use of an ATmega168 or ATmega328 processor is recommended, because if necessary the "auto-scale" function of the analog-to-digital converter (ADC) switches from the 5 V reference (VCC) to the reference voltage. internal. The internal benchmark of ATmega8 is 2.56 V, while that of other processors is 1.1 Volt. With 1.1 V a better resolution of the ADC converter is achieved when measuring voltages below 1 Volt. The ATmega8 processor can be replaced by an ATmega168 or ATmega328 without any modification of the tester schematic! Here is the part of the diagram responsible for the measurements.

The circuits for battery power and automatic shutdown are not shown in this diagram.

Components marked in red are not required, but can help improve measurement accuracy. The components marked in green are changed from the original project by Markus F. CAD files in Eagle format for three variants made available by Asko B. can be found in the discussion thread under the address: Mikrocontroller.net/topic/ 248078? Page = 4 # 2891344

Here is the article about the first version of the transistor tester by Markus F.: AVR-Transistortester

Downloads (German)

The current version of software and documentation is now available at GitHub.com/Mikrocontroller-net/transistortester .

The documentation is available in German and English and in Russian and Czech .

The users can use the command "git clone https://github.com/Mikrocontroller-net/transistortester " to download a complete copy of the archive into a newly created transistortester directory. In the working directory transistortester you can bring the copy up to date with "git checkout".

Downloads (russisch) - Downloads (russian)

All versions of the software and documentation stored at GitHub.com/Mikrocontroller-net/transistortester are available for downloads .

ttester.pdf instructions (Russian) Version 2023-03-22

~~The user can download the selected directory as a "GNU archive" via svnbrowser https://www.mikrocontroller.net/svnbrowser/transistortester/ .~~

Downloads (you)

You can get the most up-to-date versions of software and documentation (english/ttester.pdf) now at GitHub.

The documentation is also available in german and russian and czech.

Users can download the full archiv with "git clone https://github.com/Mikrocontroller-net/transistortester" into a new created transistortester directory. You can update your local copy in the working directory transistortester with the command "git checkout".

Downloads (English)

The current version of the software and documentation can be downloaded now from the github repository: GitHub

The documentary is available in German , English , Russian and Czech .

Users can download a full copy of the archive with the command "git clone https://github.com/Mikrocontroller-net/transistortester " to a newly created local directory *transistortester* . By positioning the working directory on the *transistortester* directory the copy can be updated using the "git checkout" command.

Downloads (Portuguese - Brazil)

All software and documentation versions are saved in the GitHub.com/Mikrocontroller-net/transistortester archiver .

~~Users can download a "GNU" package from all previous directories with svnbrowser https://www.mikrocontroller.net/svnbrowser/transistortester/ .~~

Downloads (Español)

All versions of the software and documentation are at GitHub.com/Mikrocontroller-net/transistortester .

~~Users can download a "GNU tarball" from the selected directory using svnbrowser https://www.mikrocontroller.net/svnbrowser/transistortester/~~

Downloads (Slovak)

All software versions and documentation are stored in GitHub.com/Mikrocontroller-net/transistortester .

~~Through *svnbrowser* , located at https://www.mikrocontroller.net/svnbrowser/transistortester/, it is possible to download the complete contents of the currently displayed directory by clicking on the *"Download GNU tarball"* link .~~

Downloads (Čeština)

The latest software and documentation version (english / ttester.pdf) is now available at GitHub.com/microcontroller-net/transistortester .

Checking professional Czech: Ing. Frantisek Prochazka & Milan Petko.

Users can use the "git clone https://github.com/Mikrocontroller-net/transistortester " command . download a complete copy of the archive to the newly created transistortester directory. You can use the "git checkout" command in the transistortester working directory to update the copy.

Download (Chinese)

All documents and software can be found on ~~SVN~~ GitHub.

Brief description (English version) 1.11k (2015-10-09)

Manual (English version) 1.11k (2015-02-08)

(English version) 1.13k (2021-03-20)

Method 1 Enter the directory you want to download in the GitHub browser], click to download the compressed package of this directory. Use your favorite compression software to decompress this compressed package, and you will be able to get the file you want.

Downloads (in your-language)

Feel free to put a translation *here*, but only if its done by yourself, not Google Translate. You can also put a translation of the whole article here, if it is done by yourself.

Only little understanding of the Wiki-Syntax is needed.

Hint to Cloners and Sellers 中文

Dear Transistortester Cloners and Sellers!

We don't mind if you produce and sell clones of the Transistortester. It provides an inexpensive great little tool for electronics enthusiasts and beginners, but PLEASE note the links to the project's webpage, source repo and documentation. You would add more value by giving users that information to be able to update the firmware and to understand all the features. If you do any modifications to the firmware, please send us a copy for the repo. And if you would send us your Transistortester clones, we would be able to keep the firmware as compatible as possible. Don't forget, this is an OSHW project!

Best regards, Transistortester team

Dear manufacturers and sellers of transistor tester replicas:

If you produce and sell copies of transistor testers, we won't mind. It can provide a cheap gadget for electronic enthusiasts and beginners, but please pay attention to provide the link, source code and documentation of the project web page when selling. Through the link, provide users with information to update the firmware and understand all functions to increase the value of the product. If you make any changes to the firmware, please send us a backup. If you send us samples of the transistor tester, we will be able to keep the firmware as compatible as possible. Don't forget, this is an OSHW (Open Source Hardware) project!

Best wishes, Transistor Test Team

TableClonesEn.pdf - List of clones

Directory structure of the Git archive

Folder structure and description of the *paths* in the SVN

Ordner/directory		Files / files	Description / description
Tissue			Contains the documentation as PDF and as pdflatex source text
	trunk		Last development status of the documentation including pictures and diagrams
	trunk/pdftex/german		contains the German texts, Makefile and PDF documentation of the developer version
	trunk/pdftex/english		contains the English text, Makefile and PDF documentation of the developer version
	trunk/pdftex/russian		contains the Russian text, Makefile and PDF documentation of the developer version
	tags	*changel*	*Every change should be entered here with the version number*

		og.txt	
	tags/german		*Current PDF documentation in German*
	tags/english		*Current PDF documentation in English*
	tags/russian		*Current PDF documentation in Russian language*
	tags/old/german		*PDF documentation for previous software versions*
	tags/old/english		*PDF documentation for earlier software versions*
Hardware			Hardware description
	strip_grid		Directory for a strip circuit board
	strip_grid/ttester_strip_grid.diy		Example of a stripline board, DIYLC file
	strip_grid/TTester_strip.pdf		Result of the strip circuit board in PDF format
	strip_grid/LiesMich.txt		Short documentation for stripline board
	strip_grid/ReadMe.txt		Short documentation for the strip grid board
	Markus		Design by Markus R. with LED dimmer in Eagle 6.4.0 format
Software			Software for AVR-GCC 4.8.2
	trunk		Current software development branch
	trunk/default		Makefile and programming data for ATmega168 with standard layout
	trunk/mega168_1.9V		Makefile and data for ATmega168 with button cell operation (FiFi)
	trunk/mega168_3.3V		Makefile and data for ATmega168 with LiPo battery operation (FiFi)
	trunk/mega		Makefile and data for ATmega168 for stripline board

	168_strip_grid		
	trunk/mega328		Makefile and data for ATmega328 with standard layout (from version 1.08k)
	trunk/mega328_1.9V		Makefile and data for ATmega328 with button cell operation (radio amateur)
	trunk/mega328_3.3V		Makefile and data for ATmega328 with LiPo battery operation (radio amateur)
	trunk/mega328_2X16_menu		Makefile and data for ATmega328, 2x16 characters text display, pulse encoder + voltage measurement
	trunk/mega328_dogm		Makefile and data for ATmega328 with standard layout, 2x16 characters DOG-M LCD
	trunk/mega328_strip_grid		Makefile and data for ATmega328 for stripline board (from version 1.08k)
	trunk/mega328_strip_grid_dogm		Makefile and data for ATmega328 for stripline board with DOG-M display
	trunk/mega328_st7565		Makefile and data for ATmega328 with standard layout, 126x64 pixel LCD, ST7565 controller
	trunk/mega328_st7108		Makefile and data for ATmega328 with standard layout, 126x64 pixel LCD, ST7108 controller
	trunk/mega328_st7920		Makefile and data for ATmega328 with standard layout, 126x64 pixel LCD, ST7920 controller
	trunk/mega328_fish8840		Makefile and data for Chinese Fish8840 version, ATmega328, 126x64 pixel LCD, ST7565 controller
	trunk/mega328_wei_st7565		Makefile and data for Chinese WEI_M8_LGTST version, 126x64 pixel LCD, ST7565 controller, LiIon Accu
	trunk/mega328_GM328		Makefile and data for Chinese GM328 version, ATmega328, 126x64 pixel LCD, ST7565 controller
	trunk/mega328_T3_T4_st7565		Makefile and data for Chinese T3 or T4 version, ATmega328, 126x64 pixel LCD, ST7565 controller
	trunk/mega328_T5_st7565		Makefile and data for Chinese T5 version, ATmega328, 126x64 pixel LCD, ST7565 controller, LiIon Accu

	trunk/mega328_ssd1306I2C		Makefile and data for ATmega328 with standard layout, 126x64 pixel OLED, SSD1306 controller, I2C interface
	trunk/mega328_ssd1306SPI		Makefile and data for ATmega328 with standard layout, 126x64 pixel OLED, SSD1306 controller, SPI interface
	trunk/mega644_LCD2004		Makefile and data for ATmega644 / 1284 with 4x20 characters LCD
	trunk/arduino_m2560		Makefile and data for Arduino Mega (ATmega2560) with 2x16 characters LCD
	trunk/mega8		Makefile and data for ATmega8. From version 1.00k the self-test for the ATmega8 can no longer be configured.
	tags		Finished software versions packed as ZIP
		changelog.txt	*Every change should be entered here with the version number*
	Markus		Alternative software from Markus R., please refer to README! The software has been cleaned up and is much better structured, but only runs on an ATmega168 or ATmega328. The software only runs on the standard layout.
	Measurement technician		Remote control function for the "m" version (Markus).

Category :

- AVR-Projekte