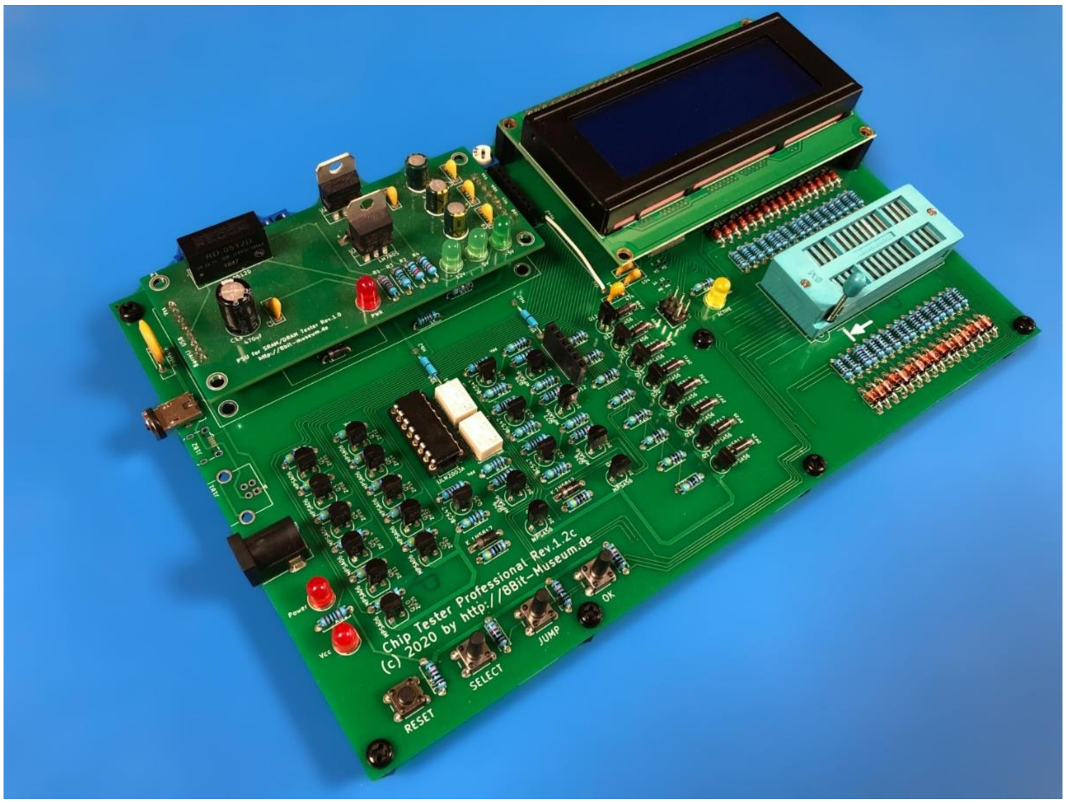
Version: 2024-09-21

Retro Chip Tester Professional

ASSEMBLY, PROGRAMMING AND TROUBLESHOOTING


Chip Tester Professional Rev.1.2c
(c) 2020 by http://8Bit-Museum.de
RESET
SELECT
JUMP
OK
Power
Vcc

Rev.1.2

Note:

This manual contains explanations of features that
will be available in the next firmware release.

This page is intentionally left blank

Table of Contents

Safety notes:

Please take note of the following safety notes for assembly:

- When soldering, the soldering iron, solder and also the components being soldered become hot.
- Since soldering produces toxic fumes, ensure good ventilation.
- Observe the warnings in the instructions for your soldering station or soldering iron.
- When assembling and screwing the individual parts together, pay attention to sharp edges on tools, screws and components.
- The RCT with its extensions is very complex and not suitable for children.

Please note the following instructions for operation:

- Never leave the device unattended when in operational mode.
- When the device is not in use, do not leave it connected to the power supply.
- If you notice an unusual odor or heat development on the device, switch off the device immediately.
- Depending on the load, the linear regulators (7805, 7905) can become hot over longer periods of use.
- Make sure that the device is used in a location out of the reach of direct heat sources, open flames and flammable materials and liquids. The device is designed for operation at room temperature.
- The device is designed for USB operation and a stabilized supply voltage of maximum 9V.

The memory tester has already been built several times and works for the developer. However, this does not necessarily mean that this is also automatically the case with a reproduction. The intention of this document is to explain the structure and the function and to provide information how the memory tester can be built. Additional technical information is often given. Not everything is absolutely necessary for understanding. However, these can be helpful in case of problems.

Anyone who builds the tester is responsible for the correct selection of the components and the assembly. There is no guarantee of correct functionality. No liability will be assumed if devices involved are damaged by using the memory tester. In this context, please also read the warnings in the user manual.

Some sections are marked with a "DANGER". This does not really mean that the content is "dangerous", but that the function could damage the Tester.

DANGER

Warning notices are marked accordingly.

This page is intentionally left blank

Essential tools and materials

To build the memory tester, you need:

- A soldering iron / soldering station
- Soldering tin and suitable flux
- Thin desoldering braid
- A 6 pin. ISP programmer (e.g. "Diamex USB ISP programmer for Atmel AVR, Rev.2" or "Diamex USB ISP programmer for AVR, STM32, LPC-Cortex (Prog-S2)") for programming the firmware. Cost: less than 20 EUR. The well-known USBASP is not recommended!

If the ATmega2560 is not already pre-assembled, there are some good “learning videos” for soldering the chip in the TQFN-100 package:

- https://www.youtube.com/watch?v=5uiroWBkdFY
- https://www.youtube.com/watch?v=nyele3CIs-U
- https://www.youtube.com/watch?v=YUryJOAiPa4
- https://www.youtube.com/watch?v=IlPAJLaG1BQ
- https://www.youtube.com/watch?v=BvhE16vBfX4
- https://www.youtube.com/watch?v=Cww1ZGKClXw

Copyright and Limited Warranty Statement

© Software and Hardware: Stephan Slabihoud, 8Bit-Museum.de

All rights reserved. No part of this manual, including the products and software described in it, may be reproduced, transmitted or commercially used in any form or by any means without the express written permission of the developer.

The RCT printed circuit board (equipped with the ATmega2560 or unequipped without any components (“blank board”)), the firmware, and all additional printed circuits boards (e.g. adapter boards) are referred to as "products" in the following.

Use of the products with its hardware and software is at your own risk. No liability is accepted for any damage that may arise when using the software and hardware. No guarantee is given that the testing functions are accurate and without faults. There is also no entitlement to a bug fix, even if the developer tries to fix bugs as quickly as possible.

If the products are bought outside the EU this warranty lasts for one year from the date of the sale. If the products are bought in the EU this warranty lasts for two years for from the date of the sale. Any additional components that must be purchased by the user in order to build the working RCT are not covered by this warranty. Likewise, no guarantee is given for the fully assembled RCT, except for the delivered printed circuits boards.

The developer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the user, including improper installation or testing or when the product has been altered or modified in any way. This limited warranty is the end-user’s sole and exclusive remedy against the developer where permitted by law. The products are provided “as is” and “with all faults”. The developer disclaims all other warranties, express or implied, of merchantability for a particular purpose.

The products are designed to test supported ICs in old home computers, synthesizers, arcade and pinball machines. Only non-commercial use in a private context is intended. The products are not authorized for use in safety-critical situations where a failure of the product would reasonably be expected to cause severe personal injury or death in any form. The user acknowledges and agrees that any other use of these products is solely at the users’ risk, and that the user is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

Consequential Damages Waiver. In no event shall the developer be liable to the user or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the developer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

1 General information about the Retro Chip Tester Professional

Whoever sets up the Tester should first get an overview of the now quite extensive documentation.

EN-1 – Assembly, programming and troubleshooting

The document discusses the assembly, programming and troubleshooting.

There are also additional files:

- Folder “Programming Step 1 - Set Fuses“
- Folder “Programming Step 2 - Flash Firmware“

Everything is there to program the ATmega. It is usually not necessary to load additional software from the Internet.

EN-2 – User Manual

This document discusses the usage of the RCT.

You will also find

- CRC32 lists (used to identify ROMs),
- Custom IC definitions (used to define very special ICs),
- Comparison Lists (used to find the correct test settings)

in the sub-folders where this document is stored.

EN-3 – Bill of Materials (BOM)

The document contains an overview of the required components and references to prepared shopping baskets. The same directory also contains component lists for Reichelt Electronics and Digikey. There are also interactive assembly layouts there.

2 Assembly of the PCB

The parts lists of the individual versions of the boards are in a separate document. The components are usually all labeled:
e.g. „104“ = 100nF, „224“ = 220nF, „22“ = 22pF, „4K7“ = 4700 Ohm, „4.7“ = 4.7 Ohm.

Depending on the version of the board, the assembly can vary slightly.

Take your time to assemble the board!

Plan at least 4 to 6 hours!

Make sure there is good ventilation when you populate the board

or use an extractor for soldering fumes.

Please wear protective goggles when cutting off wire ends.

The following step is not necessary if you have a pre-assembled circuit board:

The ATmega2560 should be soldered first. There is no clear recommendation: Some use an oven or hot air dryer with solder paste, others use a lot of flux and a hollow cone soldering tip (my preferred variant). Superfluous solder and short circuits are eliminated with flux and SMD de-soldering braid. Be sure to use a multimeter set to continuity or 'beep test' to check if there are short circuits or solder bridges. Especially check all solder joints on the ATMEGA 2560 microcontroller using magnification. If you power on the Tester and a short circuit is present it can damage it.

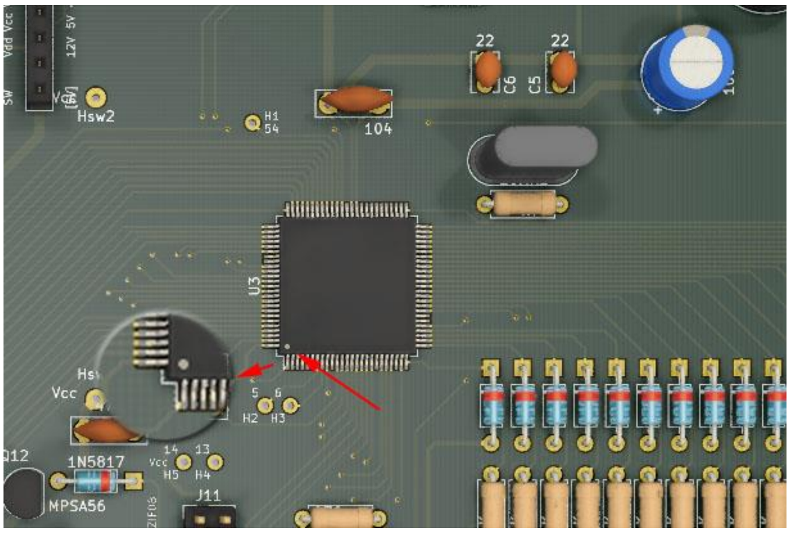


Figure 2.1: Orientation of the ATmega2560

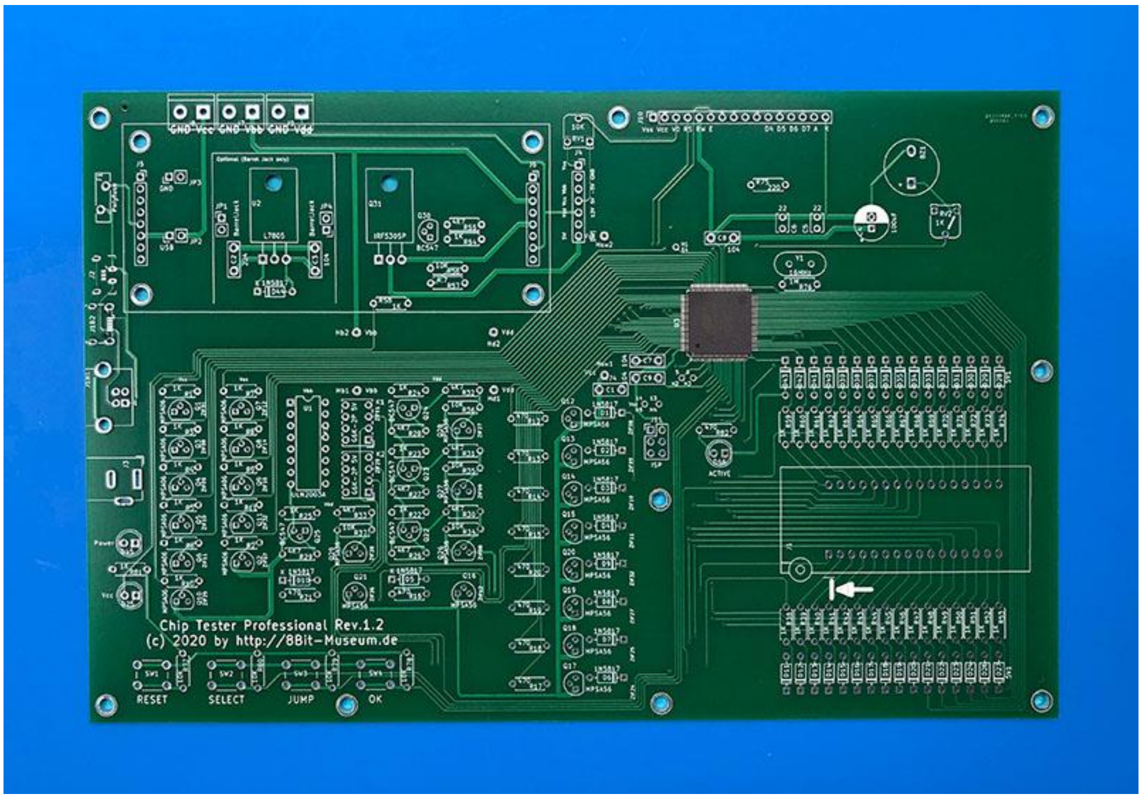

Chip Tester Professional Rev.1.2
(c) 2020 by http://8Bit-Museum.de
RESET
SELECT
JUMP
OK

Figure 2.2: PCB equipped with the ATmega2560

Then all resistors, the USB socket, the diodes and Z-diodes should be soldered.

Do not try to solder all resistors or diodes at once, as they are very close together and you still need space for the soldering iron. It is best to plan three or four passes.

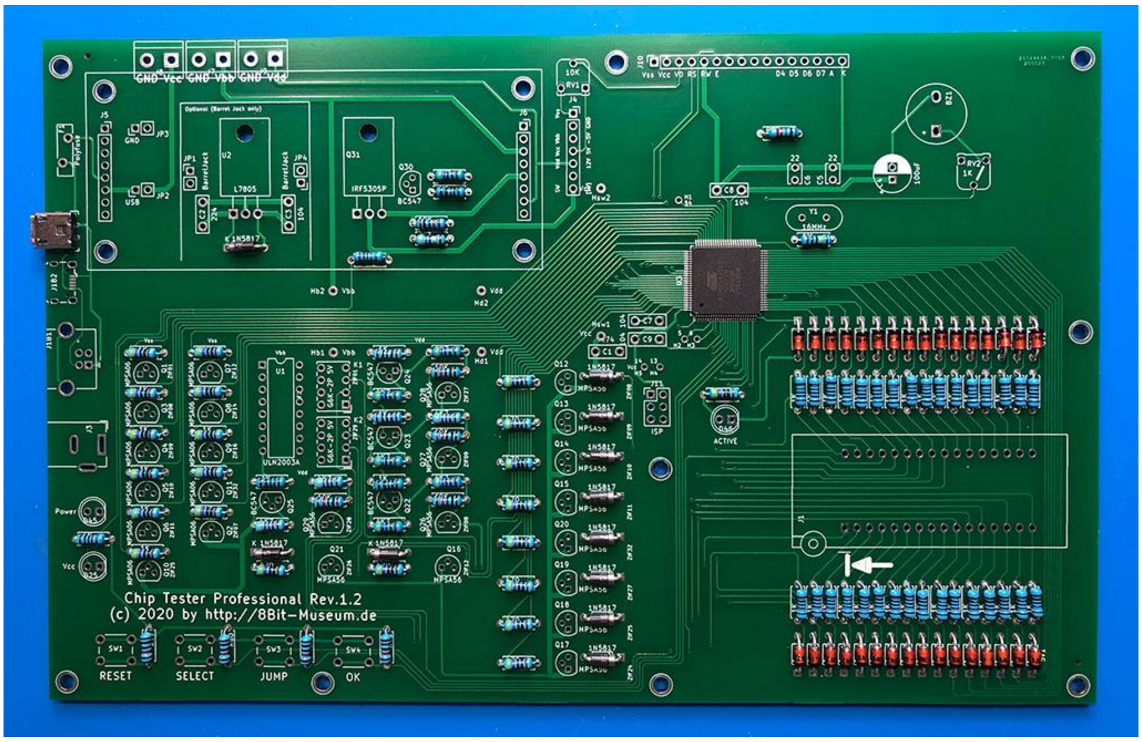

Chip Tester Professional Rev.1.2
(c) 2020 by http://8Bit-Museum.de
RESET
SELECT
JUMP
OK

Figure 2.3: PCB equipped with Z-diodes and resistors

Next is the voltage regulator, MOSFET, the two relays, the 16-pin socket and the three wire jumpers. Attention: To the right of the MOSFET at location R57 a 4.7 Ohm resistor is fitted, not a 4.7k Ohm resistor!

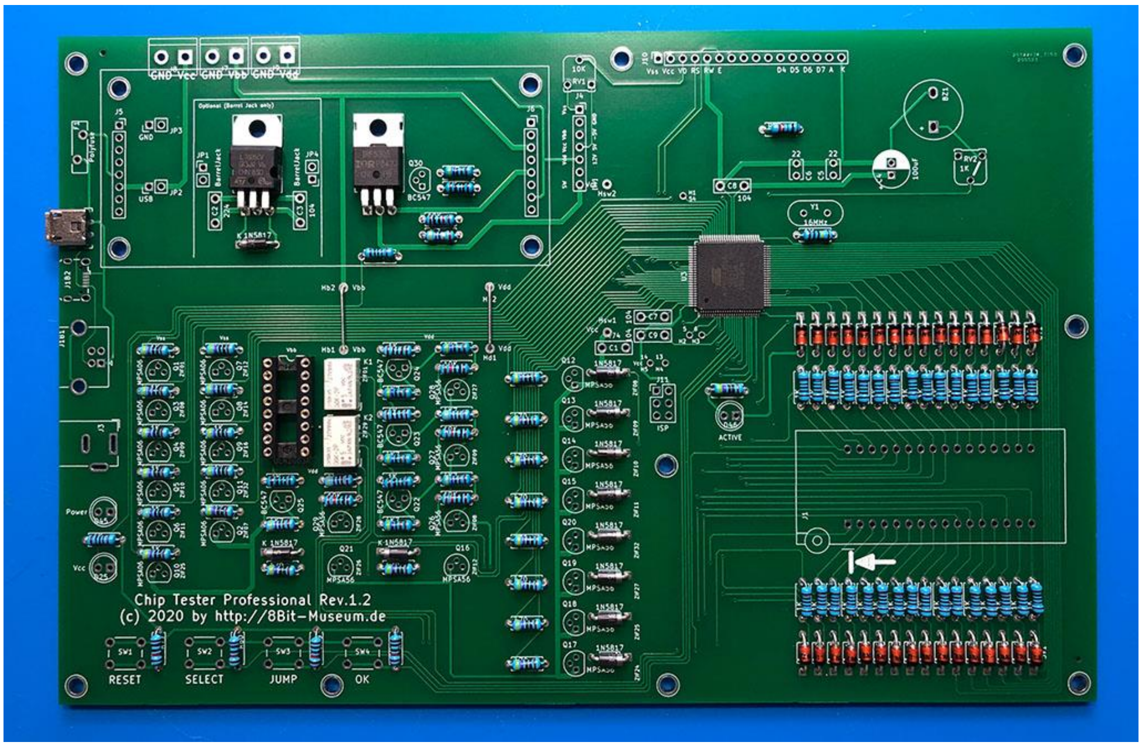

Chip Tester Professional Rev.1.2
(c) 2020 by http://8Bit-Museum.de
RESET
SELECT
JUMP
OK

Figure 2.4: PCB equipped with relays and voltage regulators

Please check the orientation of the socket and the relays carefully:

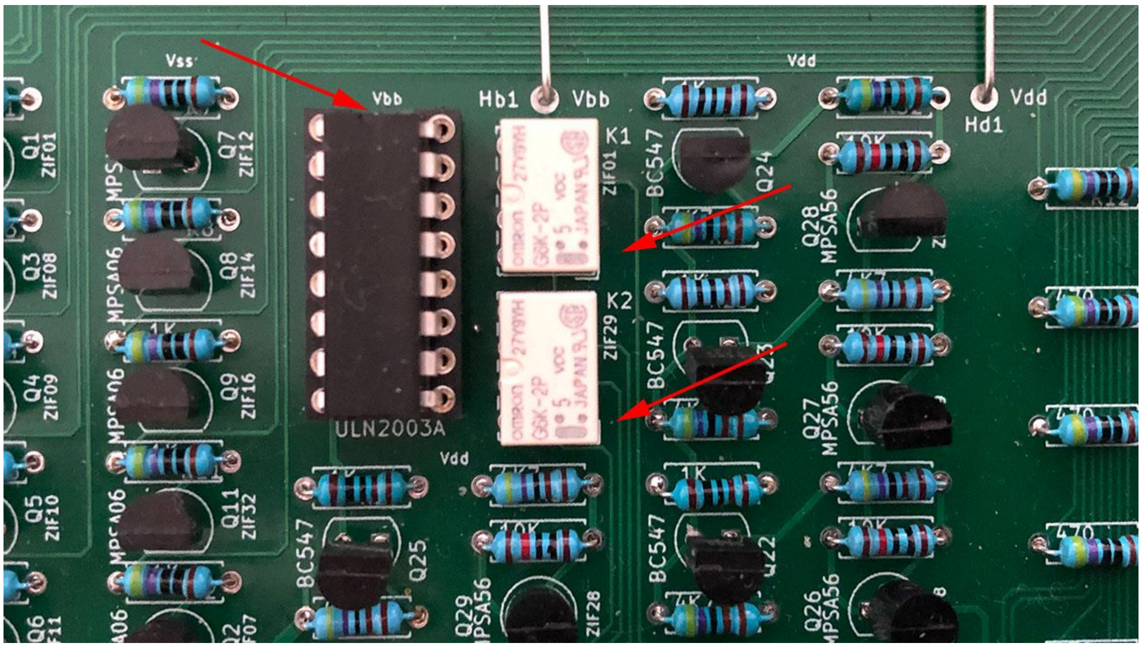


Figure 2.5: Detail of PCB equipped with relays and IC.

The wire jumpers again in detail:

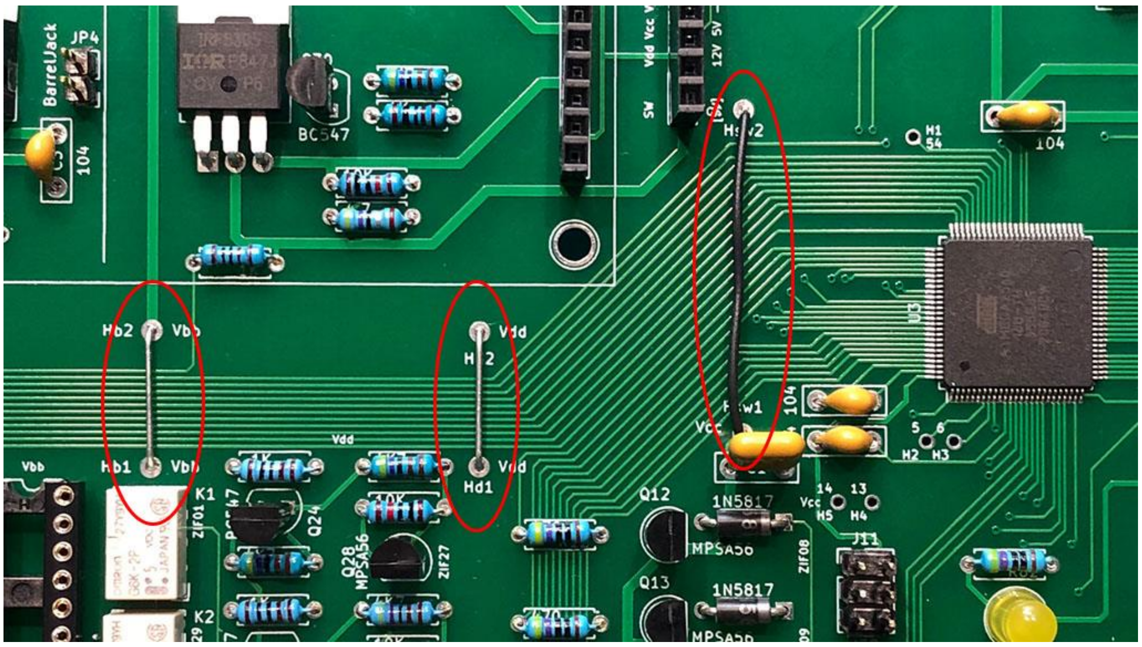


Figure 2.6: Wire jumpers

Now solder the transistors. Please do not use too much solder because the connections are very close together. Please be careful when cutting that there are no short circuits!

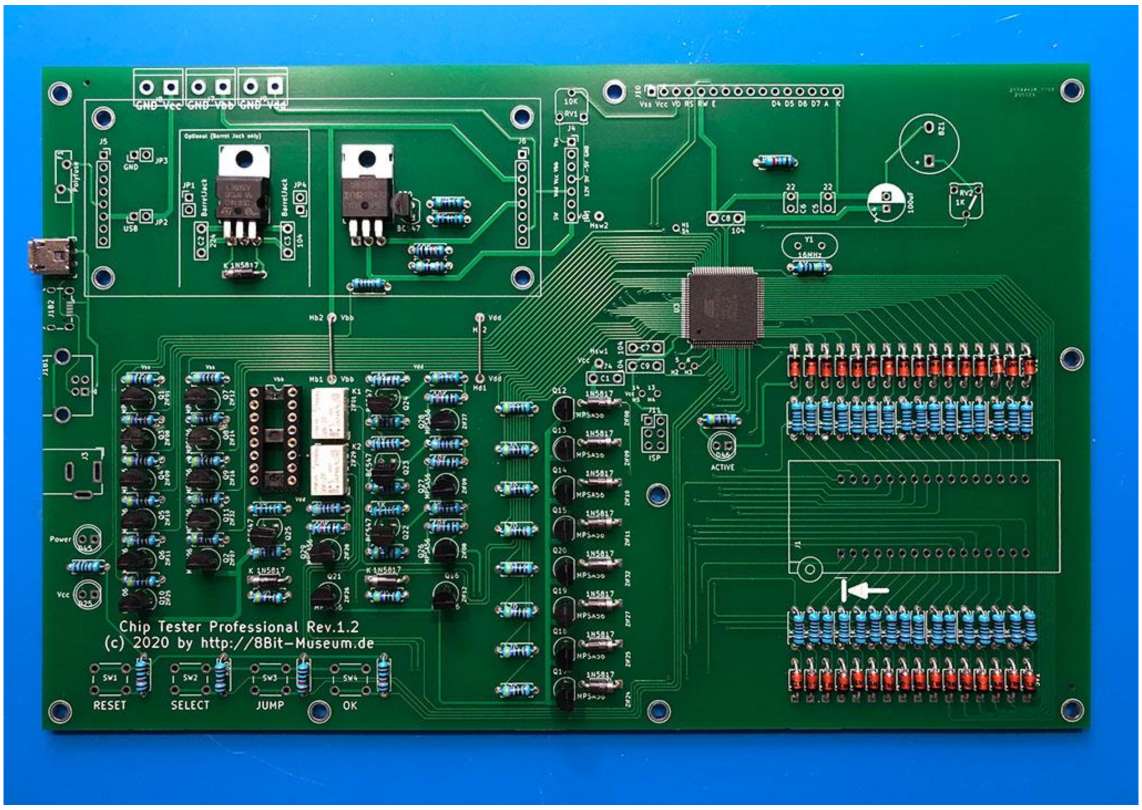

Chip Tester Professional Rev.1.2
(c) 2020 by http://8Bit-Museum.de
RESET
SELECT
JUMP
OK

Figure 2.7: PCB equipped with the transistors

Finally, the other components are assembled. The following can be used as a sequence: LEDs, pin headers, potentiometer, electrolytic capacitors (install horizontally) and capacitors, polyfuses, crystal oscillator, spacers, the piezo loudspeaker, tactile switches and the ZIF socket. Optionally, the terminal blocks, additional USB headers and the barrel connector can be soldered.

If a shrouded header is used, check the orientation:

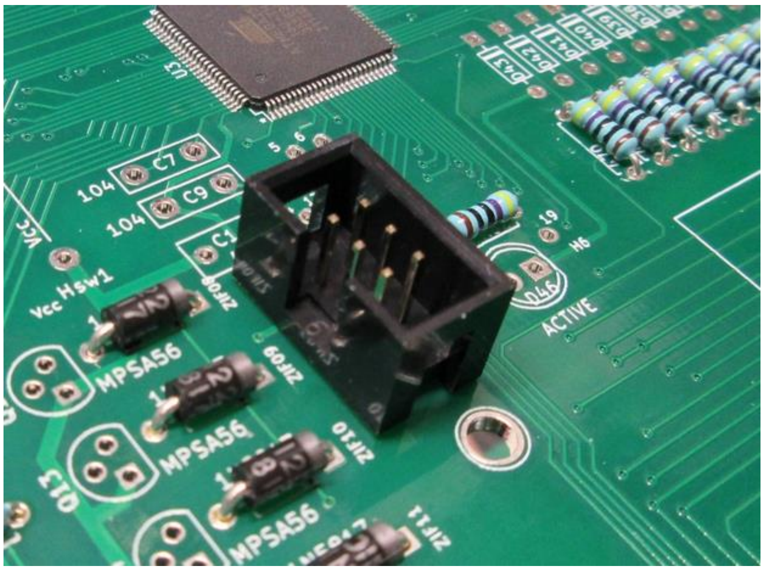


Figure 2.8: Shrouded Header (orientation)

The assembled board looks like this:

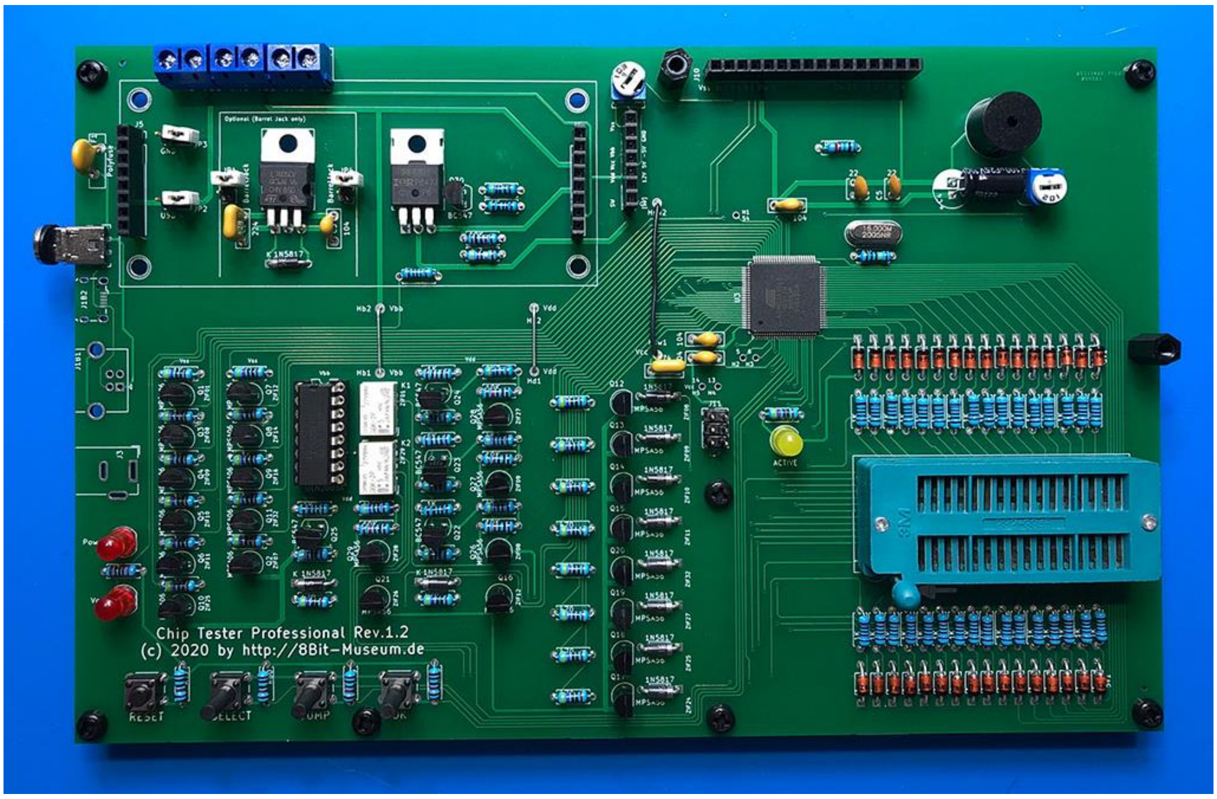

Chip Tester Professional Rev.1.2
(c) 2020 by http://8Bit-Museum.de
RESET
SELECT
ACTIVE

Figure 2.9: The assembled board

The fully assembled circuit board with display and DC/DC module look as follows:

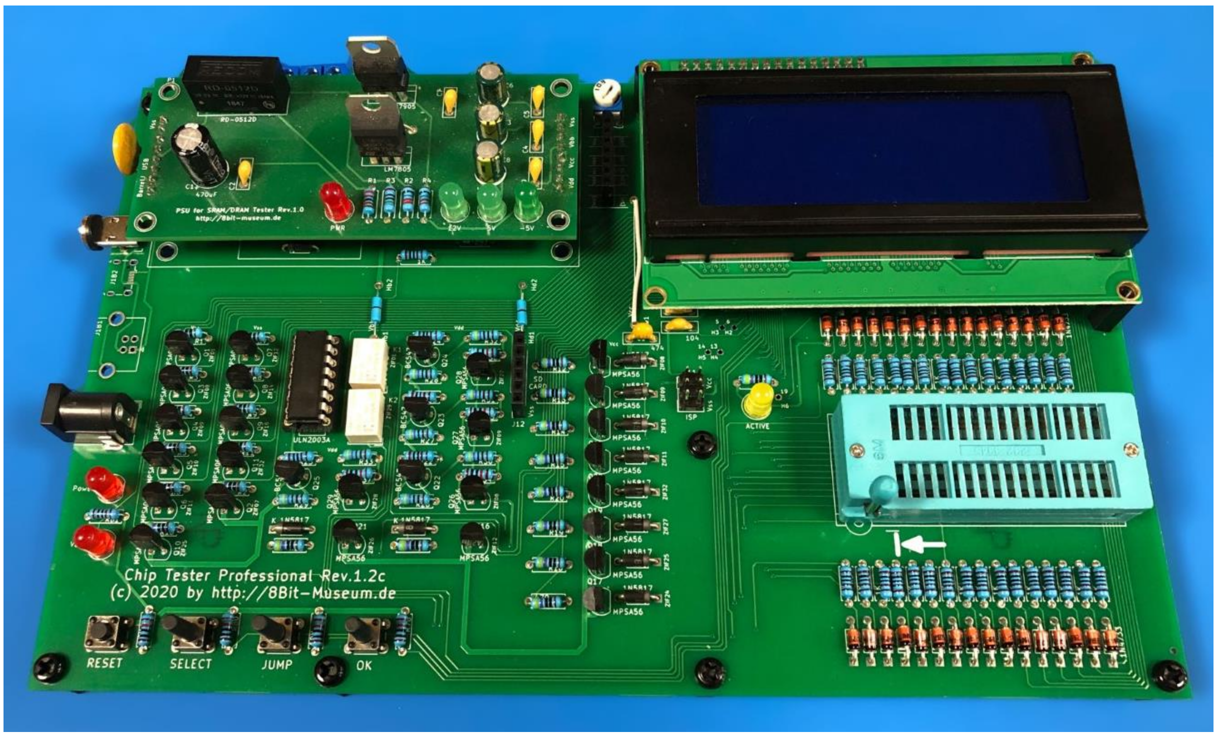

PSU for SRAM/DRAM Tester Rev.1.0
Chip Tester Professional Rev.1.2c
(c) 2020 by http://8Bit-Museum.de
RESET
SELECT
JUMP
OK
ACTIVE
ULN2003A

Figure 2.10: The fully assembled PCB

Depending on the version of the board, the assembly can vary slightly.

From revision 1.2h there is another footprint (Q32, SOT-23) on the board. If the MOSFET in the TO-220 package (Q31) is not available, Q32 in a SOT-23 package can alternatively be fitted.

Spacers should be used so that the board has a good stand and the display has a good grip. The following pictures show how the spacers can be installed.

The hex spacers for the feet should be 5-10mm long:

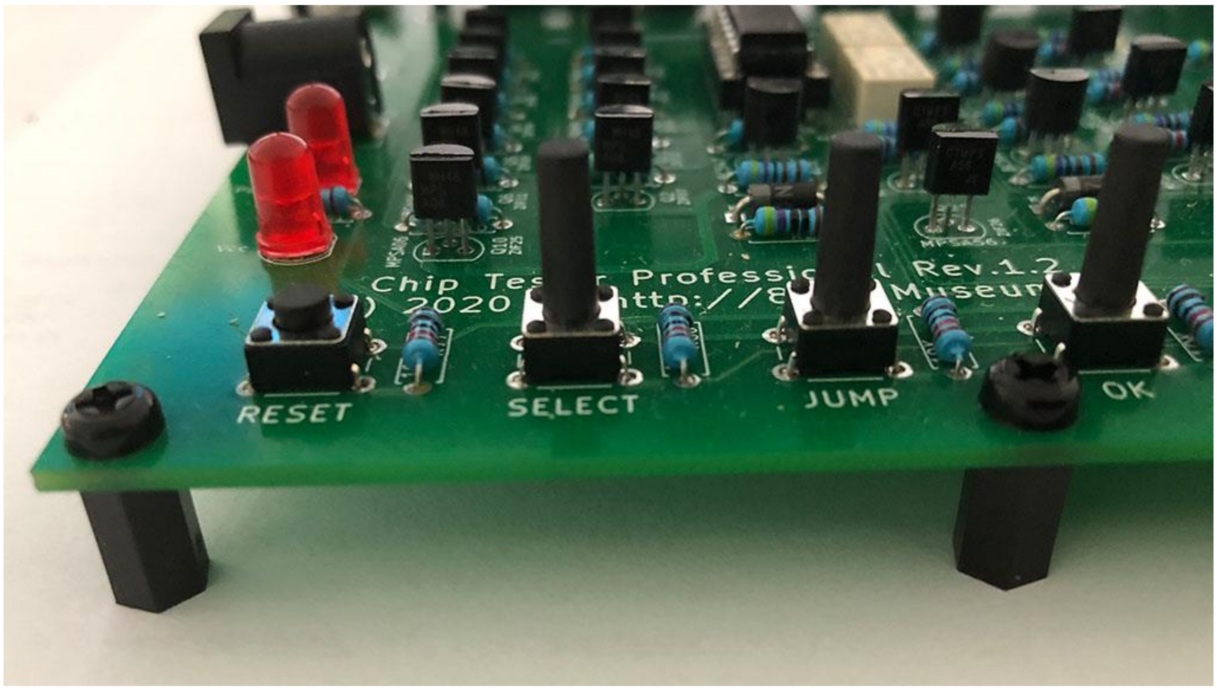


Figure 2.11: Assembly of feet

The hex spacers for the display should be 11mm long:

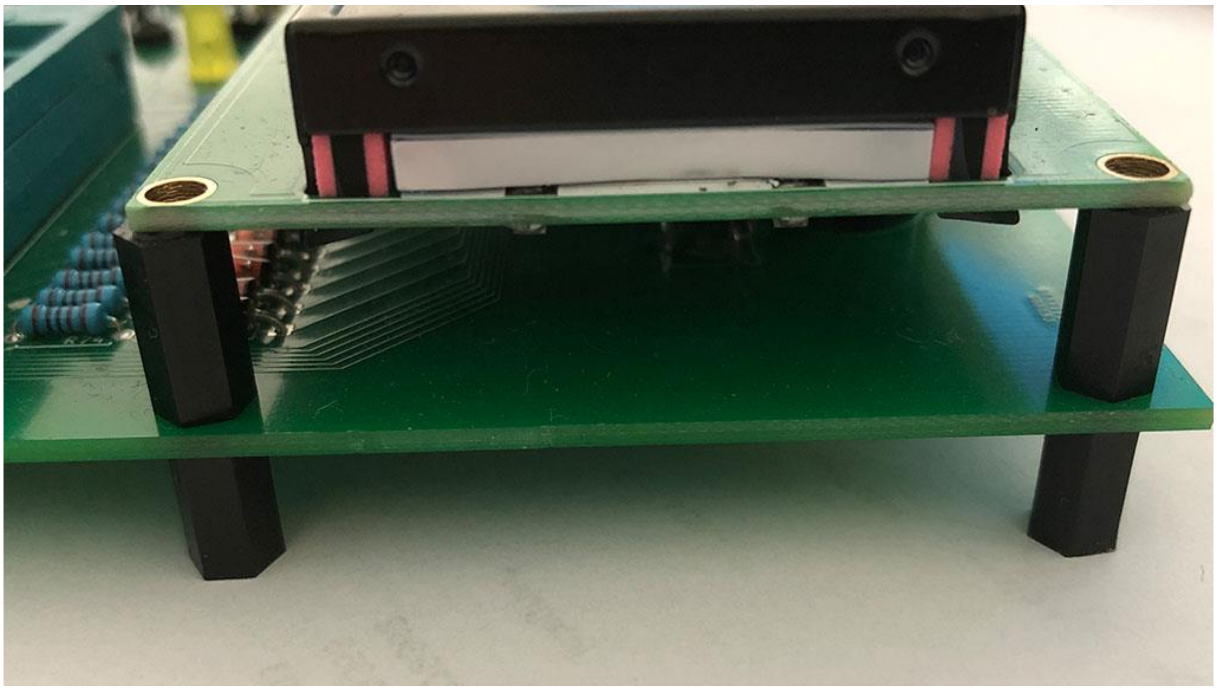


Figure 2.12: Assembly of the display

The holes have a diameter of just over 3mm (so suitable for M3 screws). The hex spacers should be made of plastic so that no scratches can occur on the table top.

3 Power supply

Not all components need to be assembled. The table shows the possible assembly variants (red = assembly required, green = assembly can be omitted). Of course, the board can also be fully equipped, jumpers determine how it can be used.

The individual options are shown again in detail in the coming sections.

Chips to be tested	Power supply via	assembly
no 4116/4108 (-5V/12V not required)	**only USB (5V)** JP2 “USB” set JP3 “GND” set	DC / DC converter module not required Main board ("option") can be omitted (1x 7805, 2x capacitor, 1x rectifier)
	Barrel Jack (6-12V) or USB (5V) JP2 “USB” set JP3 “GND” set JP1/JP4 „Barrel Jack” set	DC / DC converter module not required Main board ("option") must be equipped (1x 7805, 2x capacitor, 1x rectifier)
all chips (-5V/12V required)	**Barrel Jack (6-12V) or USB (5V)** JP1, JP2, JP3, JP4 open	DC/DC converter module required Main board ("option") can be omitted (1x 7805, 2x capacitor, 1x rectifier)
	only Screw Terminal for 5V, 12V and -5V JP1, JP2, JP3, JP4 open	DC / DC converter module not required Main board ("option") can be omitted (1x 7805, 2x capacitor, 1x rectifier)

Figure 3.1: Overview of possible power supply

If a DC/DC converter module is used, jumpers 1, 2, 3 & 4 must be open!

If a power supply with a barrel connector is used, the power supply should deliver between 7V to 9V. Operation with 12V is quite possible, but then the 7805 should be provided with a heat sink. Operation above 12V is not recommended.

Important note when operating via USB:

The supply voltage Vcc (+ 5V) comes directly from the USB supply. Unfortunately, some cheap USB power supplies have incorrectly rated amperage and sometimes deliver 4.8V or less. Due to the voltage drop (transistors, diodes, etc.) of approx. 0.3V, the 4.8V is still within the acceptable range (usually +/- 10% tolerance at Vcc). If it becomes less, the reading of e.g. ROMs fail. 6540 ROMs in particular need the correct voltage or the tests will fail.

If a ROM cannot be read out (symptom: repeated readout provides different CRC32), then try to operate the Tester using the power supply via the barrel connector. The linear regulator (whether with or without a DC/DC module) delivers exactly 5V for Vcc.

Operation on a USB port:

Operation on a normal USB 2.0/USB 3.0 port is possible. However, there is no power negotiation, as is the case with many other devices with USB power supply. The Tester assumes that 500 mA is available. Usually this does not lead to any problems. In the “worst case” the USB port only provides 100 mA, which is not enough for the Tester. In this case, the Tester should be connected to a USB power supply unit or a USB hub with its own power supply.

Warning:

Never supply the RCT with power via USB and barrel connector at the same time!

3.1 … via USB or barrel connector with DC/DC module (recommended)

With the DC/DC module in place, all chips can be tested. The module provides the necessary supply voltages of -5V, 5V and 12V. Power can be supplied either via USB or the barrel connector (7 - 9V).

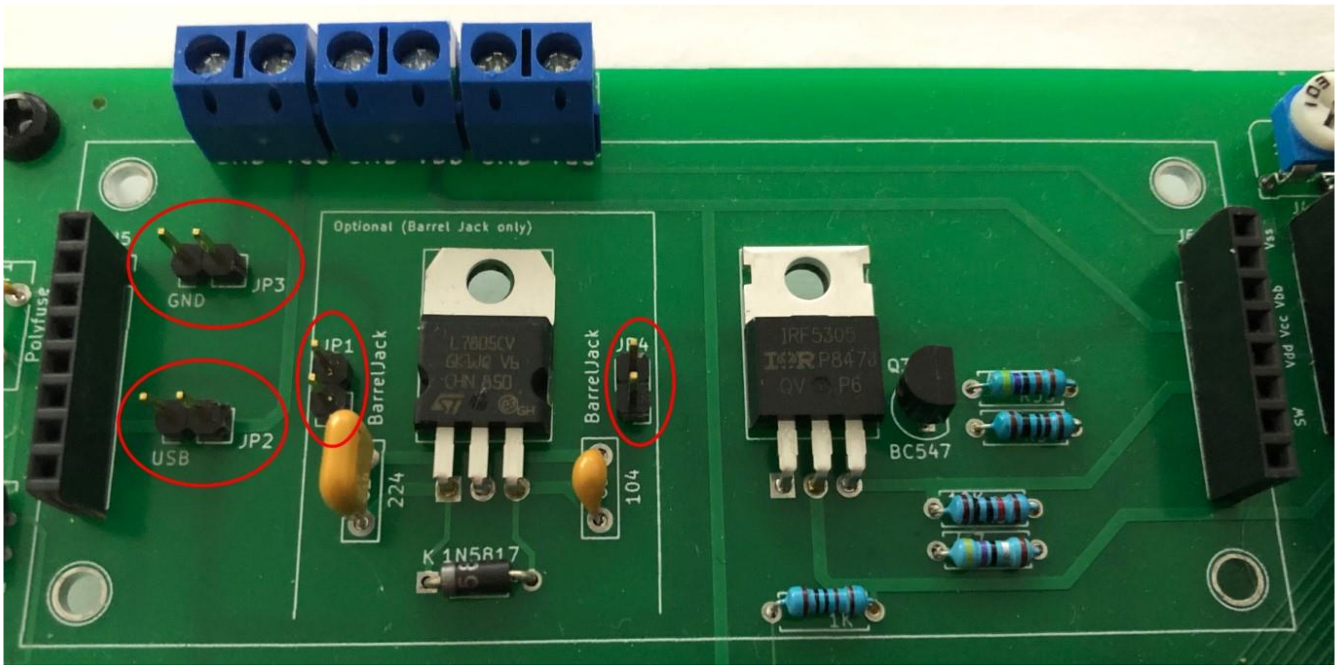


Figure 3.2: All jumpers open

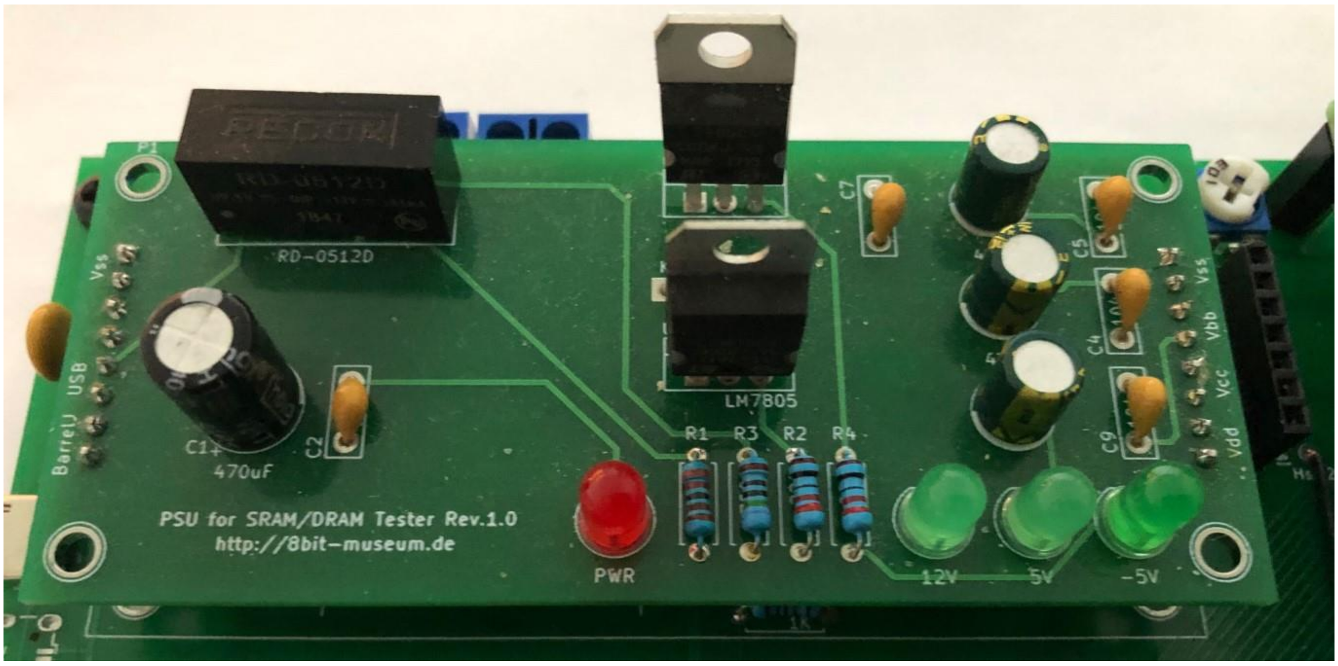


Figure 3.3: Plugged DC/DC module

If the DC/DC module is selected for the voltage supply, it is not necessary to equip the voltage regulator on the main board at location U2 in the "Optional (Barrel Jack only)" section. Jumpers 1, 2, 3 & 4 should be open.

3.2 … via screw terminals without DC/DC module

Even without a DC/DC module plugged in, all chips can be tested if the power is supplied via the screw terminals.

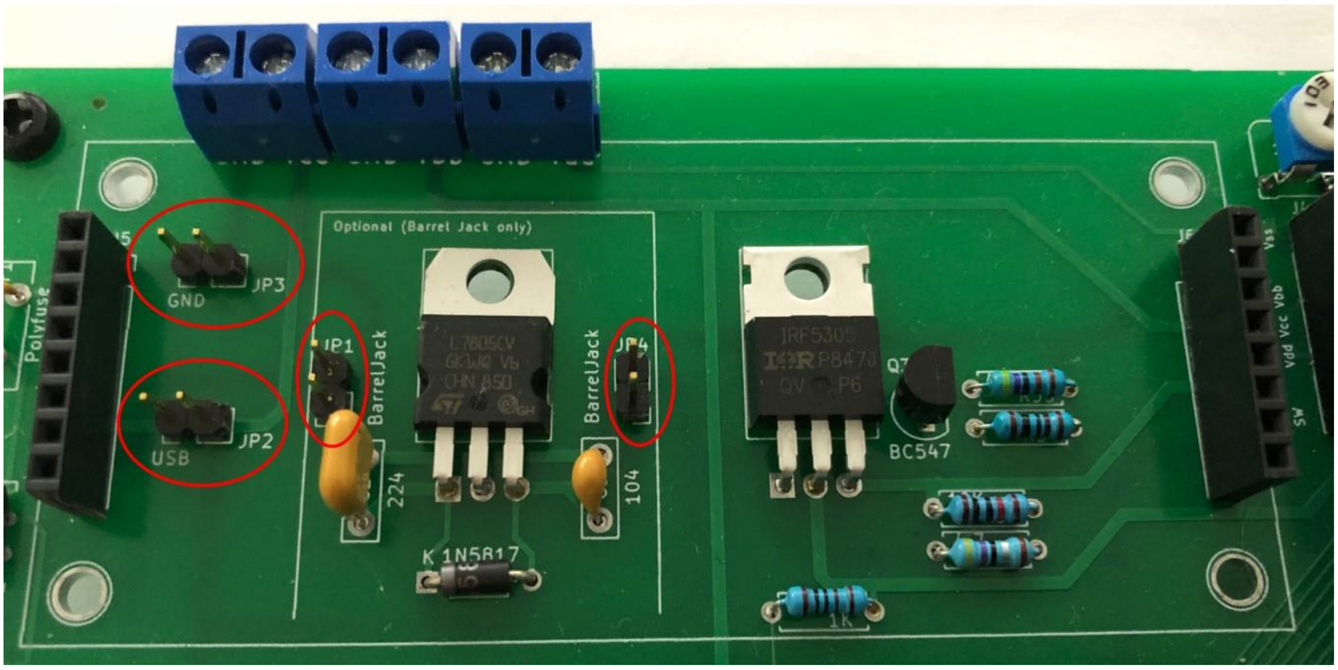


Figure 3.4: All jumpers open

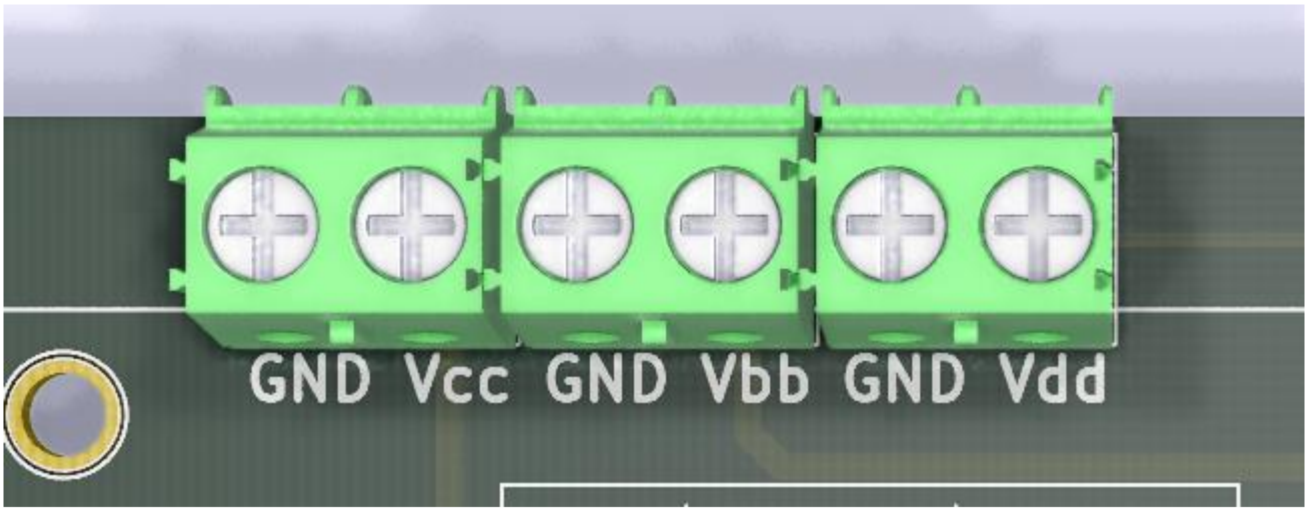


Figure 3.5: Assignment of the screw terminals (Vcc = 5V, Vbb = -5V, Vdd = 12V)

If the screw terminals are used, jumpers 1, 2, 3 & 4 should be open. The voltage supply must be stabilized. It is best to use a PC PSU for power supply. Pay attention to the polarity and the correct voltages!

3.3 … via USB or barrel connector without DC/DC module

Not all chips can be tested without a plugged-in DC/DC module: Chips that require a negative supply voltage of -5V and/or 12V cannot be tested without the DC/DC module.

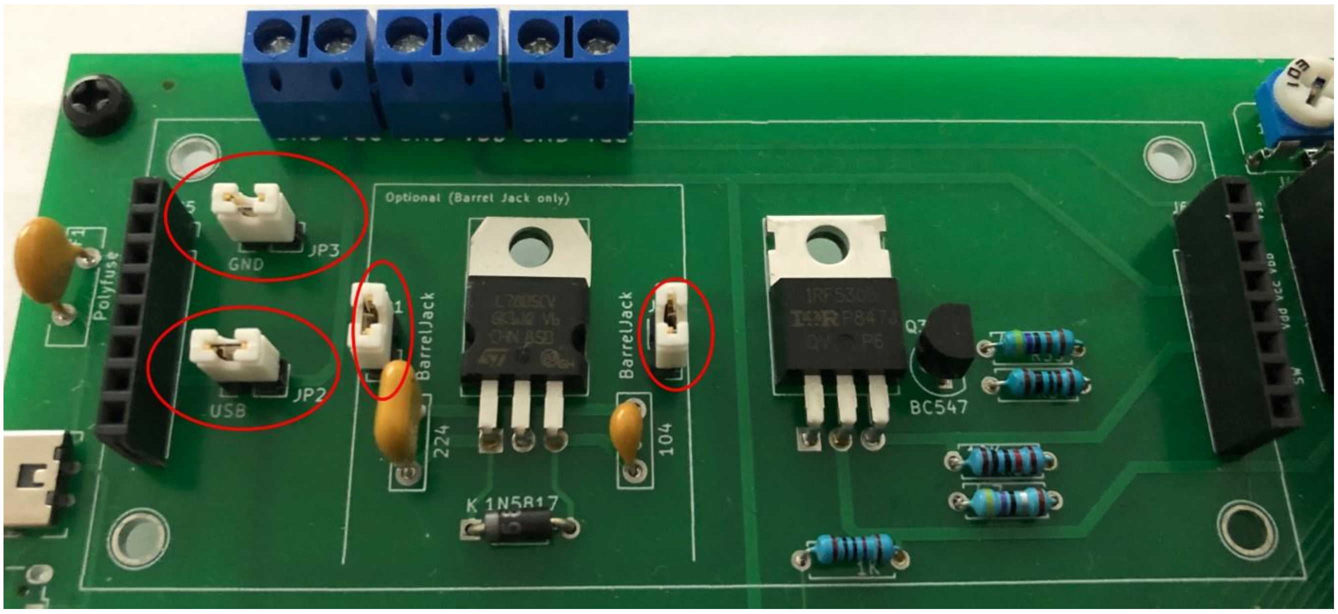


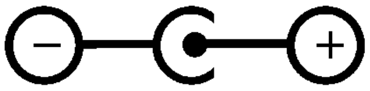
Figure 3.6: All jumpers closed

If jumpers JP2 and JP3 are closed, the tester is supplied directly via the USB supply voltage.

If JP3, JP1 and JP4 are closed, the voltage at the barrel connector is regulated to 5V by the voltage regulator.

When all four jumpers are closed, the power can be supplied either via USB or barrel connector.

The barrel connector has positive polarity.



⚠ If the jumpers are shorted the DC/DC module must not be connected to the main board!

3.4 Power supply using an optional DC/DC module

The following table gives an overview of the available options:

	Option #1 / Standard **RD-0512D**	**Option #2** **RD-0512D/LM317**	**Option #3** **RD-0512D/RS-0505S**
Supply via barrel connector	7.5V – 9V 7.5V recommended	7.5V – 9V	4.5V – 9V (works also with 4V)
Vcc at USB	Vcc corresponds to input voltage	Vcc corresponds to input voltage	Vcc = 5V
Vcc using barrel connector	Vcc = 5V	Vcc = 4.8V / 5V / 5.15V / 5.3V	Vcc = 5V
Specials		Tests with increased voltage	from 4.5V input voltage
Costs	cheap	cheap	expensive

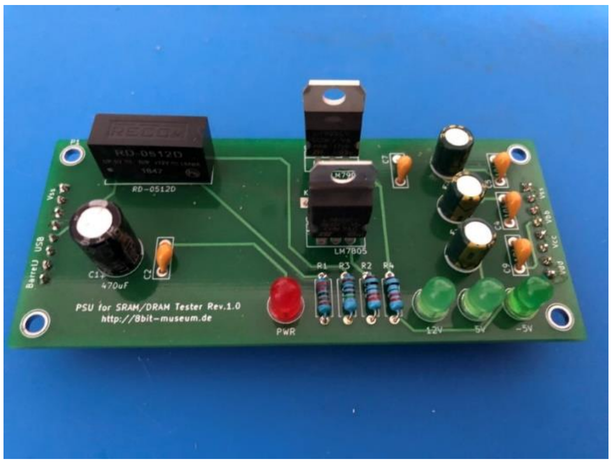
Warning:

Never supply the RCT with power via USB and barrel connector at the same time!

Option #1 (using RD-0512D) - Standard

This option uses a RECOM DC/DC controller. No heat is generated even after long use and the output voltage is properly regulated. This power supply is the standard power supply for the chip tester.

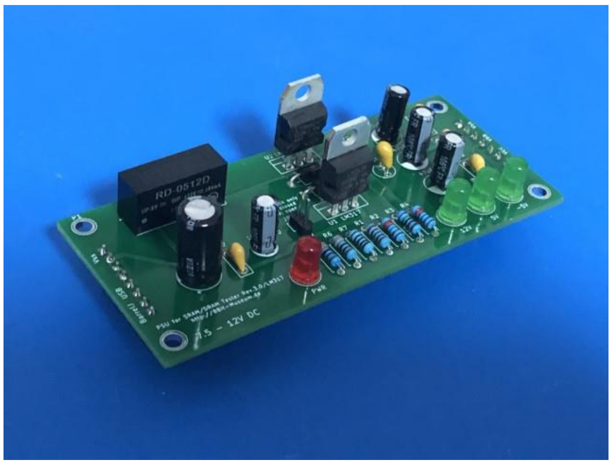
On the circuit board, R1 can still be labeled “2.2k”. 2.2k is sufficient for low-power LEDs, but the value for normal LEDs should be 1k so that the LED is sufficiently bright.



Option #2 (with RD-0512D and LM317)

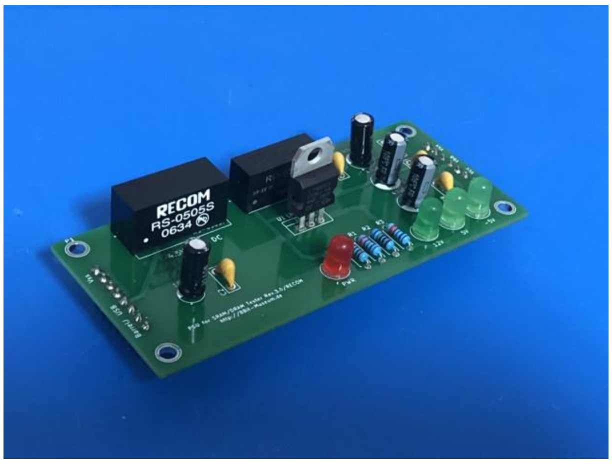
Like option #1, the second option uses a RECOM controller, but instead of the fixed voltage controller an LM317 for the 5V supply voltage is used. By setting a jumper, the supply voltage can be raised from 5V up to 5.3V. This may be necessary for special tests, e.g.

- for NVRAMs that go into a read-only mode below 4.8V (rare),
- in the case of aged memory chips, which behavior you would like to test under increased voltage.



Option #3 (with RD-0512D and RS-0505S)

The third option uses two RECOM controls. If the chip tester is supplied with less than 5V, e.g. USB connection below 5V (USB 1.0 allows between 4.25V and 5.25V) or by battery, then a regulated 5V voltage is initially generated by the additional module.



4 Programming the Tester (fuses and firmware)

You need a 6-pin ISP programmer (e.g. "Diamex USB ISP programmer for Atmel AVR, Rev.2" or "Diamex USB ISP-Programmer for AVR, STM32, LPC-Cortex (Prog-S2)"). The power is supplied by the ISP programmer during the programming process.

When programmer the microcontroller for the first time the programming is done in 2 steps.You can find this in the archives provided:

- a batch file and AVRdude for setting the Fuses, and
- a batch file and AVRdude for programming the ATmega.

The files are partially preconfigured and have to be adapted for your own programmer. Usually. no additional software is required (apart from the programmer and AVRdude).

The above actions can be done in different ways (please choose according to your own preference). The ISP programmer used (parameter "-c") and, if applicable, the COM port (parameter "-P") must be adapted according to your own hardware. These parameters are marked in red below.

Diamex	`-c stk500`	`-P COMx` required
Pololu	`-c stk500`	`-P COMx` required
avrisp mk2	`-c avrispmkii`	`-P` not required
Atmel Ice	`-c avrispmkii`	`-P` not required

When programming with one of the programmers mentioned above, the tester can be supplied with voltage by these. Most programmers have a dip switch (e.g. Diamex) or a jumper for this. Please make sure that the programmer provides the 5V supply voltage and thus supplies the ATmega2560 (the display may have to be removed).

In section 9 there are further instructions for some programming devices. Suitable programmers are listed in sec. 8.1!

If the Tester is supplied with voltage (5V) via ISP, it must not be supplied with voltage via USB, barrel connector etc.!

The experiences with the USBASP are mixed. A recommendation for this programmer cannot be given. Please also check sec. 8.1!

The SD card module shares lines with the ISP port. The SD card module must not be plugged in during programming, otherwise it will occupy the ISP port and programming will fail.

Warnings regarding programming the memory tester:

Never supply the memory tester with voltage using the ISP programmer and via USB, barrel connector or screw terminal **at the same time**.

If the ATmega2560 is programmed via ISP with the plugged DC/DC converter "Alternative #2" (with Recom module), please note that if the DC/DC converter has a supply voltage from the programmer, it generates Vdd and Vbb. The programmer should be able to tolerate this additional load, otherwise the DC/DC converter should be removed beforehand. The recommended programmer tolerates this and chips can even be tested.

4.1 Programming the Fuses

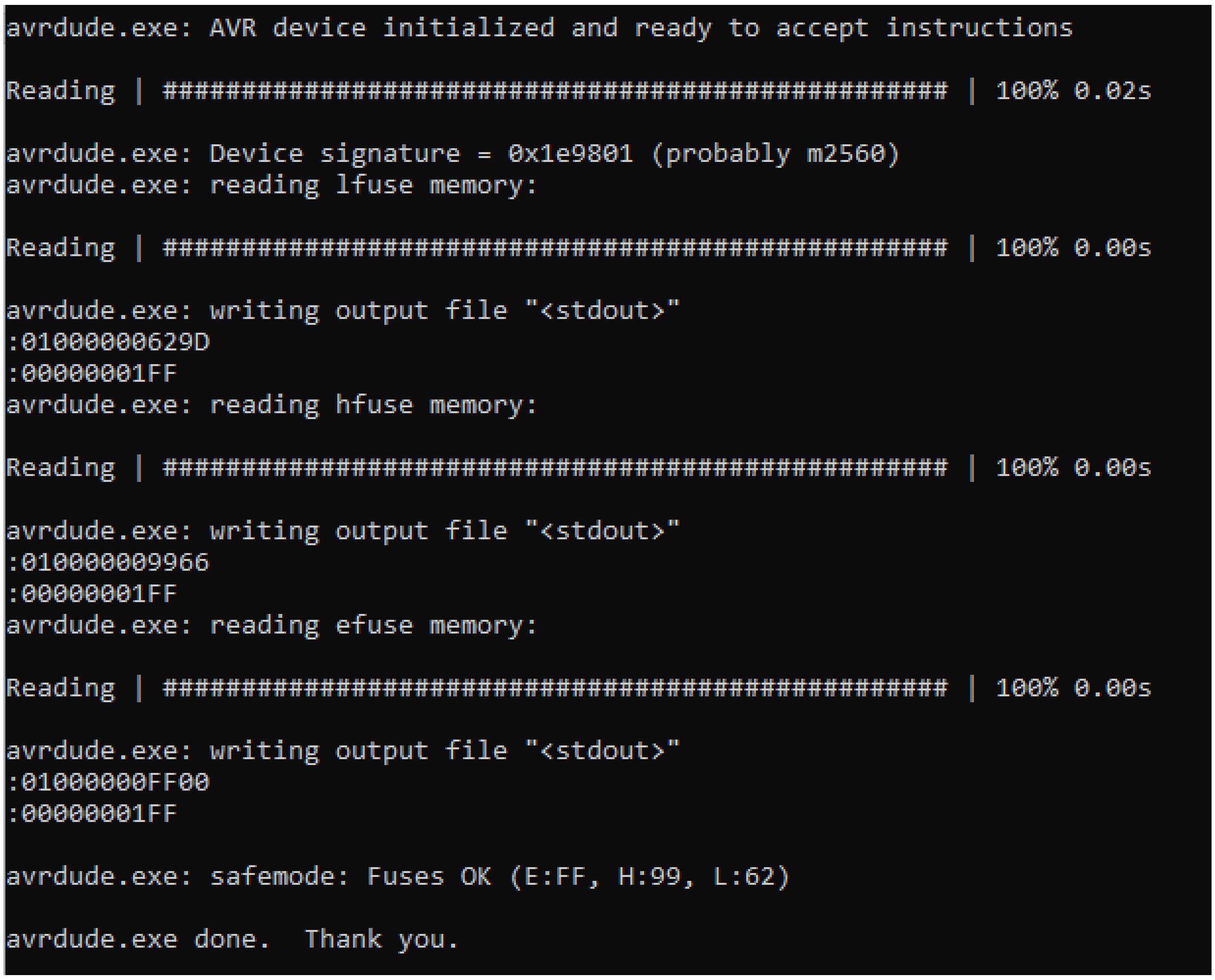
STEP 1

Possibility 1: via command line

First, communication should be tested, this can be done via

```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cstk500 -PCOM5
-U lfuse:r:-:i -U hfuse:r:-:i -U efuse:r:-:i
```

With this command the fuses set are displayed only.



```
avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.02s

avrdude.exe: Device signature = 0x1e9801 (probably m2560)
avrdude.exe: reading lfuse memory:

Reading | ################################################## | 100% 0.00s

avrdude.exe: writing output file "<stdout>"
:01000000629D
:00000001FF
avrdude.exe: reading hfuse memory:

Reading | ################################################## | 100% 0.00s

avrdude.exe: writing output file "<stdout>"
:010000009966
:00000001FF
avrdude.exe: reading efuse memory:

Reading | ################################################## | 100% 0.00s

avrdude.exe: writing output file "<stdout>"
:01000000FF00
:00000001FF

avrdude.exe: safemode: Fuses OK (E:FF, H:99, L:62)

avrdude.exe done.  Thank you.
```

Figure 4.1: “blank” ATmega2560

The fuses are then programmed as follows:

```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cstk500 -PCOM5
-U lfuse:w:0xf7:m -U hfuse:w:0xd7:m -U efuse:w:0xff:m
```

Note: By setting the h-fuse to "`0xd7`", the configuration of the tester is saved permanently and survives a new programming of the firmware. By changing the h-fuse from "`0xd7`" to "`0xdf`", the EEPROM is erased with each flash process and with it an existing configuration.

Possibility 2: by batch file

In the folder "fuses" there is a batch file `flash_fuses_for_stk500-wiring-usbasp.bat`.

This batch file queries the following information one after the other:

1. Programmer used (STK500 comp., Wiring, USBASP)
2. COM port used (with STK500 or Wiring)
3. Delete the configuration in the EEPROM

Then the currently set fuses are displayed and reset after a keypress. If desired, a batch file `flash_firmware.bat` can be created, which can be used in step 2 - programming the firmware - and which contains the parameters already entered.

A video is available on YouTube:

https://www.youtube.com/watch?v=KFFmAwlJ9ns (German)

https://www.youtube.com/watch?v=ijMVnq-y_vA (English)

Note on the error message “stk500v2_command(): command failed”:

Depending on the programmer, AVRDUDE can output some messages that can be irritating.

The message

```
avrdude: stk500v2_command(): command failed
avrdude: stk500v2_getparm(): failed to get parameter 0x9a
```

is not an error, just an information that a command is not supported by the programmer.

The Atmel STK-500 supports extensions (top cards), which are queried with the command 0x9a from AVRDUDE. Except for the original, these extensions are not supported by any programmer, hence the message.

Important note (experts know-how only, all other can skip this ;)):

With the above settings, the crystal that drives the ATmega is operated with the "Full Swing Crystal Oscillator", which requires a little bit more power. This makes sense, as it means that crystals with a low voltage swing (= poor quality) can also be used. If you want, you can try the "Low Power Crystal Oscillator". If there are random resets or problems flashing the firmware later, you can switch back to the "Full Swing Crystal Oscillator" at any time.

When the "Low Power Crystal Oscillator" is used, the value of the `lfuse` has to be changed from "`0xf7`" to "`0xff`" when setting the fuses.

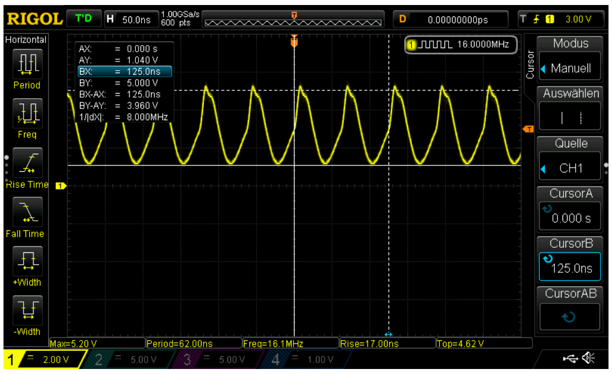
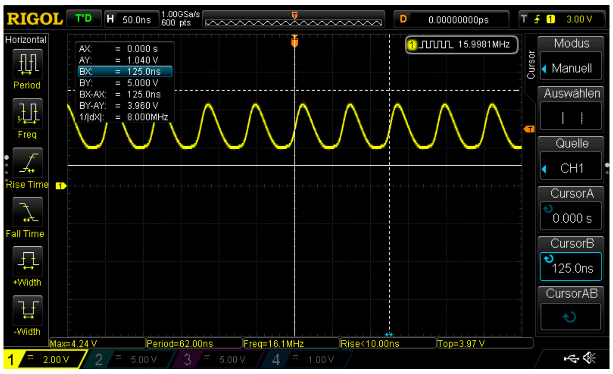


Figure 4.2: a “good“ crystal



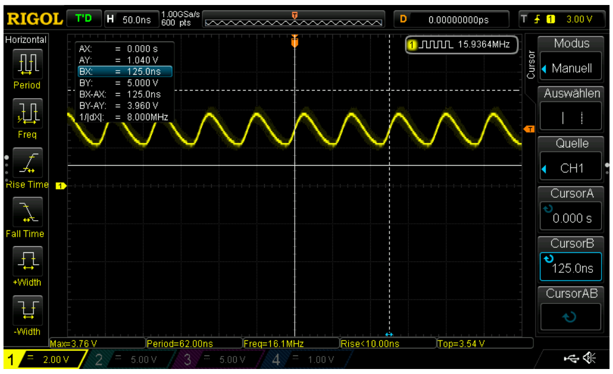
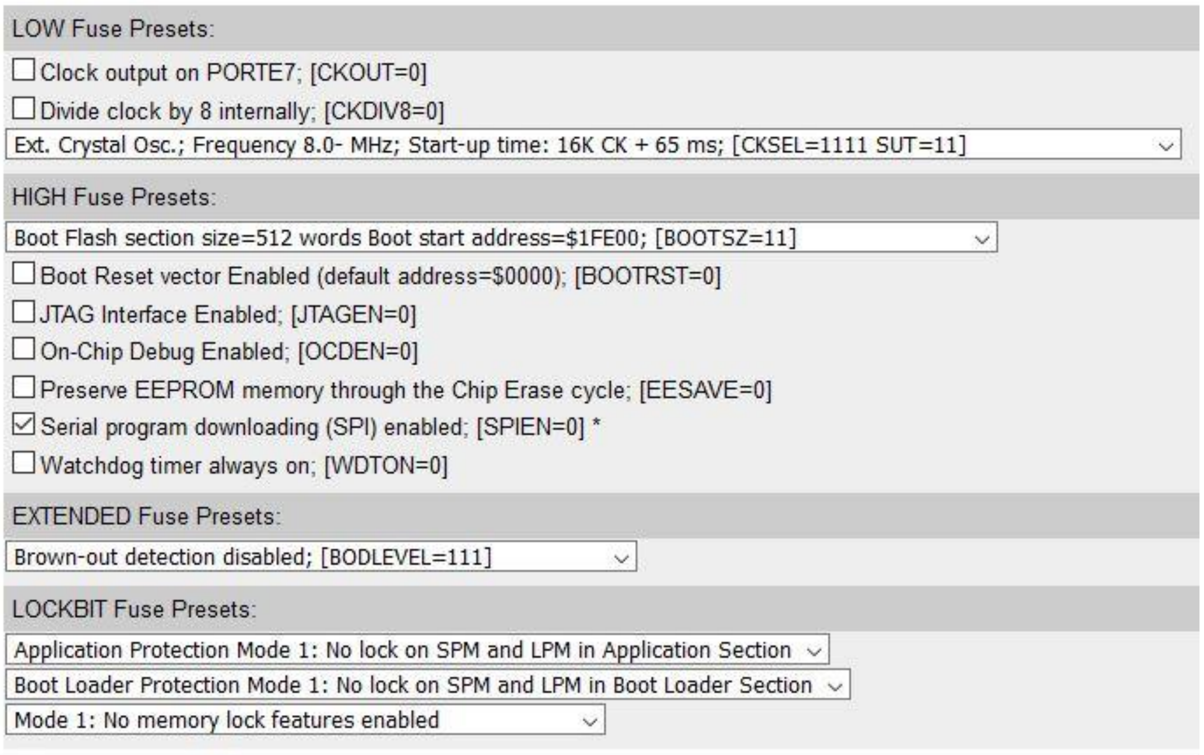


Figure 4.3: two “not so good“ crystals

In the pictures above, three different crystals were used in the same oscillator circuit.

When you would like to set the Fuses with another program, you will find the recommended settings in the following pictures.

LOW = FF (“Low Power Crystal Oscillator“), HIGH = DF (“Don’t save EEPROM”), EXTENDED = FF



LOW Fuse Presets:

☐ Clock output on PORTE7; [CKOUT=0]

☐ Divide clock by 8 internally; [CKDIV8=0]

Ext. Crystal Osc.; Frequency 8.0- MHz; Start-up time: 16K CK + 65 ms; [CKSEL=1111 SUT=11]

HIGH Fuse Presets:

Boot Flash section size=512 words Boot start address=$1FE00; [BOOTSZ=11]

☐ Boot Reset vector Enabled (default address=$0000); [BOOTRST=0]

☐ JTAG Interface Enabled; [JTAGEN=0]

☐ On-Chip Debug Enabled; [OCDEN=0]

☐ Preserve EEPROM memory through the Chip Erase cycle; [EESAVE=0]

☑ Serial program downloading (SPI) enabled; [SPIEN=0] *

☐ Watchdog timer always on; [WDTON=0]

EXTENDED Fuse Presets:

Brown-out detection disabled; [BODLEVEL=111]

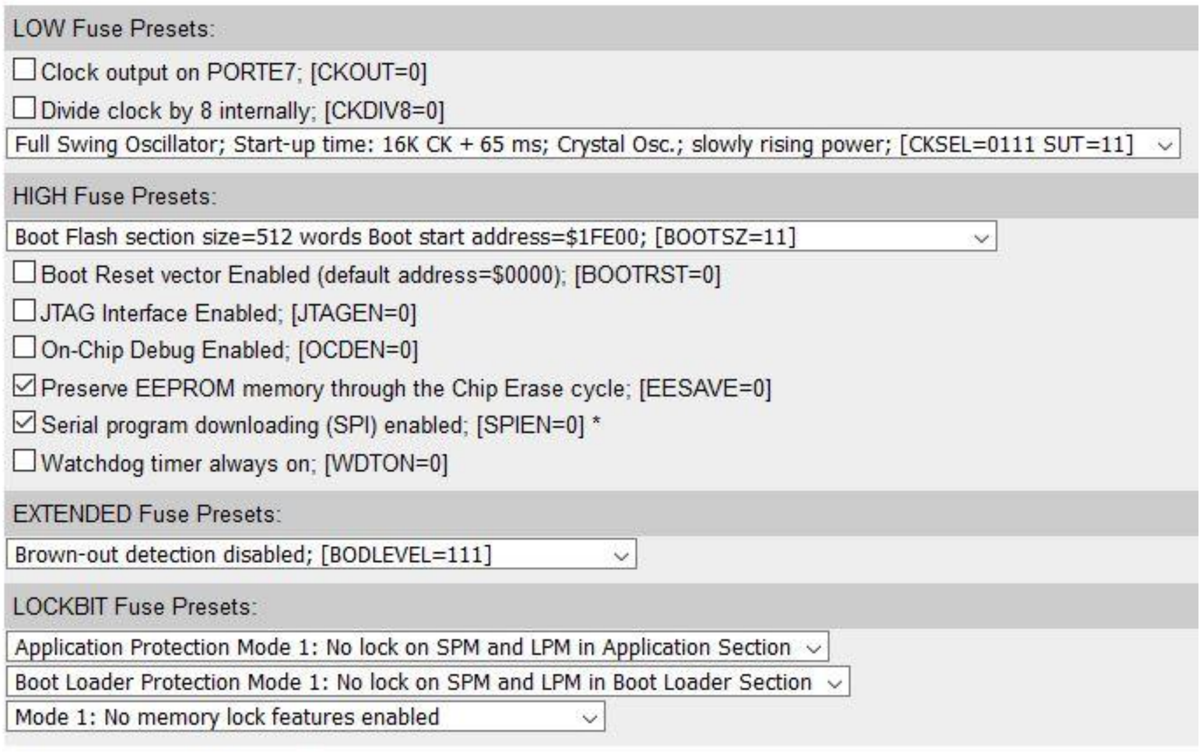
LOCKBIT Fuse Presets:

Application Protection Mode 1: No lock on SPM and LPM in Application Section

Boot Loader Protection Mode 1: No lock on SPM and LPM in Boot Loader Section

Mode 1: No memory lock features enabled

LOW = F7 (“Full Swing Crystal Oscillator”), HIGH = D7 (“Preserve EEPROM”), EXTENDED = FF



LOW Fuse Presets:

☐ Clock output on PORTE7; [CKOUT=0]

☐ Divide clock by 8 internally; [CKDIV8=0]

Full Swing Oscillator; Start-up time: 16K CK + 65 ms; Crystal Osc.; slowly rising power; [CKSEL=0111 SUT=11]

HIGH Fuse Presets:

Boot Flash section size=512 words Boot start address=$1FE00; [BOOTSZ=11]

☐ Boot Reset vector Enabled (default address=$0000); [BOOTRST=0]

☐ JTAG Interface Enabled; [JTAGEN=0]

☐ On-Chip Debug Enabled; [OCDEN=0]

☑ Preserve EEPROM memory through the Chip Erase cycle; [EESAVE=0]

☑ Serial program downloading (SPI) enabled; [SPIEN=0] *

☐ Watchdog timer always on; [WDTON=0]

EXTENDED Fuse Presets:

Brown-out detection disabled; [BODLEVEL=111]

LOCKBIT Fuse Presets:

Application Protection Mode 1: No lock on SPM and LPM in Application Section

Boot Loader Protection Mode 1: No lock on SPM and LPM in Boot Loader Section

Mode 1: No memory lock features enabled

4.2 Programming the Firmware

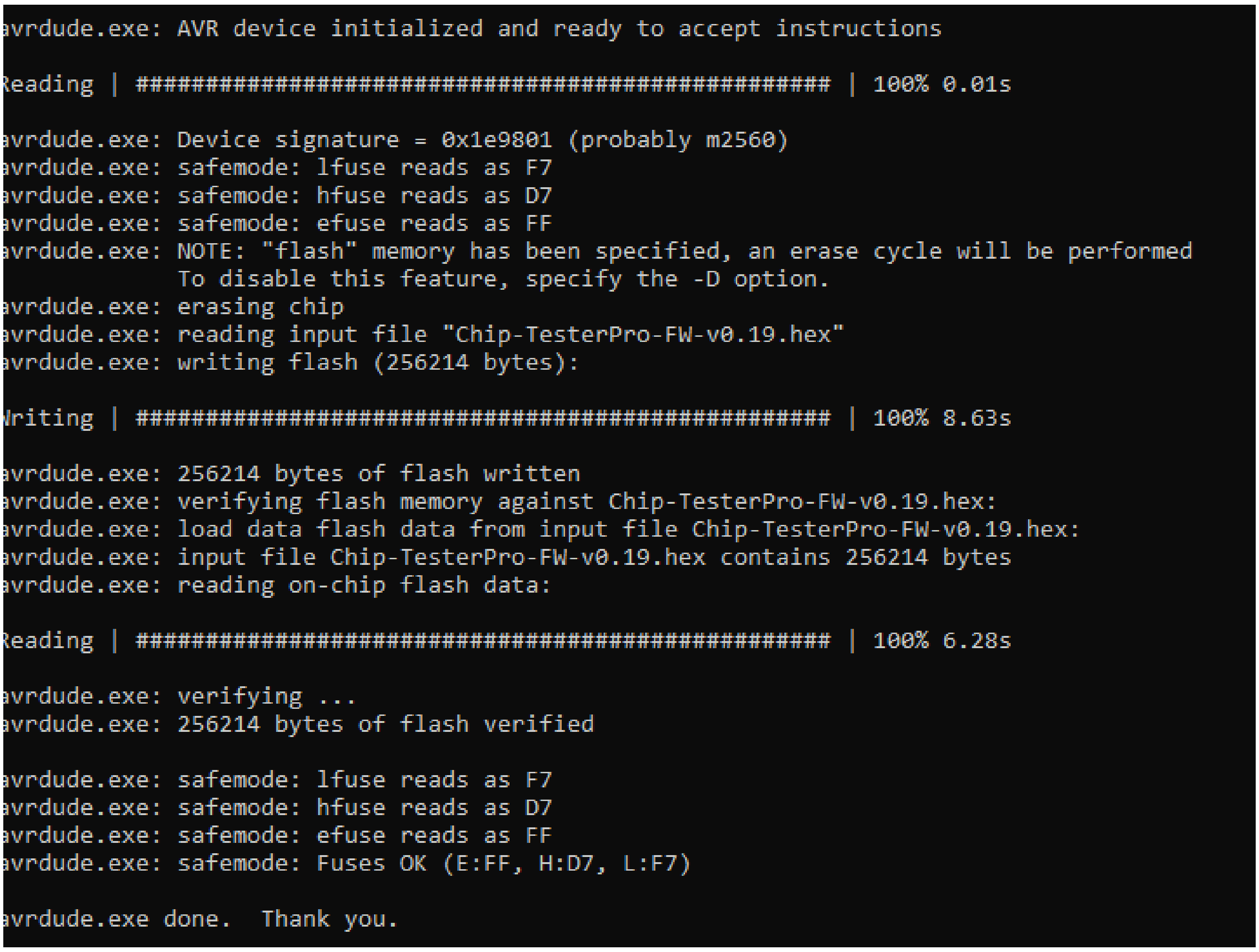
STEP 2

Possibility 1: via command line

The firmware is programmed as follows:

```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cstk500 -PCOM5
-Uflash:w:Chip-TesterPro-FW-v0.XX.hex:i
```

The file name must be adapted to the version of the firmware to be burned.



```
avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.01s

avrdude.exe: Device signature = 0x1e9801 (probably m2560)
avrdude.exe: safemode: lfuse reads as F7
avrdude.exe: safemode: hfuse reads as D7
avrdude.exe: safemode: efuse reads as FF
avrdude.exe: NOTE: "flash" memory has been specified, an erase cycle will be performed
             To disable this feature, specify the -D option.
avrdude.exe: erasing chip
avrdude.exe: reading input file "Chip-TesterPro-FW-v0.19.hex"
avrdude.exe: writing flash (256214 bytes):

Writing | ################################################## | 100% 8.63s

avrdude.exe: 256214 bytes of flash written
avrdude.exe: verifying flash memory against Chip-TesterPro-FW-v0.19.hex:
avrdude.exe: load data flash data from input file Chip-TesterPro-FW-v0.19.hex:
avrdude.exe: input file Chip-TesterPro-FW-v0.19.hex contains 256214 bytes
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 100% 6.28s

avrdude.exe: verifying ...
avrdude.exe: 256214 bytes of flash verified

avrdude.exe: safemode: lfuse reads as F7
avrdude.exe: safemode: hfuse reads as D7
avrdude.exe: safemode: efuse reads as FF
avrdude.exe: safemode: Fuses OK (E:FF, H:D7, L:F7)

avrdude.exe done.  Thank you.
```

Figure 4.4: finished programmed ATmega2560 (here FW v.19)

When the message

```
avrdude.exe: verifying ...
avrdude.exe: verification error, first mismatch at byte 0x0000
             0xXX != 0xYY
avrdude.exe: verification error; content mismatch
```

appears, then it is very likely that the programmer (USBASP used?) is not suitable for programming an ATmega2560 (see sec.8.1).

Possibility 2: by batch file (self-created)

The batch file `flash_firmware_(please_edit).bat` must be adapted to the programmer used and the COM port. The file is written in such a way that the firmware found is burned.

Possibility 3: by batch file (automatically created)

If the batch file `flash_firmware.bat` was created when the fuses were set, this can be used to burn the firmware. To do this, copy it to the firmware folder and start it.

Possibility 4: by interactive batch file

The batch file `winflash.bat` is included. If the firmware file is dropped on it, it queries the programmer used and, if applicable, the COM port interactively.

TIP:

When the fuses are correctly set, you can flash the firmware. With parameter „`-B1`“ you can speed up the flashing significantly. However, this should not be tried out during the initial programming.

4.3 Updating the Firmware

A firmware update only requires programming the firmware as described in section 4.2. It is not necessary to set the fuses again.

5 First start-up

The chip tester should be ready for use immediately after it has been assembled and programmed. As soon as it is supplied with voltage, a power-on message should appear for approx. two seconds and then the menu.

If not, don't panic and go through the following short checklist:

No text appears on the display and the chip tester is operated without the DC/DC module?

- Are the jumpers set on the board?

No text appears on the display and the chip tester is operated with the DC/DC module?

- Are all three (green) LEDs on the module for the supply voltages lit? If an LED does not light up, this may indicate a short circuit. Please disconnect the chip tester from the voltage and refer to section 8.

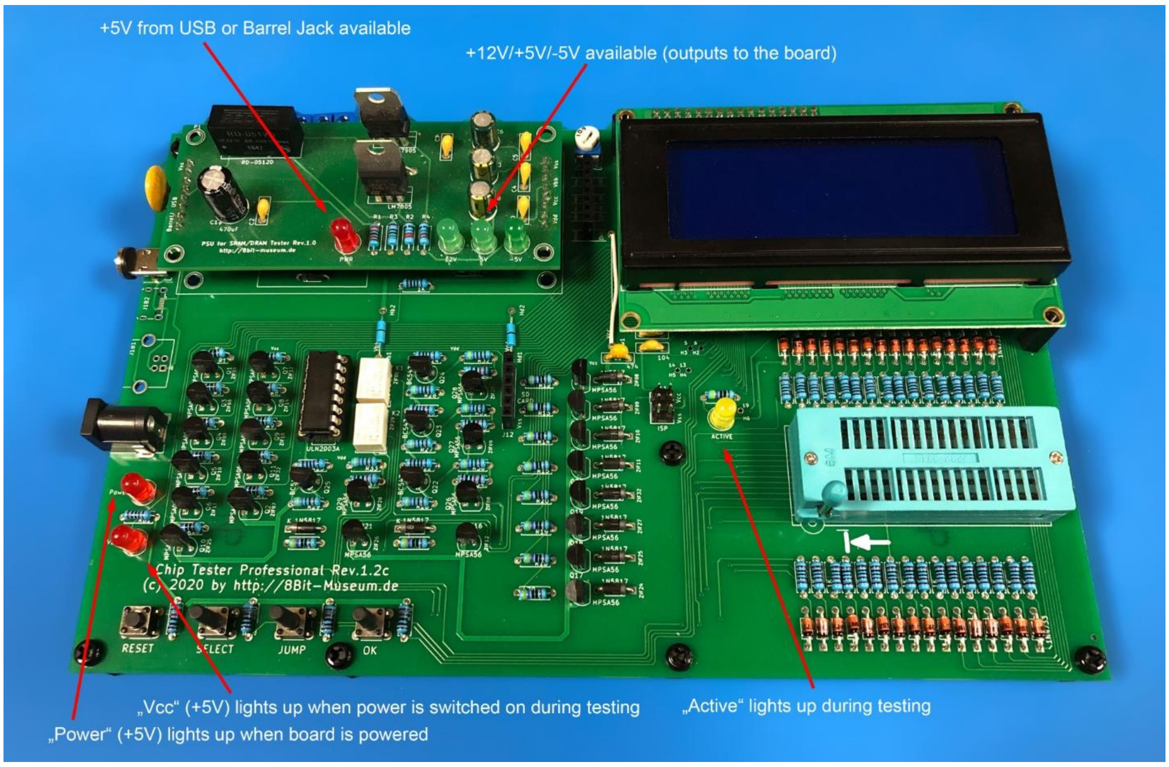
Is the display lit, but no text appears?

- If so, then the contrast is probably not set correctly. This can be changed with the potentiometer to the left of the display.

Does the power-up menu appear, but operation is extremely slow?

- If the tester reacts very slowly, then the fuses of the ATmega2560 have not been set correctly, see section 4.1. If this was forgotten, the microprocessor only runs at 1 MHz and not at 16 MHz.

These are the most common problems when starting up for the first time. If you have any further problems, please refer to section 8.


+5V from USB or Barrel Jack available
+12V/+5V/-5V available (outputs to the board)
„Vcc“ (+5V) lights up when power is switched on during testing
„Active“ lights up during testing
„Power“ (+5V) lights up when board is powered

"Vcc" lights up if the supply voltage is switched on during the tests. "Power" lights up as soon as the Tester is supplied with voltage. "Active" indicates that testing is in progress.

6 Connecting the SD card adapter

In order to be able to dump memory modules, an SD card adapter must be connected to the memory tester. There are several options for this.

6.1 Connection to the 6-pin header

Boards from Rev.1.2c have a 6-pin header. The SD card adapter can be set directly in this header. The connection "Vss" is shown denoting pin 1. Inserting the SD card module the wrong way may damage it.

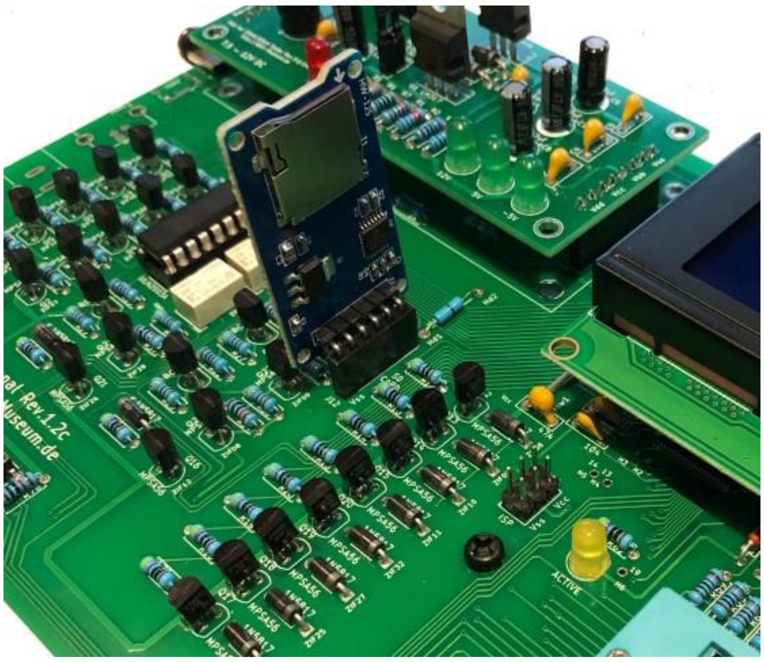


Figure 6.1: SD card adapter in the assembled header.

If you want, you can also remove the existing pin header on the SD card adapter and insert a new pin header on the back. This allows the SD card adapter to be installed horizontally (watch out for short circuits).

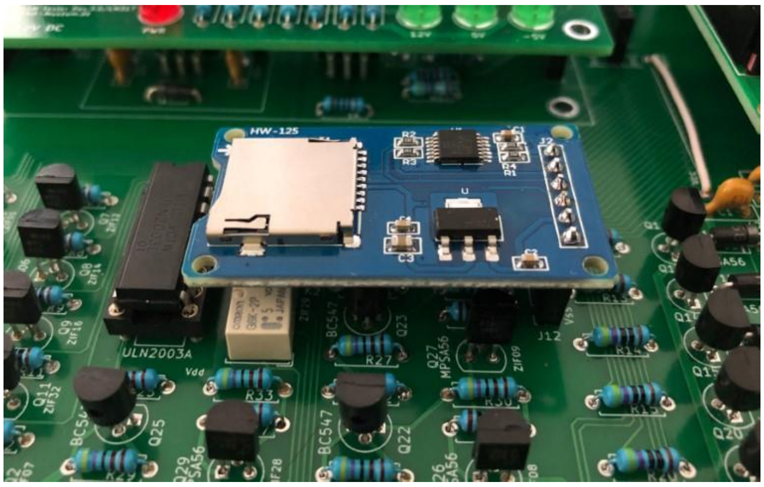


Figure 6.2: SD card adapter installed horizontally

6.2 Preparing the SD card

Before the SD card can be used, support for SD cards must be activated in the configuration (“**SD Card: 1**”) and the SD card must be FAT32 formatted. The supported maximum size is 32 GByte.

When the SD card is not recognized, the card should be formatted using the “SD Memory Card Formatter” from the SD Association.

https://www.sdcard.org/downloads/formatter/

This tool formats any SD card in compliance with the standards. All files are stored in the main directory of the SD card.

6.3 Connection to the SPI connection via the 6-pin ISP connector

The connection to the SPI port is possible as follows (note that CS is connected to GND because the ISP does not have a CS signal):

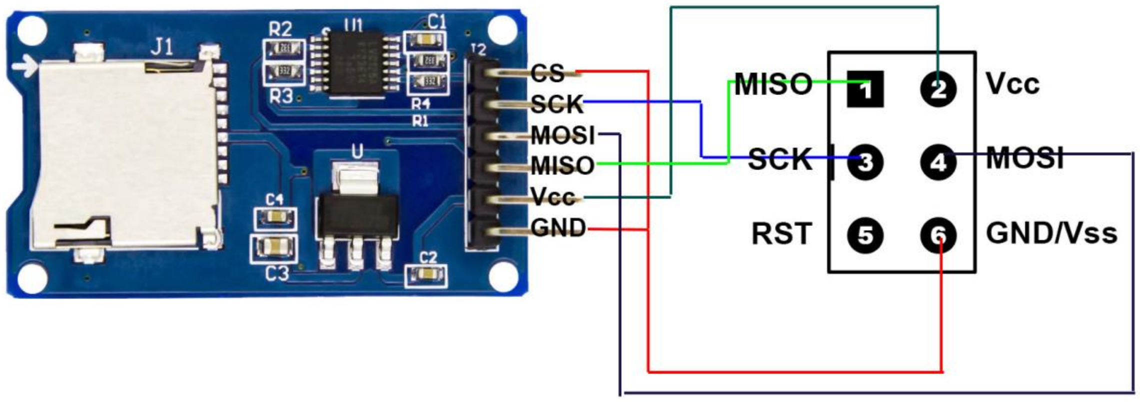


Figure 6.3: Wiring of the SD card adapter

A suitable cable can be assembled quickly with a crimping tool for Dupont plugs:

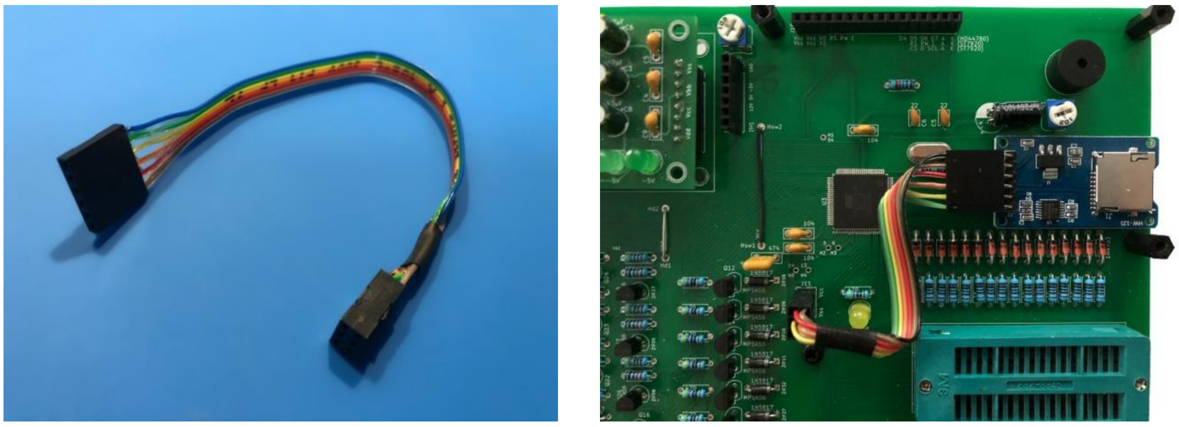


Figure 6.4: SD card adapter with extension

The SD card adapter fits right on the circuit board directly under the display. Make sure that there is no short circuit. Thanks to Greg64 for the idea.

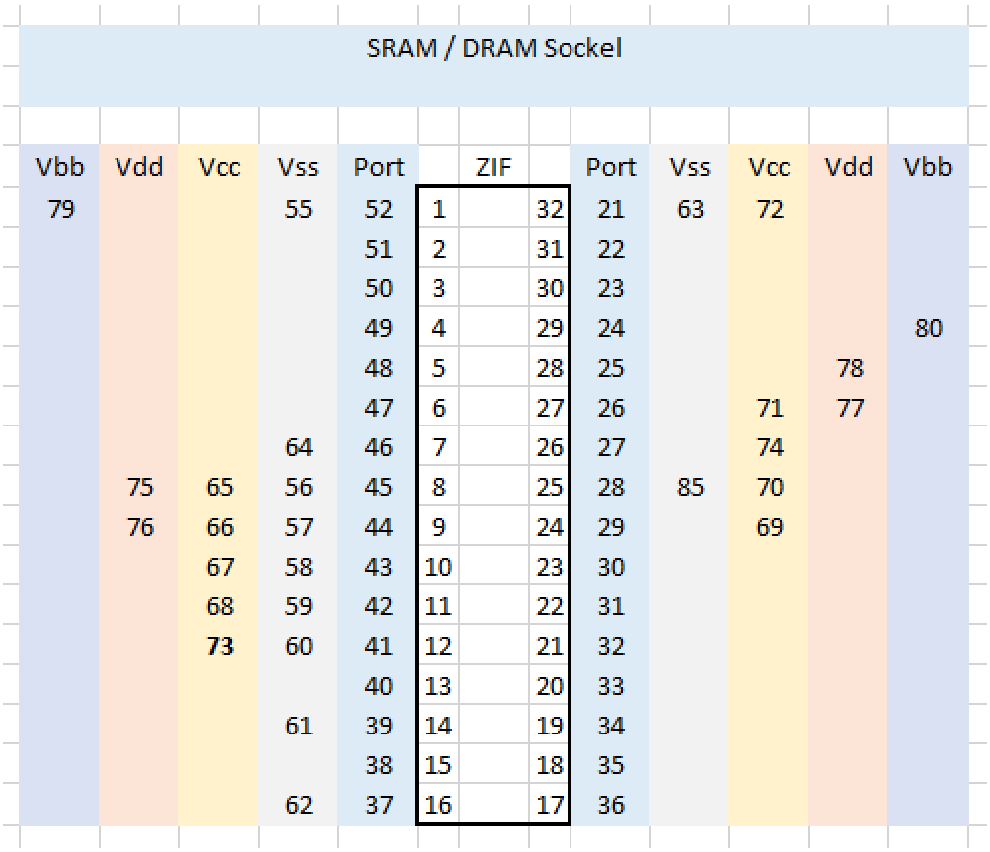
From Rev.1.2c the connection should be made using the pin header provided for this purpose (see sec. 6.1).

7 Own experiments

Even if the actual software and hardware are "closed-source", you are welcome to do your own experiments. This chapter explains how the socket is assigned and how the supply voltages can be switched.

7.1 Pin assignment of the socket

The pin assignment of the socket (AVR-Pinout) is as follows:



SRAM / DRAM Sockel

Vbb	Vdd	Vcc	Vss	Port	ZIF		Port	Vss	Vcc	Vdd	Vbb
79			55	52	1	32	21	63	72		
				51	2	31	22				
				50	3	30	23				
				49	4	29	24				80
				48	5	28	25			78	
				47	6	27	26		71	77	
			64	46	7	26	27		74		
	75	65	56	45	8	25	28	85	70		
	76	66	57	44	9	24	29		69		
		67	58	43	10	23	30				
		68	59	42	11	22	31				
		73	60	41	12	21	32				
				40	13	20	33				
			61	39	14	19	34				
				38	15	18	35				
			62	37	16	17	36				

Figure 7.1: Controlling the socket

SELECT, OK, and JUMP are inputs (buttons), ACTIVE (LED), Buzzer and Vcc Power (switches the MOSFET) are outputs.

SELECT	10
JUMP	11
OK	12
Vcc switch	20
Active LED	7
Buzzer	53

The display uses RS=9, EN=8, D4-D7: 81, 82, 83, 84.

All names still correspond to those of the MegaCore Framework, but the software no longer uses it and addresses the ports directly for reasons of speed.

A boot loader is not required because programming is carried out via the ISP connection (faster and saves memory and costs).

7.2 Programming with the help of the Arduino framework

Due to its past, the Chip Tester can still be programmed using the Arduino framework. Since the standard framework of the Arduino Mega 2560 does not support all Digital I/O and the numbering of the pins is not really intuitive, the development was loosely based on the MegaCore framework. The advantage: If you want, you can use this framework to create your own firmware.

For an easier start, here are some examples for programming the chip tester. The wiring framework is not very fast; if necessary, alternative ways of setting the states should be used.

7.2.1 Addressing the HD44780 compatible display

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(9, 8, 81, 82, 83, 84);
lcd.begin(20, 4);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Hello Chip Tester");
```

7.2.2 Switching on/off Vcc

```
#define PIN_POWER 20
pinMode(PIN_POWER, OUTPUT);
digitalWrite(PIN_POWER, HIGH);        // Vcc off
digitalWrite(PIN_POWER, LOW);         // Vcc on
```

7.2.3 Buttons and LED

```
#define PIN_SELECT 10
pinMode(PIN_SELECT, INPUT);
buttonState = digitalRead(PIN_SELECT); // state of SELECT


#define PIN_ACTIVE 7
pinMode(PIN_ACTIVE, OUTPUT);
digitalWrite(PIN_ACTIVE, LOW);        // LED off
digitalWrite(PIN_ACTIVE, HIGH);       // LED on
```

7.2.4 Set and query signals

```
pinMode(52, OUTPUT);                  // ZIF01 is output
digitalWrite(52, LOW);                // ZIF01 set LOW

pinMode(52, INPUT);                   // ZIF01 in input
ZIF01State = digitalRead(52);         // state from ZIF01
```

7.2.5 Switching Vss, Vcc, Vbb and Vdd

```
    pinMode(79, OUTPUT);                   // ZIF01 Vbb
    digitalWrite(79, HIGH);                // ZIF01 Vbb on

    pinMode(55, OUTPUT );                  // ZIF01 Vss
    digitalWrite(55, HIGH);                // ZIF01 Vss on

    pinMode(75, OUTPUT);                   // ZIF08 Vdd
    digitalWrite(75, HIGH);                // ZIF08 Vdd on

    pinMode(65, OUTPUT);                   // ZIF08 Vcc
    digitalWrite(65, LOW);                 // ZIF08 Vcc on
```

7.3 Pin assignment of the power connector

The voltages 12V, 5V, -5V can be obtained at the 7-pin connector at location J4 situated between the DC/DC module and the LCD (e.g. if you want to make an adapter socket for exotic chips and need Vdd (12V) or Vbb (-5V)).

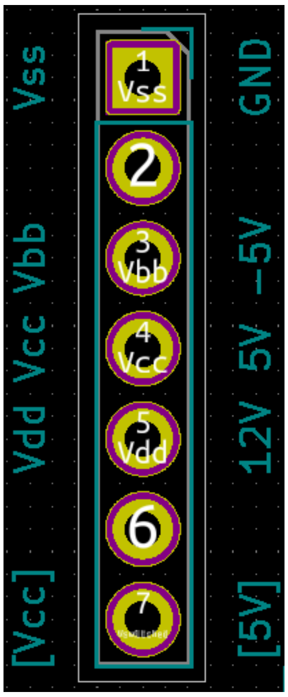
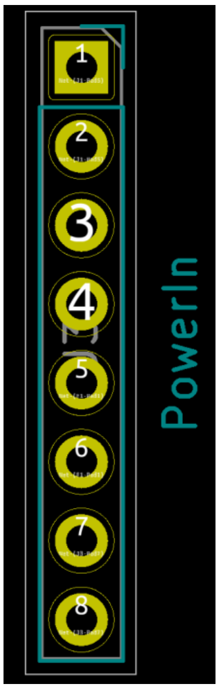


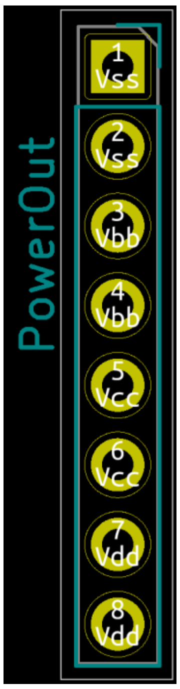
Figure 7.2: Voltage connector

[Vcc] / [5V] is only present during a test when the MOSFET is switched on (when the VCC LED is on). The other voltages are permanently available.

7.4 Pin assignment of the DC/DC-Converter PCB



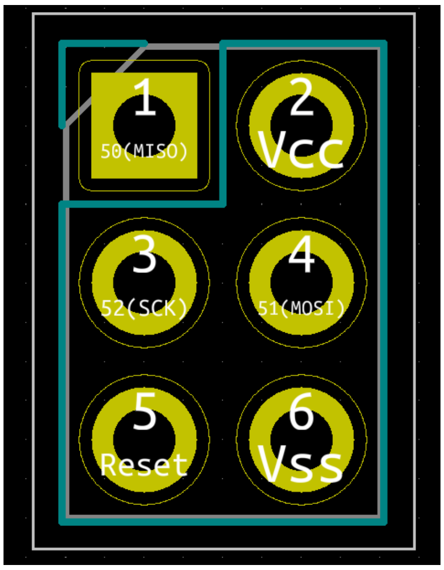
1	Vss
2	Vss
3	nc
4	nc
5	USB 5V
6	USB 5V
7	Barrel Jack (6-9V)
8	Barrel Jack (6-9V)



1	Vss
2	Vss
3	Vbb
4	Vbb
5	Vcc
6	Vcc
7	Vdd
8	Vdd

Figure 7.3: DC/DC module pinout

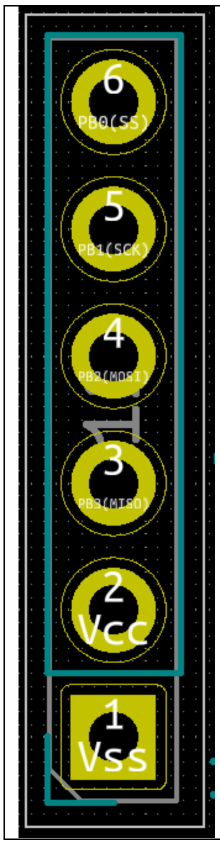
7.5 Pin assignment of the ISP-Header



1	MISO
2	Vcc
3	SCK
4	MOSI
5	Reset
6	Vss

Figure 7.4: ISP pinout

7.6 Pin assignment of the SD-Card-Header



6	CS
5	SCK
4	MOSI
3	MISO
2	5V / Vcc
1	GND / Vss

Figure 7.5: SD-Card module pinout

8 Troubleshooting

8.1 Overview and notes on suitable ISP programmers

8.1.1 Well-established programmers

The following programmers are tried and tested and can program an ATmega2560 without errors:

- Diamex USB ISP-Programmer for AVR, STM32, LPC-Cortex (Prog-S2) (Item nr. 102106), approx. 20 Euro
- Diamex USB ISP-Programmer für AVR, STM32, LPC, ESP32/8266 (Prog-S2E) (Item nr. 102106), approx. 20 Euro
- Diamex USB ISP-Programmer for Atmel AVR (Item nr.102104), approx. 18 Euro
- Tremex USB ISP-Programmer, approx. 20 Euro
- Pololu USB AVR Programmer, approx. 20 Euro
- Pololu USB AVR Programmer v2.1, approx. 12 Euro
- mySmartUSB MK2, approx. 30 Euro
- mySmartUSB MK3, approx. 40 Euro
- mySmartUSB light, approx. 16 Euro
  The programming seems to not work with AVRDUDE, use the enclosed “myAVR ProgTool”
- Atmel ICE, approx. 120 Euro
- Atmel STK500
- Diamex USB ISP-Programmer Stick for AVR (Artikel Nr. 102108, 2019+(?)), ca. 18 Euro
- misc. china clones of the AVR ISP MKii, approx. 16 Euro, driver installation required (works, but not recommended)
- Microchip PICkit4, approx. 100 EUR

Anyone looking for an easy-to-use programming device should take a look at the Diamex/Tremex programmer. If you are looking for an extensively configurable device, you should take a look at the Pololu. Both programming devices do not require drivers and are addressed under Windows through a virtual COM port.

Feedback that there were problems or the programming did not work:

- USBASP (different Models), approx. 10 Euro
- Bus Pirate. approx. 25 Euro
- Diamex USB ISP-Programmer Stick for AVR (Item nr. 102108, before 2019), approx. 18 Euro

Not tested:

- Pololu PGM03A (according to the manufacturer, the programmer does not provide a power supply, i.e. the board would have to be powered by USB).

8.1.2 Notes on some special ISP programmers

The programmer must be able to program an ATmega2560, which is not possible by all devices that promise to be able to program AVR chips.

USBtinyISP

E.g. the instructions for the USBtinyISP very clearly states

> *Works with any AVR ISP chip with 64K of flash (or less) - does not work with Atmega1281/1280/2561/2560*

At US$ 22, it is not even cheap.

USBASP

The experience with the USBASP is very bad. Google provides tens of results with problems related to the ATmega2560. But if you still want to try it out, you can find the first information here:

https://forum.arduino.cc/index.php?topic=363772.0

There is at least one indication that the firmware

> https://www.fischl.de/usbasp/usbasp.2011-05-28.tar.gz

```
avrdude.exe -p atmega8 -c usbasp -U flash:w:usbasp.atmega8.2011-05-28.hex:i -F -P usb
```

should work if the USBASP is operated in "slow-sck" mode (J3 must be set for this). The programming then takes an extremely(!) long time (over 40 minutes).

For ease of programming with the USBASP, you can use programming software called 'Khazama AVR Programmer' which is confirmed to program the ATmega2560 correctly. Note another common program named 'eXtreme Buner AVR' does not program the ATmega2560 correctly and should not be used for programming, although you can use it to program the fuses which does work ok.

A good guide on how to update the firmware of the USBASP using an Arduino can be found here: https://www.electronics-lab.com/project/usbasp-firmware-update-guide/

mySmartUSB light

mySmartUSB light seems to have problems with AVRDUDE:

> https://www.mikrocontroller.net/topic/272299

The programming process seems to work with the manufacturer's own software.

Other programmers:

In the instructions from AVRDUDE there is also this note:

```
m2560 ATmega2560 (**)
m2561 ATmega2561 (**)
```

```
(**) Flash addressing above 128 KB is not supported by all programming
hardware. Known to work are jtag2, stk500v2, and bit-bang programmers.
```

see https://www.nongnu.org/avrdude/user-manual/avrdude_4.html#Option-Descriptions

The programmer recommended earlier costs less than 20 EUR and is compatible with stk500v2.

8.2 The ATmega2560 is not recognized during programming

8.2.1 Check the supply voltage!

If the chip is not recognized, first check whether it is supplied with voltage at all, because not all programmers supply a voltage for the ATmega. Sometimes a jumper on the programmer has to be set accordingly.

If the programmer does not provide a supply voltage, the tester must be supplied with voltage via USB. In that case, however, it must be ensured that the programmer really does not deliver any, otherwise problems can arise.

8.2.2 Is the correct programmer specified at AVRDUDE?

If the supply voltage is ok and the chip is not recognized, please check whether the programmer has been correctly specified in AVRdude.

For many programmers who simulate a serial interface, **`-cwiring`** or **`-cstk500`** is correct. The serial interface is determined with the parameter "-P", e.g. **`-PCOM5`** for port 5. The serial interface speed can be set to 115200 bps with **`-b115200`**.

For USBASP, the parameter **`-cusbasp`** must be specified. "-P" and "-b" are omitted here.

The communication can be tested safely with the following command (only the current fuses are read out without making a change):

```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cstk500 -PCOM5 -b115200 -U
lfuse:r:-:i -U hfuse:r:-:i -U efuse:r:-:i
```

resp.

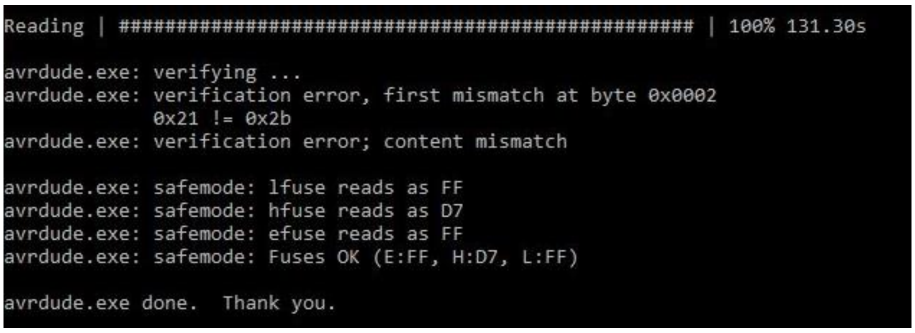
```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cusbasp -U lfuse:r:-:i -U
hfuse:r:-:i -U efuse:r:-:i
```

If the chip signature is not correct when reading the fuses (or even a different signature is displayed each time), it is probably due to the speed at which the ISP programmer is communicating with the chip. In this case, reduce the speed. Increase the parameter "-B" for the USBASP, e.g. to `-B4` or `-B8`.

The use of the USBASP is not recommended!

8.2.3 During programming a "verify error" appears

If a "verify error" appears at the end of the programming process, the programmer is very likely not suitable for programming the ATmega2560. The ATmega2560 is one of the few ATmega that have 256kb and many simple programmers only know 128kb as the upper limit (see also 128kb bug with USBASP).



```
Reading | ################################################## | 100% 131.30s

avrdude.exe: verifying ...
avrdude.exe: verification error, first mismatch at byte 0x0002
             0x21 != 0x2b
avrdude.exe: verification error; content mismatch

avrdude.exe: safemode: lfuse reads as FF
avrdude.exe: safemode: hfuse reads as D7
avrdude.exe: safemode: efuse reads as FF
avrdude.exe: safemode: Fuses OK (E:FF, H:D7, L:FF)

avrdude.exe done.  Thank you.
```

Figure 8.1: “verify error”

In this case, please check whether there is a firmware update for the programmer or switch to a suitable programmer. Old programmers only indicate general compatibility with the ATmega, which has not necessarily been the case since the ATmega2560 was released.

Some crystals supply a voltage swing that is too low, so that the processor, which expects a "Low Power Oscillator" ("`lfuse`" is programmed to `0xff`), does not function correctly. In this case the "Full Swing Oscillator" should be tried. For this, the "`lfuse`" is programmed to `0xf7`:

```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cstk500 -PCOM5
-U lfuse:w:0xf7:m -U hfuse:w:0xdf:m -U efuse:w:0xff:m
```

For the "`hfuse`" the following applies again: `0xdf` = the configuration is deleted during a firmware update, `0xd7` = the configuration is retained after a firmware update.

In very rare cases the crystal or the 22p capacitors are the fault. If the "verify error" occurs with one of the recommended programmers, these three components should be replaced. Before doing this, you should try to operate the ATmega not in the "Low Power Crystal Oscillator", but in the "Full Swing Crystal Oscillator". More on this in section 4.1.

8.2.4 Check the connection to the ATmega!

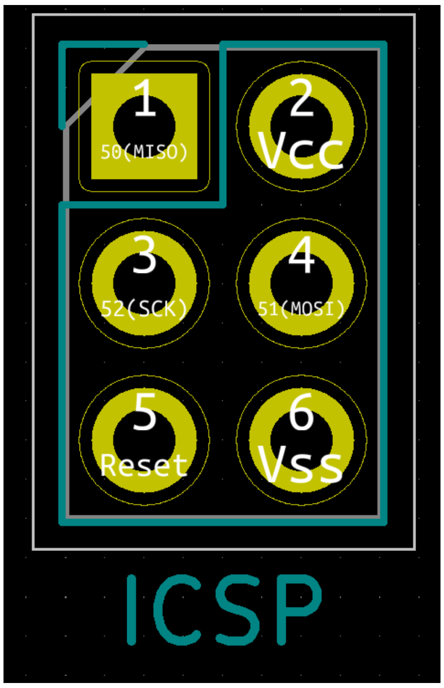
If the ATmega2560 has already been pre-assembled, this test is not necessary!

If communication is still not possible, the chip may not have been soldered correctly (cold solder connection) or may have a short circuit.

With a continuity tester, the connection can be checked as follows:

- Pin 1 of the header goes to pin 22 of the ATmega2560 (MISO)
- Pin 2 of the header goes to pin 10 of the ATmega2560 (Vcc)
- Pin 3 of the header goes to pin 20 of the ATmega2560 (SCK)
- Pin 4 of the header goes to pin 21 of the ATmega2560 (MOSI)
- Pin 5 of the header goes to pin 30 of the ATmega2560 (Reset)
- Pin 6 of the header goes to pin 11 of the ATmega2560 (Vss)

The header assignment is as follows:



The assignment of the ATmega2560 is as follows (the counting starts at the mark counter-clockwise):

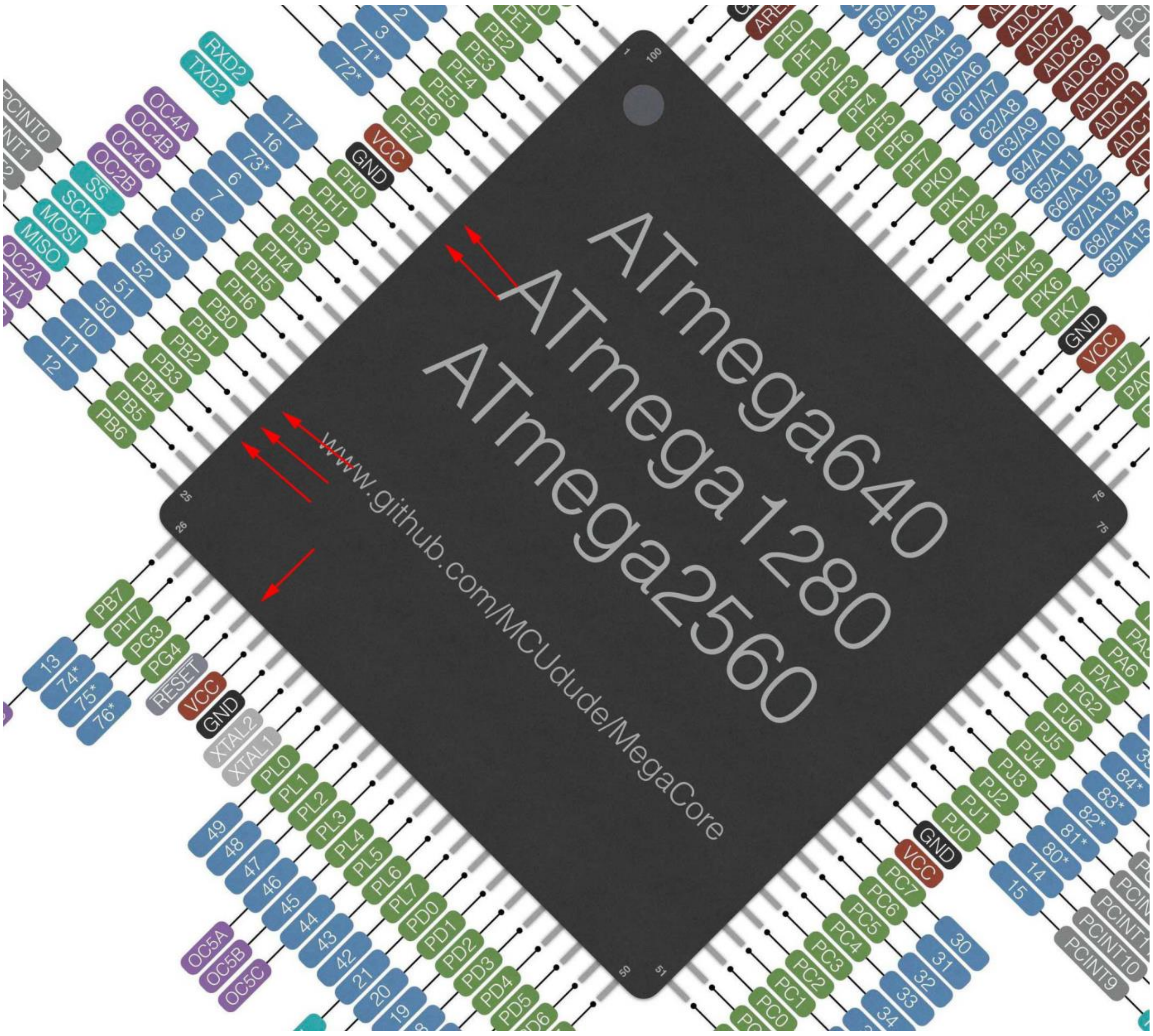


Figure 8.2: ATmega2560 pinout

Please also test whether there is a connection to a neighboring pin. Is everything is ok and if communication is still not possible, the chip could be defective. Was the chip heated too much when soldering in? Did the chip come from a reliable source (be careful with dealers in China)?

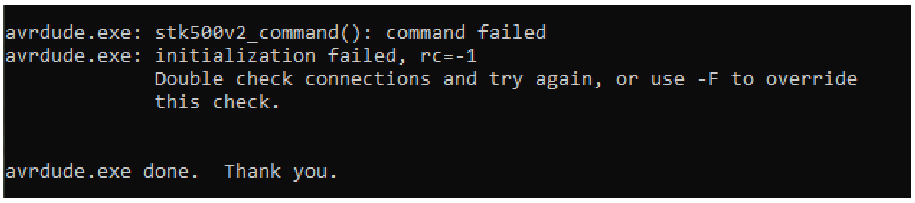
8.2.5 “Help! I can no longer program the ATmega2560!”

Was it possible to address the ATmega2560 when programming the fuses or when programming the firmware, but an error occurred and now the chip can no longer be addressed?

Basically, two types of errors are possible:

1. By setting the fuses, the internal 1MHz clock is switched to the external crystal. If the ATmega2560 can no longer be addressed immediately afterwards, the crystal or one of the 22pF capacitors may be defective.
2. If you started to flash the firmware after setting the fuses, then an error occurred and afterwards communication with the ATmega2560 is no longer possible, the fuses may have been affected.

If the following error message appears: First of all, don't panic! It is almost impossible to lock yourself out of your ATmega2560 by software.



```
avrdude.exe: stk500v2_command(): command failed
avrdude.exe: initialization failed, rc=-1
             Double check connections and try again, or use -F to override
             this check.


avrdude.exe done.  Thank you.
```

Figure 8.3: AVRDUDE – No communication with the ATmega2560

Presumably the fuses are in a state that the external crystal is not used and the internal 1 MHz is not available either (e.g. if all fuses have been deleted and an external clock is required), i.e. the chip lacks the clock source.

You can set the fuses again quite easily: All you have to do is apply a clock source to pin 34 (XTAL1) of the ATmega2560. It sounds more complicated than it is.

First you need a clock source (when you have a Diamex Prgorammer or another programmer that supports a clock, jump to the end of this chapter). Here you can use an Arduino Uno/Nano or Mega, which you program with the following program:

```
#include <avr/io.h>

void setup() { }

void loop() {
    DDRB = 0xFF;
    while(1) {
        PORTB ^= 0xFF;
    }
}
```

This program generates a clock (1.59 MHz) on pins 8 to 13 on the Arduino UNO/NANO and pins 10-13, 50-52 on the Arduino MEGA. If this program is loaded into the Arduino, the LED connected to pin 13 lights up only very weakly due to the constant on/off processes.

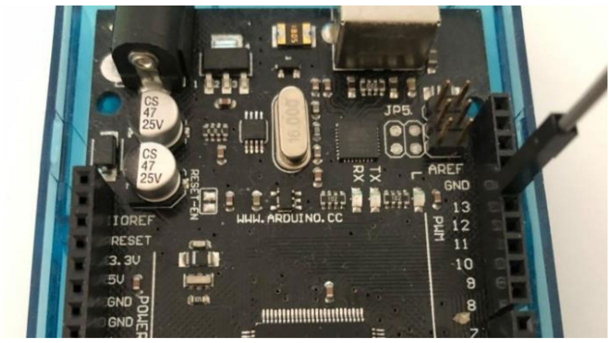


Figure 8.4: Pin 13 on the Arduino Mega

Now you connect pin 13 to XTAL1. The right leg of the 1 MOhm resistor below the crystal is ideal here. The connection must not have any wobbles and must not come into contact with the crystal. Do not forget to connect GND (ground) to GND of the tester (usually it works that way if both devices are operated on the same computer).

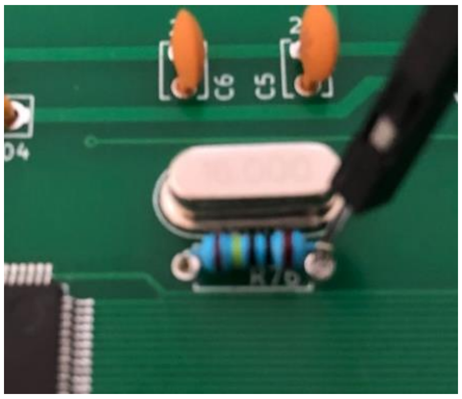


Figure 8.5: XTAL1

Usually, the crystal does not have to be unsoldered because the 5V clock of the Arduino is much stronger than the weak signal of the crystal. However, the firmware should not be fully programmed in this way, but rather the fuses should only be correctly programmed "quickly" so that the programming process can take place again without the external clock (with the existing crystal).

An alternative source for the clock is to wire up a Crystal Oscillator. These are usually a metal can 13mm x 13mm or 13mm x 20mm and contain additional parts that make the crystal oscillate when powered on. Using the crystal oscillator datasheet or a pinout as reference, simply wire the Vcc and GND pins to the J4 header (located between the DC/DC module and the LCD) and wire the output of the oscillator to that same 1M Ohm resistor shown in the picture above.

Once this connection has been established and the ISP programmer is connected, you can now try again to set the fuses (the programmer and port must be adjusted accordingly).

```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cstk500 -PCOM5
-U lfuse:w:0xf7:m -U hfuse:w:0xd7:m -U efuse:w:0xff:m
```

This process does not necessarily work the first time it is called, repeat it several times if necessary. In some rare cases the 16MHz crystal must also be removed to get the fuses programmed successfully. If the fuses have now been set correctly, remove the extra crystal oscillator/Arduino and program the firmware using the normal method.

Programmer with clock support (e.g. Diamex)

If the ISP programmer used offers a clock output, this can be used very easily.

With Diamex, simply specify the parameter `-xfosc=1MHz` when setting the fuses and connect the clock from pin 3 (OSC) of the 10-pin connector to XTAL1 (see above).

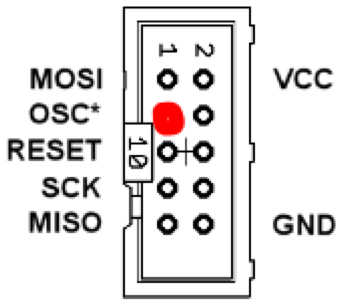


Figure 8.6: Clock on the 10-pin ISP connector

The comment line therefore reads:

```
avrdude.exe -C"avrdude.conf" -xfosc=1MHz -B0.5 -patmega2560 -cstk500
-PCOM5 -U lfuse:w:0xf7:m -U hfuse:w:0xd7:m -U efuse:w:0xff:m
```

8.2.6 Use of COM10 and higher (Windows)

If AVRDUDE is to be operated on COM10 and higher, the interface possibly must be specified as follows:

For example for COM12:

```
-P \\.\com12
```

8.3 Problems with the display

8.3.1 No text appears on the display

Is the contrast set correctly? This can be changed with the potentiometer.

8.3.2 The display is fairly dark

A 220 Ohm resistor is located on the circuit board next to the display. This ensures that even displays that have no series resistor for the backlight can be operated safely (usually these displays should also work with 100 Ohms). With all other displays, the resistance can be completely eliminated. It is easy to see on the back whether there is a suitable series resistor.

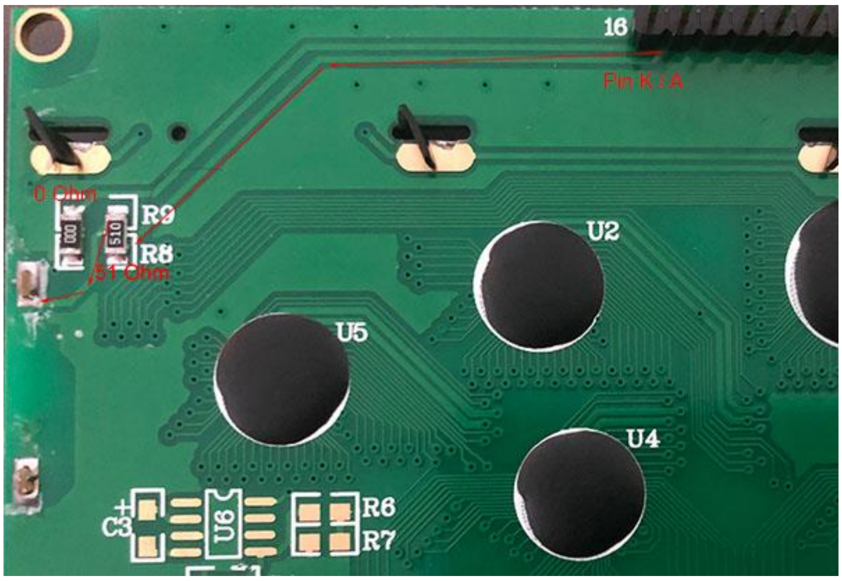


Figure 8.7: 51 Ohm series resistor to the backlight

When you see a series resister you can use a wire bridge or 0 Ohm resistor instead.

8.3.3 The display of an OLED is quite dark

When an OLED has been installed and the display is unusually dark, it may be because pins 15/16 are not - as described in the data sheet - not connected ("nc"). Normally, pins 15/16 should not be connected on an OLED, as these are usually used for the backlight:

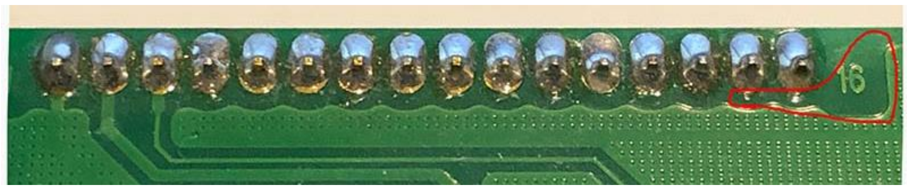


Figure 8.8: Pins 15/16 not connected

With a Winstar OLED, however, these pins are used even though they are marked as "nc" in the data sheet:

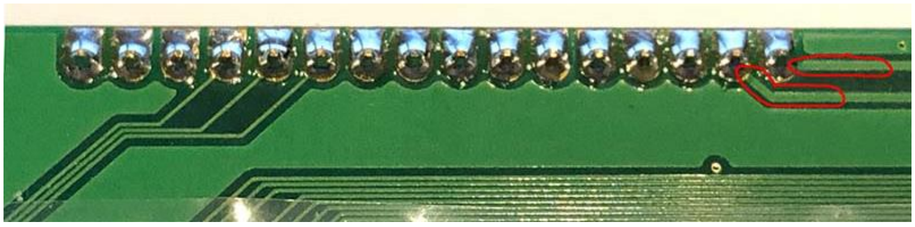


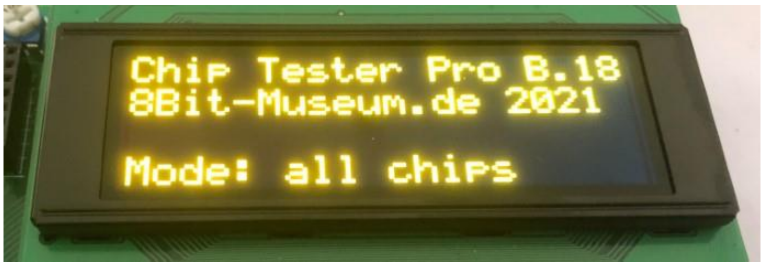
Figure 8.9: Pins 15/16 connected

In this case, simply shorten the two pins (or cut the two traces) so that they no longer have any contact with the socket. The display should then show the text with maximum contrast.

8.3.4 The displayed text of an OLED is incorrect (1)

Unfortunately, not all OLEDs are fully compatible with the HD44780 controller. Most(?) displays seem to work without any problems, but there are also OLEDs that produce incorrectly displayed text.

Normally, the power-on message should look like this:



```
Chip Tester Pro B.18
8Bit-Museum.de 2021

Mode: all chips
```

Figure 8.10: Displayed text is ok

After several resets, lines can suddenly be mixed up:

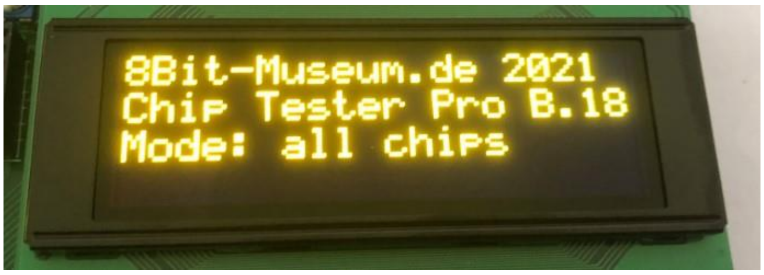


Figure 8.11: lines are switched

After a reset, the display can rarely display an incorrect text:

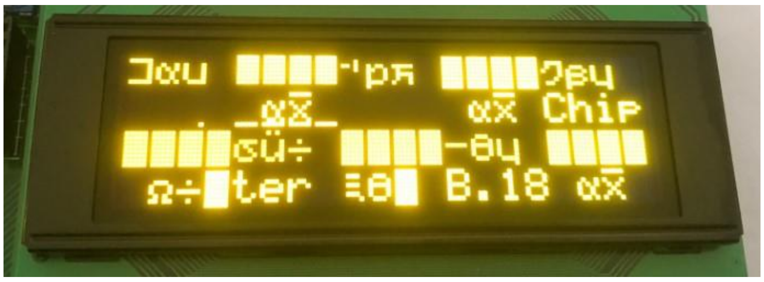


Figure 8.12: incorrectly displayed text

The problem probably only occurs after a reset. After restarting by switching the Tester off and then on again, the problem is gone again. It is not a straightforward timing problem and no known solution has yet been found.

The error occurs for example with OLEDs from Winstar and clones (e.g. Newhaven) but not with all, probably depending on the board revision.

A small intervention is necessary to operate the display without errors. On the back of the OLED, a thin wire is connected to the reset line of the controller. The left connection of the 10k resistor is suitable for soldering the wire.

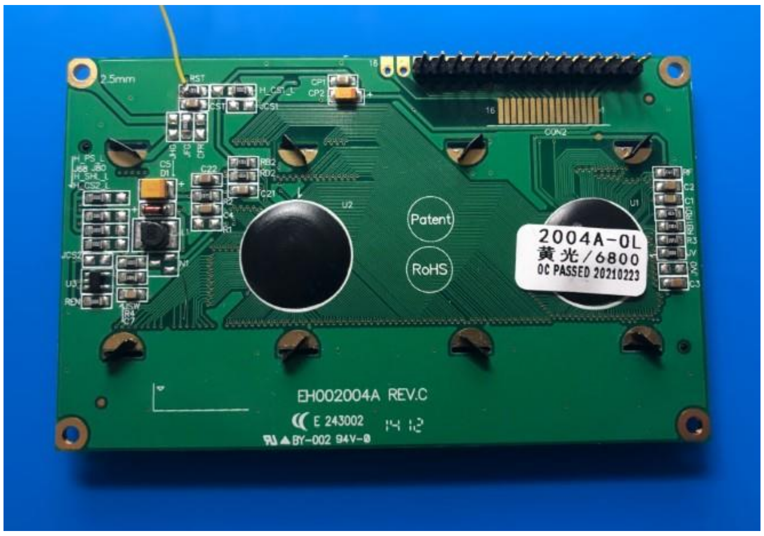


Figure 8.13: backside of the OLED PCB

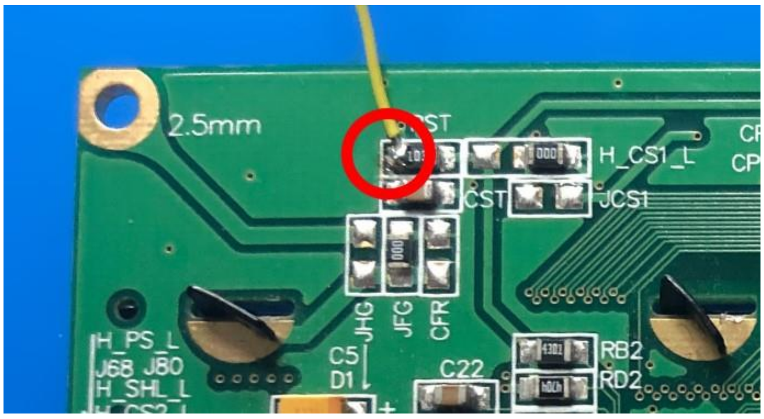


Figure 8.14: left side of the 10k resistor

The wire is then connected to the reset line of the Tester. The ISP connector is suitable for this (seen from the front, the lower left connection; from the rear, the lower right connection).

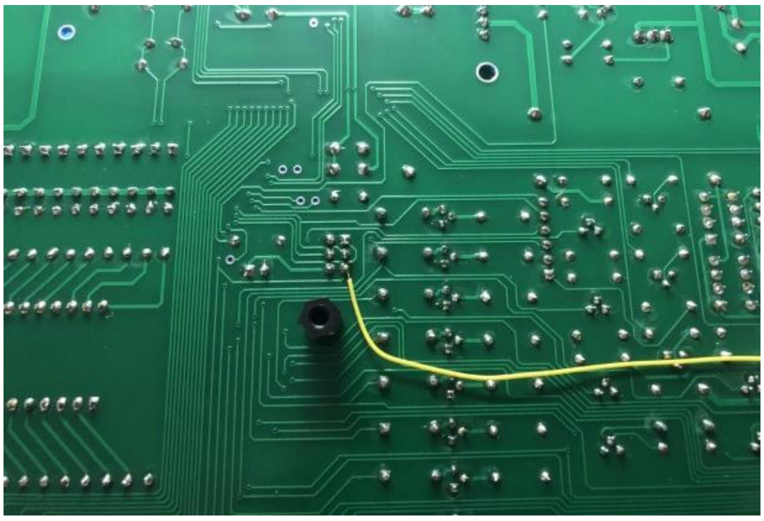


Figure 8.15: Reset at the ISP connector

After this modification, after a Reset, there should be no more mixed up lines or a faulty screen.

8.3.5 The displayed text of an OLED is incorrect (2)

In order for a Winstar 4x20 OLED display to work on the RCT, the board must be configured with jumpers as shown in the picture. On delivery, it can happen in rare cases that these are configured differently. In this case, the display remains dark or only “strange” characters appears.

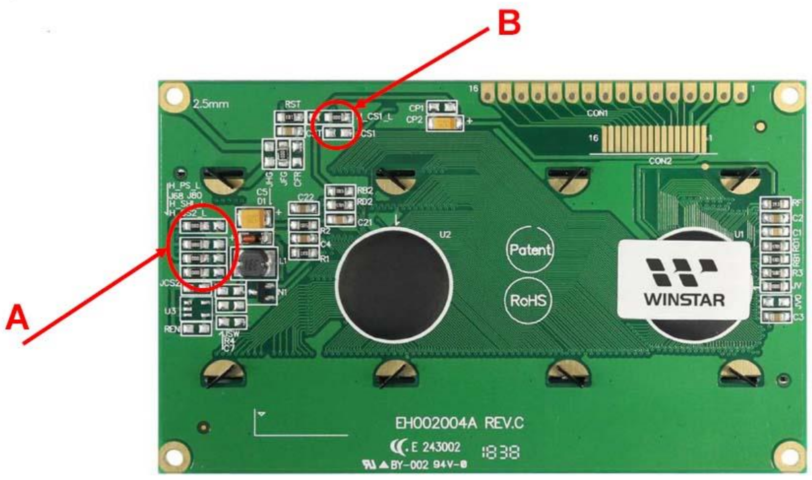


Figure 8.16: correctly set jumpers

The jumpers must be set as follows for this display type:

A: All four bridges on the left.

B: Bridges on H_CS1_L; CS1 no bridge

8.4 Problems with the DC/DC module

8.4.1 The green LEDs of the DC/DC module do not light up, and the display may be dark

Error on the DC / DC module

It could be that the DC/DC module is not delivering the supply voltages. Please remove the DC/DC module from the tester and connect the pins "Vss" and "Barrel Jack" on the left with 5V to 9V (Vss = minus, Barrel Jack = plus). The voltages 5V (Vss/Vcc), -5V (Vss/Vbb) and 12V (Vss/Vdd) should be measured on the right side. All three LEDs should also light up. If the LEDs do not light up or there is no voltage, then a component on the DC/DC module is defective. If the LEDs do not light up when the voltages are present, there could be a short circuit.

Error on the main board

There could be a short circuit on the board. In order to limit a possible short circuit, please measure the resistance between Vss/Vcc, Vss/Vbb and Vss/Vdd (ideally on the socket header J4 below the potentiometer). If the reading is close to 0 ohms, the problem is likely in one of the following areas:

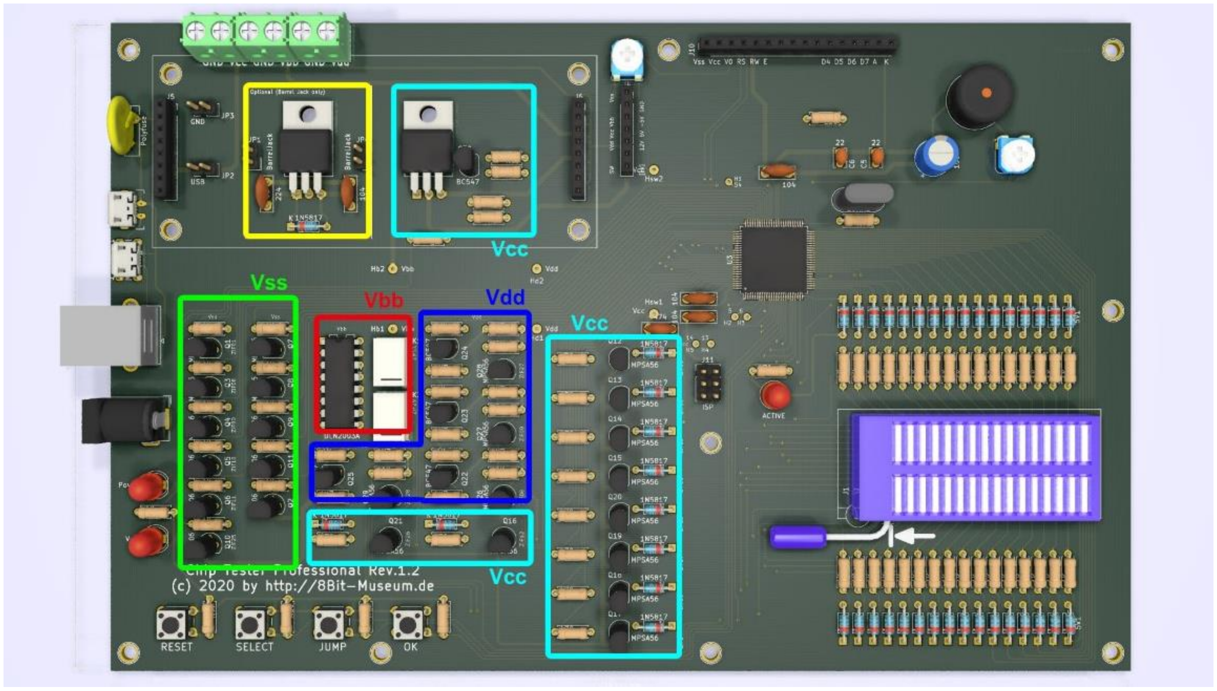


Figure 8.17: PCB overview

Please check the transistors for solder shorts in particular. Other pads that are close together can also cause a short circuit if there is too much solder used.

It has also happened that one of the capacitors was faulty and had an internal short circuit (please measure the resistance of the capacitors here, a faulty capacitor very likely has a resistance close to 0 Ohms).

8.5 Other Problems

8.5.1 The -5V (Vbb) is not installed, the -5V can be measured on the power supply

The relays click when e.g. the test for 4116 ICs is selected?

Are the relays soldered the right way round? Is the ULN2003 correctly orientated? Check this once with the Tester (Misc Logic Tests) or replace it with a known good chip.

8.5.2 Tests fail very often

If the tester is supplied with voltage via USB, please check it with a multimeter. For this, the voltage "Vcc" can be tapped on the header between the DC/DC module and the display.

The voltage "Vcc" should ideally be 5V. Down to 4.8V are just acceptable, voltages below 4.8V can cause problems. In this case, try another USB power supply or supply the tester with voltage via the barrel jack (7-9V).

If the supply voltage appears to be ok (and the display shows the main menu), it can be practically any of the more than 200 components that can produce a fault.

1. The Zener diodes protect the ATmega from high voltages. A defective Zener diode can weaken the signal level or create a short circuit to GND.
2. Resistors are rarely defective, but it can happen. Interruptions can be found quickly with an Ohmmeter.
3. When an incorrect resistance value has been assembled, a supply voltage may no longer be switched correctly. E.g. a 4.7k Ohm resistor as the Base resistance instead of the 470 Ohm leads to the transistor no longer switching fully and the voltage drop Uce to be too high. The same applies to incorrect values for the pull-up/pull-down resistors.
4. An incorrect resistance value of e.g. 4.7k Ohm instead of 4.7 Ohm for R57 (gate resistor on the MOSFET) leads to Vcc failing completely or collapsing with low load.
5. A defective subminiature relay can usually be recognized by the missing click when testing a 4116 (relay #1) or 2708 (relay #2) or by the permanent clicking of a relay. The latter can also be due to a defective ULN2003A.
6. Cold solder joints are more difficult to find. If necessary, briefly re-solder matt looking soldering joints in order to exclude a high-resistance connection.
7. If too much solder was used in the area of the transistors, short circuits can easily occur between the pins. These are easy to check with a magnifying glass.
8. The crystal is also a common error. More on this in section 8.5.3.
9. A defective capacitor often leads to a short circuit between Vcc and GND.
10. Defective transistors are the most difficult to identify. If, for example, SRAMs and many DRAMs can be tested without problems, but a special DRAM type cannot, this can indicate that a transistor that is responsible for Vcc, Vss or Vdd is not switching correctly. Here it should be checked in which pins the problematic memory type differs.

However, if only one IC of a special type always fails (e.g. a uPD416 cannot be tested, but a TMS4116 is tested without any problems), it should be assumed that this module (or the entire batch) may no longer be tested due to aging effects.

8.5.3 During a test a reset occurs or the firmware behaves strangely

If a sudden reset occurs during testing, if IC tests fail repeatedly, or the firmware behaves strangely, there are several errors possible.

1. Problems with the clock

Symptoms: Resets that occur suddenly, firmware "jumps" to other program positions (tests do not abort, display shows unusual output)

Cause of error: The crystal is probably supplying a voltage swing that is too low.

Corrective action: Instead of the "Low Power Crystal Oscillator", the "Full Swing Crystal Oscillator" of the ATmega should be set on a trial basis (see sec. 4.1). If that doesn't work, the crystal and the two 22p capacitors should be changed.

2. Insufficient voltage supply

Symptoms: A certain IC cannot be tested (possible with memory tests and logic tests).

Cause of error: The error can be due to a too low / faulty supply voltage if the error always occurs with a certain IC type, e.g. a 14-pin. logic IC or a 32pol. memory IC.

Corrective action: The self-test should be done first. If errors occur here, possibly always on the same pin, it is very likely that a component is defective or there is a cold solder joint. The error can be on the main board as well as on the DC/DC module. In the event of a short circuit, the voltage supply to the processor drops so much that it either performs a reset or no longer works correctly.

If the error occurs on different pins, the supply voltage is probably not reliable. Here the fault could be on the DC/DC module or, if Vcc is mainly affected, then on the MOSFET or the surrounding components (Q30, Q31, R54-R57).

3. Other mistakes

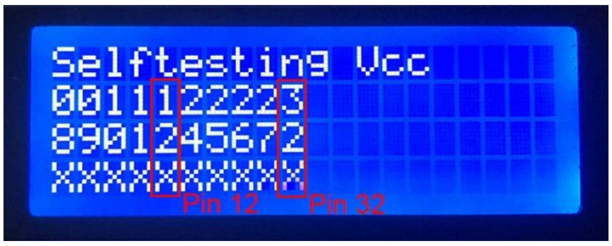
Symptoms: A specific memory or logic IC cannot be tested.

Corrective action: The memory or logic IC can of course be defective. If the error occurs with several ICs of the same type, please check whether the IC has special technical requirements. These include: Power supply (e.g. for NVRAMs), required current at the inputs for bipolar ICs (e.g. standard TTLs).

8.6 Self-test

A simple self-test can be called up via the menu, which checks the supply voltages Vcc, Vdd and Vss. Vbb cannot be tested for technical reasons.

The display shows the status of the individual ZIF pins. An "X" means that the supply voltage could be switched correctly at this pin. A "1" means that the voltage (Vcc, Vdd, Vss) is permanently present, a "0" that there is no voltage.



```
Selftesting Vcc
0011122223
8901245672
XXXXXXXXXX
```

Figure 8.18: Testing Vcc, interpreting the display

When the Tester is used without any DC/DC module, the self-test will always fail (Vcc and Vss should show “X”, Vdd should show “0”). So no panic.

No IC may be inserted. This would falsify the test result and possibly damage the IC.

No decoupling capacitor may be activated (no jumper may be set to the right of the ZIF socket).

8.6.1 Problem: A "0" is displayed for all voltages (Vcc, Vdd)

Are the three wire bridges soldered in underneath the DC/DC module?

8.6.2 Problem: Only 0's are displayed at 5V (Vcc)

Please check whether 4.7 Ohm has been soldered in for the R57. Possibly a 4.7k Ohm was used here. The MOSFET (IRF5305) may also be defective.

8.6.3 Problem: A single “1” or “0” is displayed for Vcc/Vdd/Vss

The self-test works as follows:

Vcc/Vdd is turned on
It should now be possible to read a logical H → Status A

Vcc/Vdd is turned off
It should now be possible to read a logical L → Status B

When

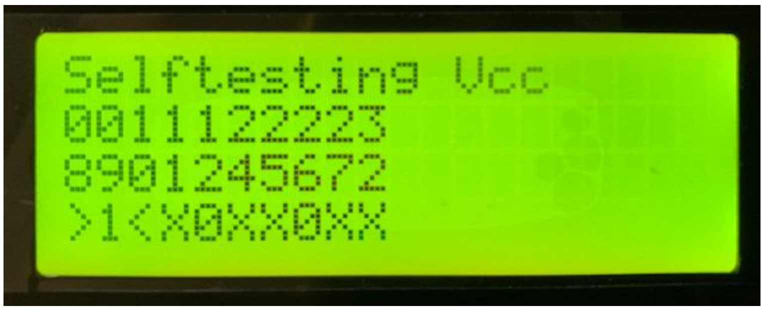
- A=HIGH and B=LOW → Test OK, Display “X”
- A=HIGH and B=HIGH → Test Fail, Display “1” (Signal stays High)
- A=LOW and B=LOW → Test Fail, Display “0” (Signal stays Low)
- A=LOW and B=HIGH → Test Fail, Display “-”

The same applies to Vss, up to firmware v.15 a “1” was displayed if Vss could be switched (i.e. an L was read), but this could not be “switched off” (i.e. an L was still read). From firmware v.16 the output has been adapted to the signal level, i.e. a "1" is displayed if a logical H was read despite activation of Vss.

If a "0" is displayed when testing Vcc/Vdd, the associated Zener diode may also be defective (short circuit).

8.6.4 Multiple Vcc and Vdd selftest errors are displayed

When several errors occur during the selftest of Vcc and Vdd, then the fuses are probably not set correctly. The error pattern is expressed in the following output (can also be slightly different):



```
Selftesting Vcc
0011122223
8901245672
>1<X0XX0XX
```

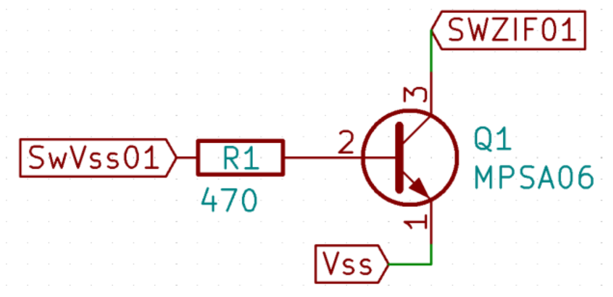
Figure 8.19: Failed to test Vcc

When the fuses are not set correctly and the JTAG interface is enabled, the pull-up resistors on pins PF7 (TDI, Pin 26), PF5 (TMS, Pin 9), and PF4 (TCK, Pin 12) will be activated even if a reset occurs.

Set the fuses and the error is fixed.

8.6.5 How it works

Vss is switched by NPN transistors. If the self-test shows that Vss cannot be switched, the problem may be a defective transistor (MPSA06). However, a cold solder joint or a defective resistor is also possible. The transistor belonging to the ZIF pin is noted on the board.



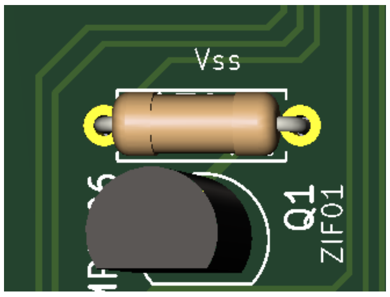


Figure 8.20: Switching of Vss, identification on the board

Vcc is switched through the PNP transistors (MPSA56) in the middle of the board. If Vcc cannot be switched, the transistor could be defective. If Vcc is not present at all, the diode may be defective or there may be a cold solder joint.

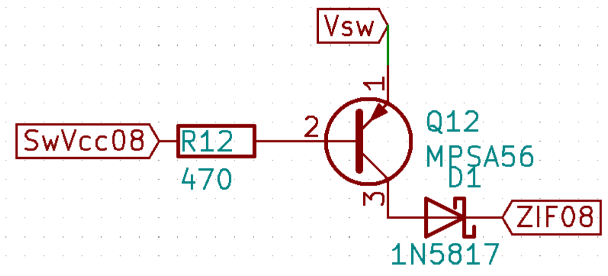


Figure 8.21: Switching Vcc

Vdd is switched by the PNP transistors (MPSA56) in interaction with the NPN transistors (BC547) in the first third of the board. If Vdd cannot be switched, a transistor could be defective. The problem can also be with a wrong or defective resistor or there can be a cold solder joint.

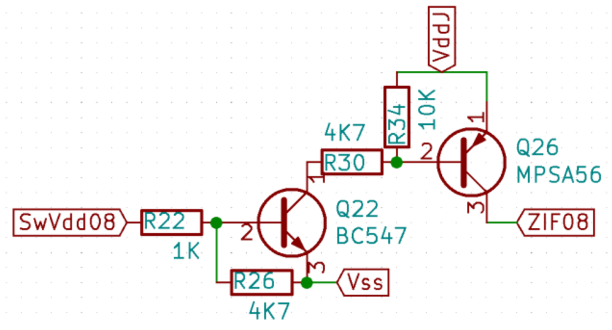
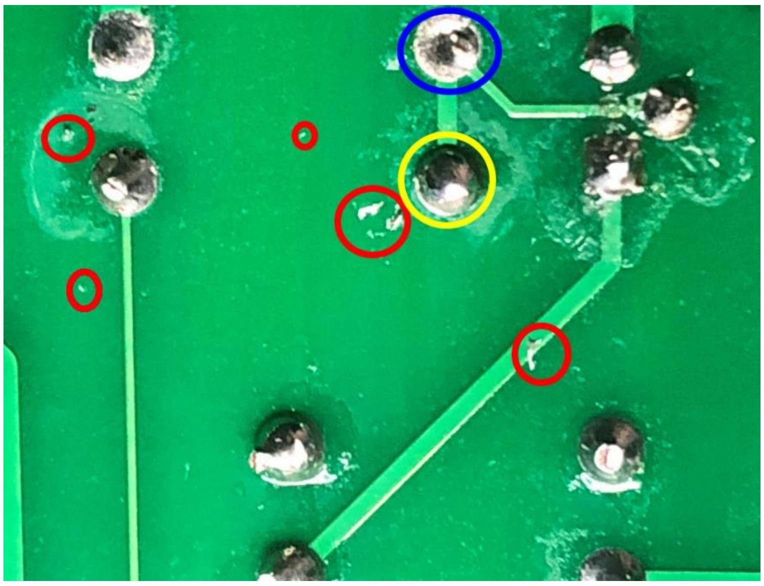


Figure 8.22: Switching Vdd

8.6.6 Bad soldering

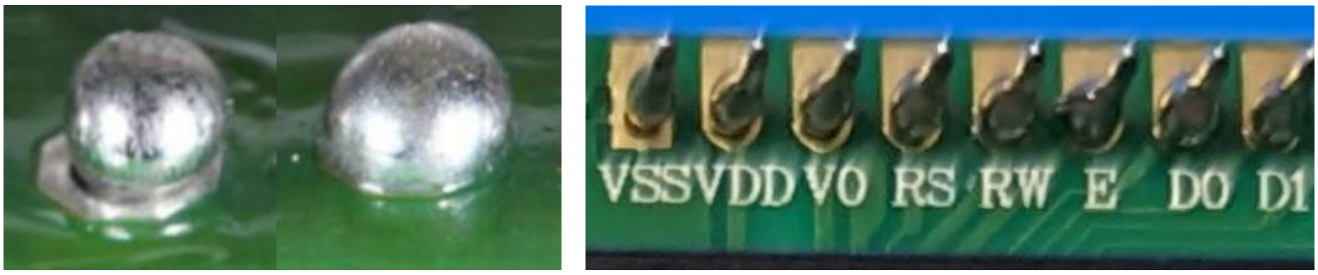
Please make sure that not too much, but also not too little, solder is used. In addition, if the solder is poor and/or the soldering tip is too hot, solder splashes can occur, which can cause short circuits.

The following pictures show a few examples:

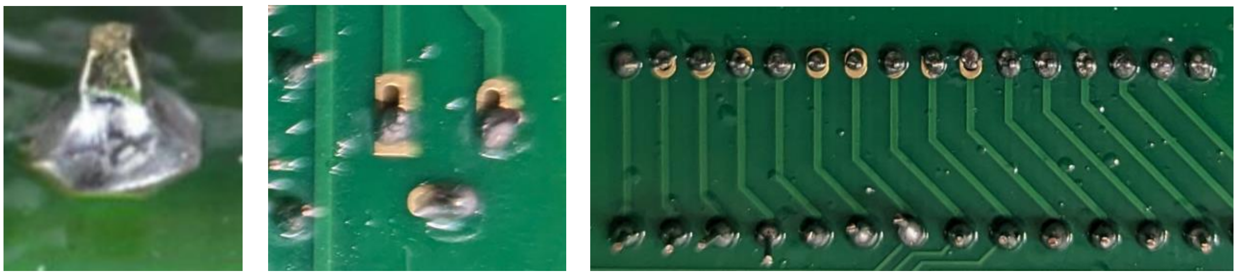


red: splashes of solder that can cause short circuits

yellow: too much solder (ball made of solder); blue: too little solder

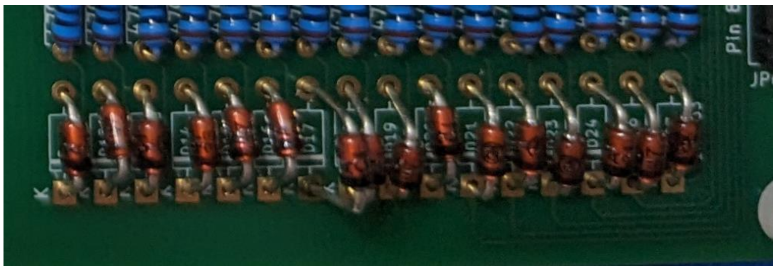


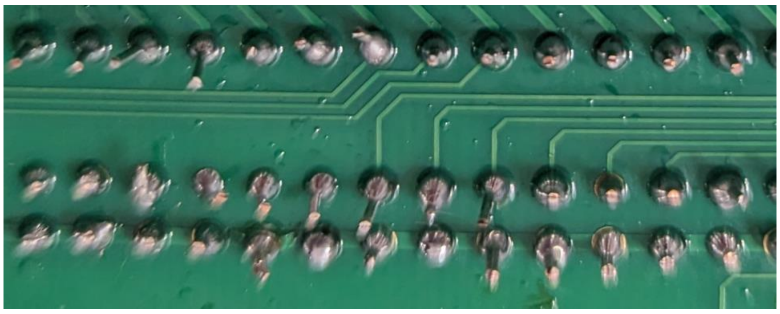
too much solder (ball made of solder), no contact

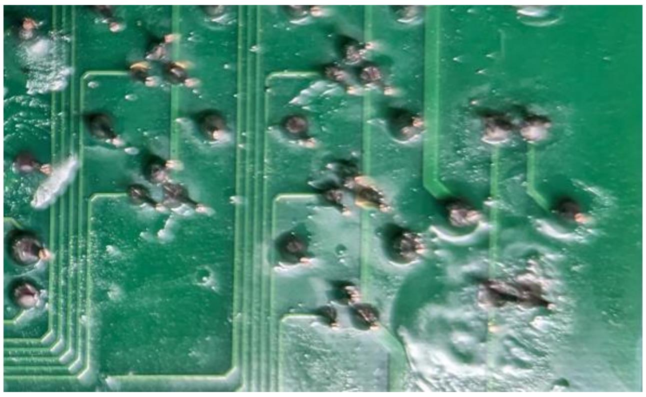


too little solder, no contact

If the soldered connection looks dull there could also be a cold soldering point (i.e. without contact, shown in the picture above, left joint not connecting properly).


Not pushed through far enough, short circuit possible

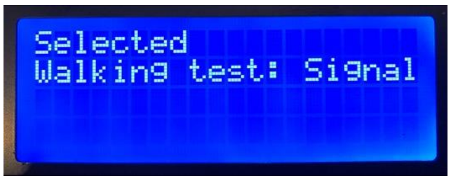
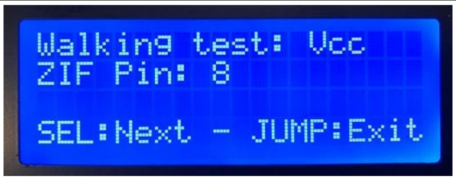

Pins not well shortened, short circuit possible

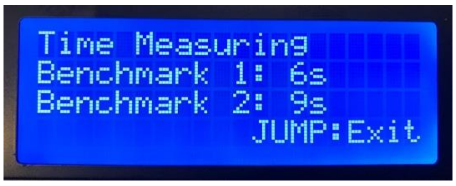

Circuit board not cleaned, short circuits possible

8.7 Diagnostic-Software

There is (simple) diagnostic software in the archive `upload-tester-pro-test.zip`. Programming is done as described in section 4.

A test can be selected in the menu with SELECT and OK. With JUMP, a test is usually completed.

Menu item	**Description**	**Picture**
Selftest Vcc,Vdd,Vss	The tester tests the supply voltages Vcc, Vdd and Vss. Vbb cannot be tested for technical reasons.	Selected Selftest Vcc,Vdd,Vss
Selftest LOW/HIGH	It is tested whether a LOW is read without pullups and a HIGH with pullups.	
Walking test: Signal	This test successively applies a signal to each individual pin of the ZIF32 socket. A pin is switched with SELECT.	 Selected Walking test: Signal
Walking test: Vcc	This test successively applies the supply voltage Vcc to the possible pins of the ZIF32 socket. A pin is switched with SELECT.	 Walking test: Vcc ZIF Pin: 8 SEL:Next - JUMP:Exit
Walking test: Vss	as above, only for Vss	
Walking test: Vdd	as above, only for Vdd	
Walking test: Vbb	as above, only for Vbb	
All ON: Vcc	This test switches all possible pins of the ZIF32 socket to Vcc at the same time.	Selected All ON: Vcc
All ON: Vss	as above, only for Vss	
All ON: Vdd	as above, only for Vdd	
All ON: Vbb	as above, only for Vbb	
Input Test: Low	When this test is started, any pin can be connected to Vss. The pin is shown in the display.	Selected Input Test: Low
Input Test: High	When this test is started, any pin can be connected to Vcc. The pin is shown in the display.	Selected Input Test: High

Menu item	Description	Picture
Benchmark	This test performs a simple benchmark. It can be used to determine whether the clock of the ATmega2560 is set correctly.	 Time Measuring Benchmark 1: 6s Benchmark 2: 9s JUMP:Exit

A logic tester can be used to test the signals quickly (be careful with Vdd: 12V). If you don't have a logic tester, you can quickly build a simple tester yourself:

All you need is an LED and a 1k Ohm resistor. The resistor is soldered directly to the LED (cathode, short pin or flattened LED side).

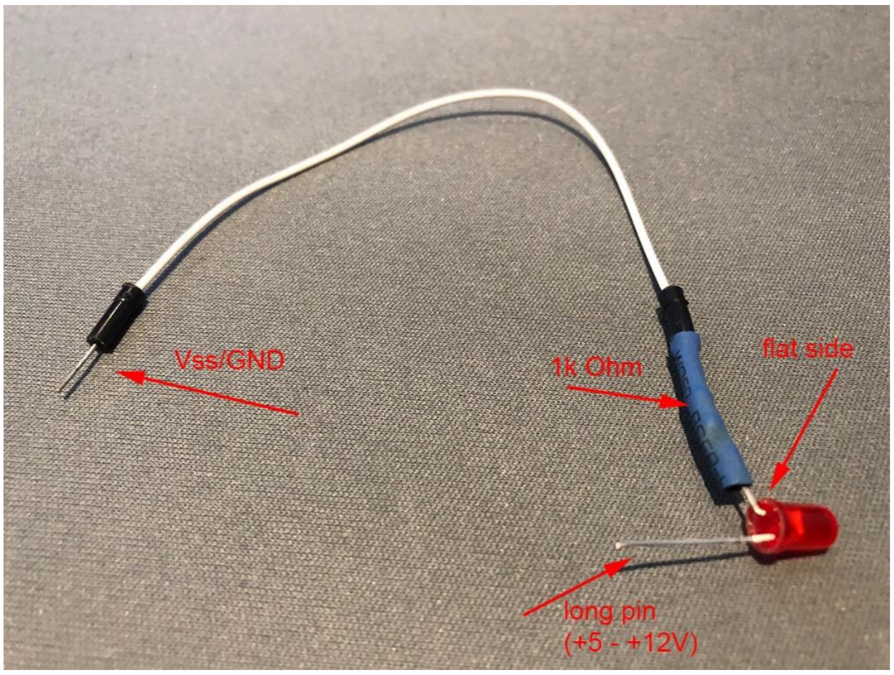
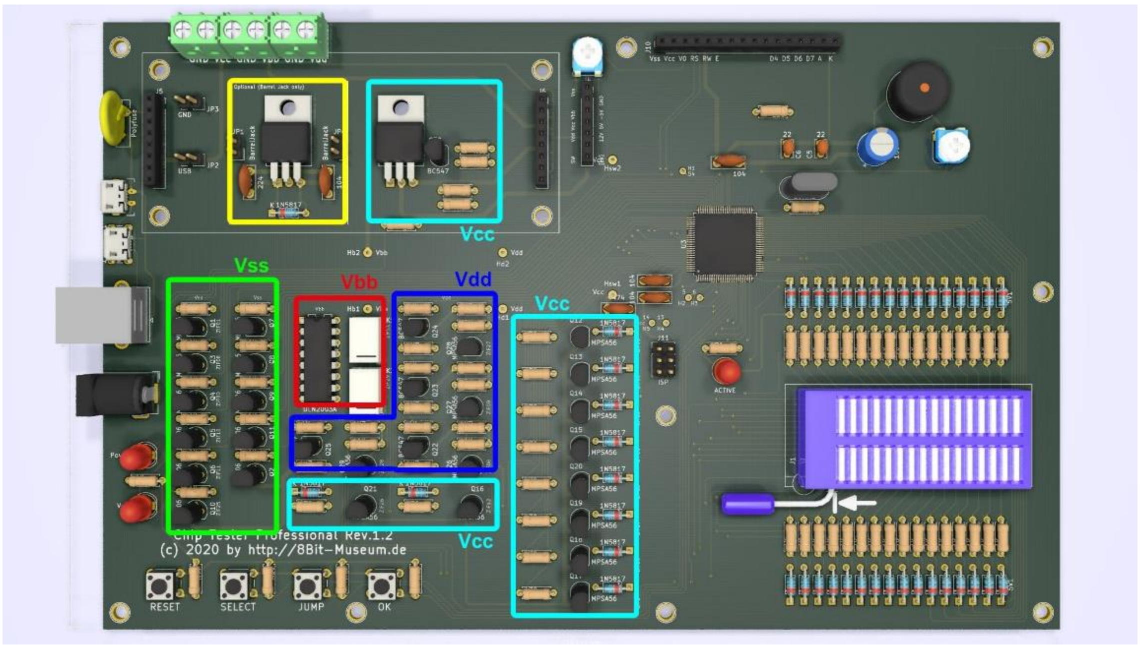


Figure 8.23: A simple logic tester

To test a positive signal, plug the connector into the "Vss" header and hold the long connection of the LED (the anode) to a pin on the ZIF32 socket. With this “tester” the presence of Vdd can also be checked.

To test Vbb (-5V), plug the long connection of the LED (anode) into the "Vss" header and test the voltage with the pin.

The transistors are labeled with the pin number of the ZIF socket. The following areas are responsible for Vss, Vbb, Vcc and Vdd:



8.8 Still-Alive Mode

The firmware (from v25) includes a small test routine that can be used to check whether the processor is working. This makes sense if there is no response after assembly and the troubleshooting procedures described did not work.

For the still-alive mode, the only requirement is that the clock works, so at least the 16 MHz crystal and the two 22pF capacitors must be in working order.

If the fuses are not set correctly or the crystal/capacitors are not assembled, an external clock can also be used (e.g. via the programmer's oscillator pin).

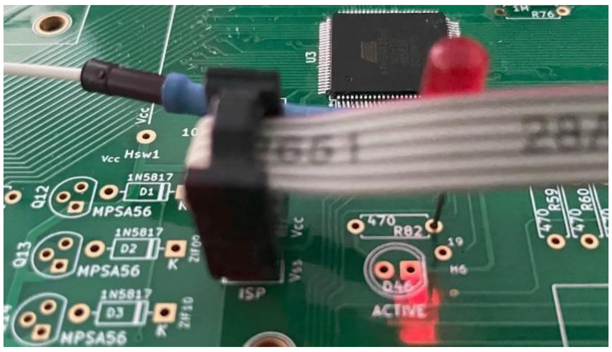
If there is a clock and the board is already assembled, nothing else needs to be considered. The mode is called up if the “Select” button is held down for at least five seconds after the supply voltage has been applied.

The Still-Alive Mode:

- fills the display (if connected) completely with “*”,
- plays a short beep every second and
- flashes the Active LED four times per second.

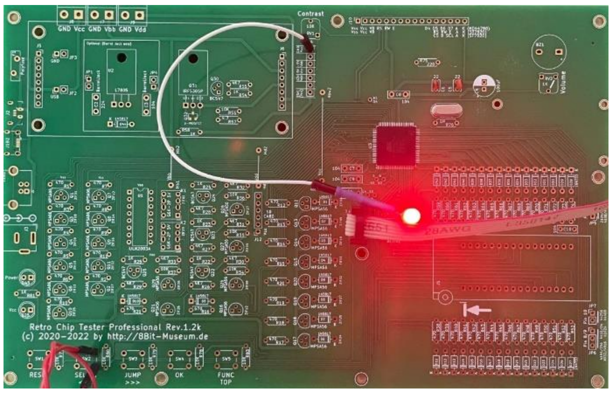
If the board is still unassembled, an Active-LED can be connected as follows:

- assemble this together with R82, or
- connect an LED with a series resistor (see previous chapter) as shown in the following picture (Vss and right contact of R82).



A wire bridge can be used instead of the Select button. The supply voltage can be provided via the ISP programmer (usually it is permanently available there).

If the board is still unassembled, a reset may have to be triggered after applying the supply voltage by briefly holding a wire bridge to the contacts of the Reset button.



Shortly after a reset, the LED should start flashing.

9 Appendix: Examples of programming the ATmega

This chapter presents methods for programming the ATmega2560 that were made available by users. Unfortunately, I cannot provide any support for this.

Many thanks to

- Tom64 for providing the guide to MacOS 10,
- Joao for providing the guide to Olimex AVR-ISP-MK2 programmer.

9.1 DIAMEX PROG-S2 ISP-Programmer for AVR / STM32 / NXP / LPC

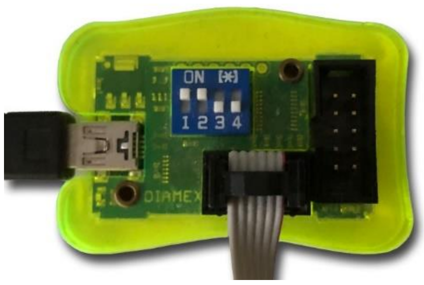
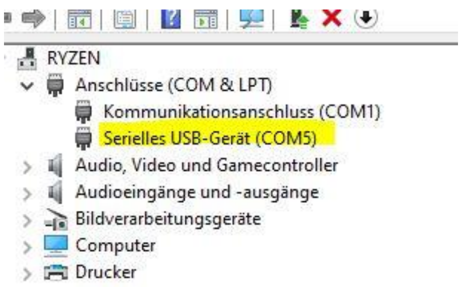


Figure 9.1: Diamex PROG-S2

The programmer is connected to the PC via USB. The device manager can be used to quickly find out which COM port is used for communication.



RYZEN
Anschlüsse (COM & LPT)
Kommunikationsanschluss (COM1)
Serielles USB-Gerät (COM5)
Audio, Video und Gamecontroller
Audioeingänge und -ausgänge
Bildverarbeitungsgeräte
Computer
Drucker

Figure 9.2: Diamex in the Device manager (here: COM5)

DIP switches 1 and 2 must be on ON (5V and power supply on), DIP switches 3 and 4 must be OFF. In this case, the Chip Tester is supplied with voltage by the programmer and must no longer be supplied with voltage via USB or barrel connector.

This configuration can be used for programming, as described in section 4.

The connection is made with the 6-pin. connection cable (marked line up):

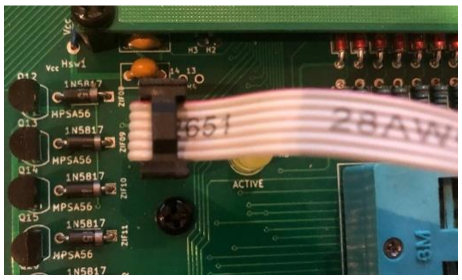
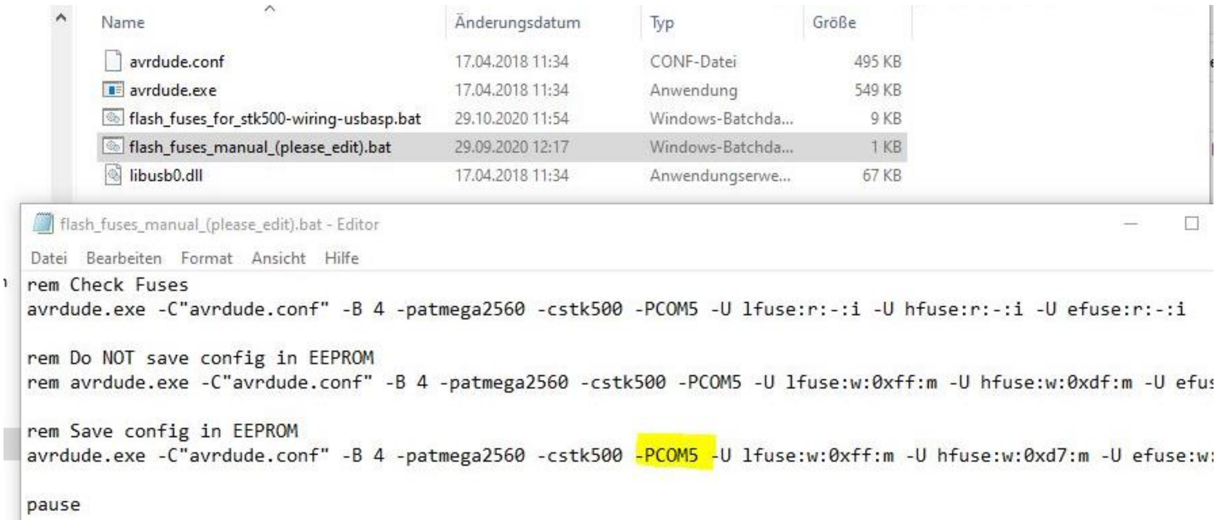


Figure 9.3: Connection of the Diamex

The fuses can then be set. There are two ways to do this:

1) Manually, via the command line (Batch: “flash_fuses_manual_(please_edit).bat”):



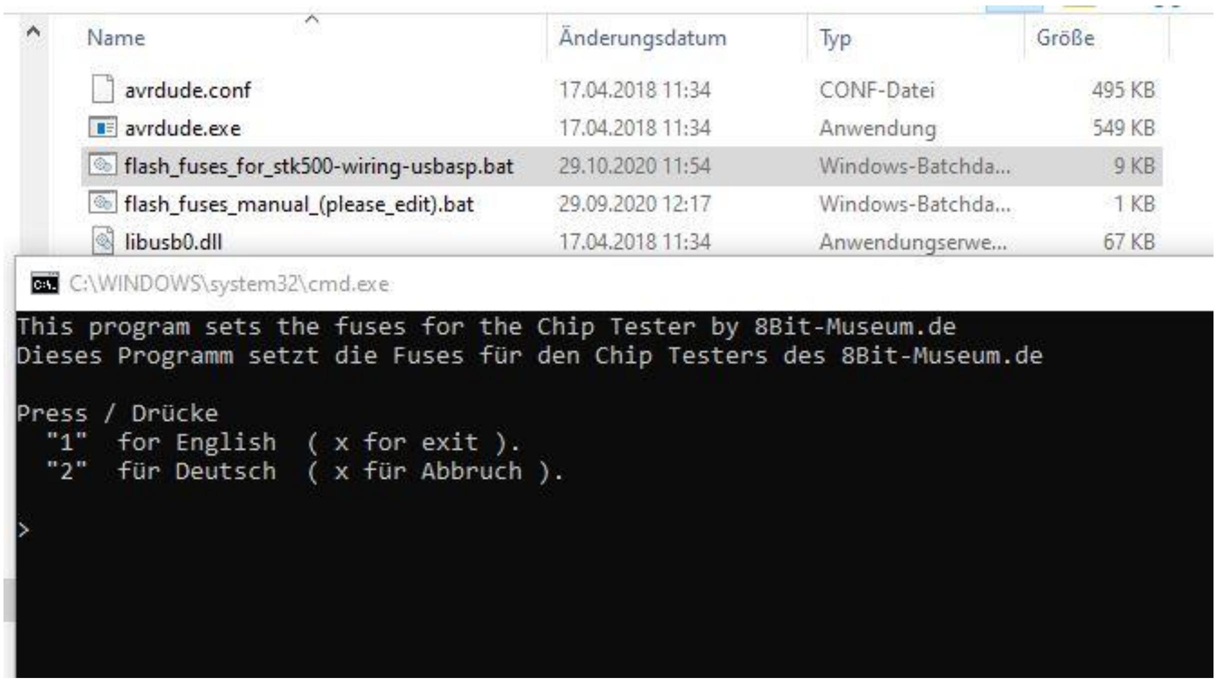
Name	Änderungsdatum	Typ	Größe
avrdude.conf	17.04.2018 11:34	CONF-Datei	495 KB
avrdude.exe	17.04.2018 11:34	Anwendung	549 KB
flash_fuses_for_stk500-wiring-usbasp.bat	29.10.2020 11:54	Windows-Batchda...	9 KB
flash_fuses_manual_(please_edit).bat	29.09.2020 12:17	Windows-Batchda...	1 KB
libusb0.dll	17.04.2018 11:34	Anwendungserwe...	67 KB

```
rem Check Fuses
avrdude.exe -C"avrdude.conf" -B 4 -patmega2560 -cstk500 -PCOM5 -U lfuse:r:-:i -U hfuse:r:-:i -U efuse:r:-:i

rem Do NOT save config in EEPROM
rem avrdude.exe -C"avrdude.conf" -B 4 -patmega2560 -cstk500 -PCOM5 -U lfuse:w:0xff:m -U hfuse:w:0xdf:m -U efus

rem Save config in EEPROM
avrdude.exe -C"avrdude.conf" -B 4 -patmega2560 -cstk500 -PCOM5 -U lfuse:w:0xff:m -U hfuse:w:0xd7:m -U efuse:w:

pause
```

Figure 9.4: Prepared batch file

In this case the COM port (marked yellow) must be adapted. Then the file can be started.

2) Interactive, through several queries (Batch: „flash_fuses_for_stk500-wiring-usbasp.bat“):



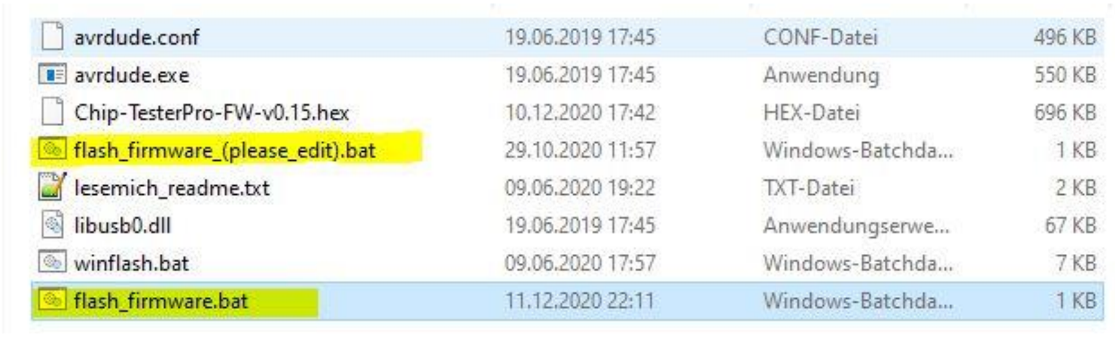
Name	Änderungsdatum	Typ	Größe
avrdude.conf	17.04.2018 11:34	CONF-Datei	495 KB
avrdude.exe	17.04.2018 11:34	Anwendung	549 KB
flash_fuses_for_stk500-wiring-usbasp.bat	29.10.2020 11:54	Windows-Batchda...	9 KB
flash_fuses_manual_(please_edit).bat	29.09.2020 12:17	Windows-Batchda...	1 KB
libusb0.dll	17.04.2018 11:34	Anwendungserwe...	67 KB

```
This program sets the fuses for the Chip Tester by 8Bit-Museum.de
Dieses Programm setzt die Fuses für den Chip Testers des 8Bit-Museum.de

Press / Drücke
  "1"  for English  ( x for exit ).
  "2"  für Deutsch  ( x für Abbruch ).

>
```

Figure 9.5: Interactive batch file

In this case the required data will be requested one after the other. At the end, a prepared batch file can be created on request, which makes flashing the firmware easier.

After the fuses have been set, the firmware can be flashed. To do this, simply copy the prepared batch file "flash_firmware.bat" into the firmware folder and start it, or adapt and start the existing batch file "flash_firmware_(please_edit).bat" accordingly.



avrdude.conf	19.06.2019 17:45	CONF-Datei	496 KB
avrdude.exe	19.06.2019 17:45	Anwendung	550 KB
Chip-TesterPro-FW-v0.15.hex	10.12.2020 17:42	HEX-Datei	696 KB
flash_firmware_(please_edit).bat	29.10.2020 11:57	Windows-Batchda...	1 KB
lesemich_readme.txt	09.06.2020 19:22	TXT-Datei	2 KB
libusb0.dll	19.06.2019 17:45	Anwendungserwe...	67 KB
winflash.bat	09.06.2020 17:57	Windows-Batchda...	7 KB
flash_firmware.bat	11.12.2020 22:11	Windows-Batchda...	1 KB

Figure 9.6: Flashing the firmware

The programming process should be completed after approx. 2 minutes.

9.2 Pololu USB AVR Programmer v2.1 / MacOS 10

The display must not be attached to the memory tester. The program "AVRFuses 1.5.2 (avrdude v6.3)"[1] is used to program the ATmega2560.

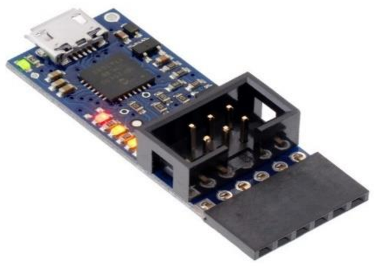
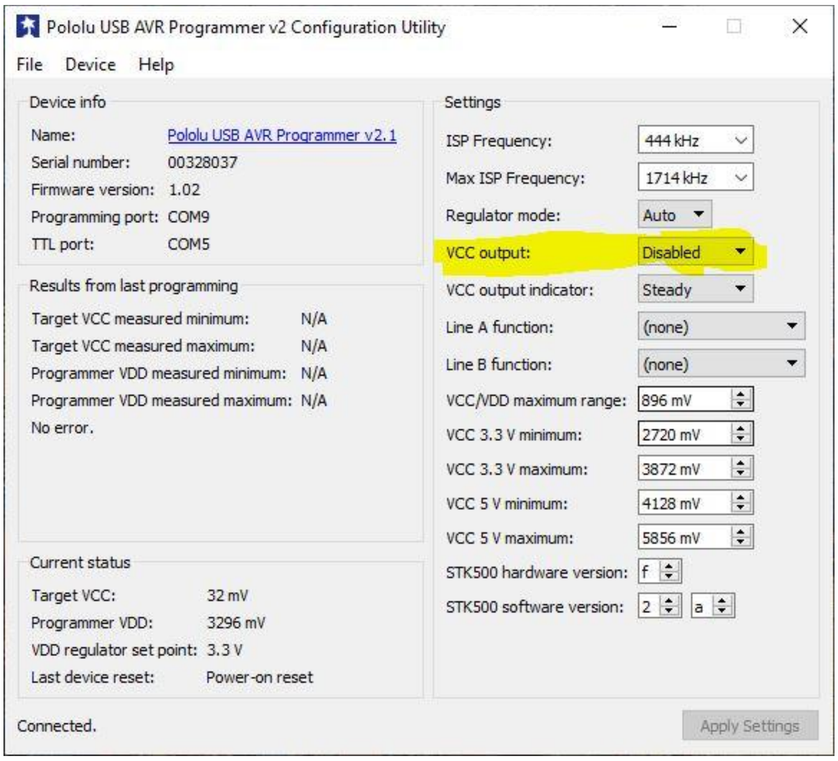


Figure 9.7: Pololu USB AVR Programmer v2.1

The Pololu Programmer is not suitable for supplying the board with voltage (it can work, but does not have to), therefore the board should be supplied with +5V via USB and the supply via the Programmer should be switched off.



Pololu USB AVR Programmer v2 Configuration Utility

File Device Help

Device info

Name:	Pololu USB AVR Programmer v2.1
Serial number:	00328037
Firmware version:	1.02
Programming port:	COM9
TTL port:	COM5

Results from last programming

Target VCC measured minimum:	N/A
Target VCC measured maximum:	N/A
Programmer VDD measured minimum:	N/A
Programmer VDD measured maximum:	N/A

No error.

Current status

Target VCC:	32 mV
Programmer VDD:	3296 mV
VDD regulator set point:	3.3 V
Last device reset:	Power-on reset

Settings

ISP Frequency:	444 kHz
Max ISP Frequency:	1714 kHz
Regulator mode:	Auto
VCC output:	Disabled
VCC output indicator:	Steady
Line A function:	(none)
Line B function:	(none)
VCC/VDD maximum range:	896 mV
VCC 3.3 V minimum:	2720 mV
VCC 3.3 V maximum:	3872 mV
VCC 5 V minimum:	4128 mV
VCC 5 V maximum:	5856 mV
STK500 hardware version:	f
STK500 software version:	2 a

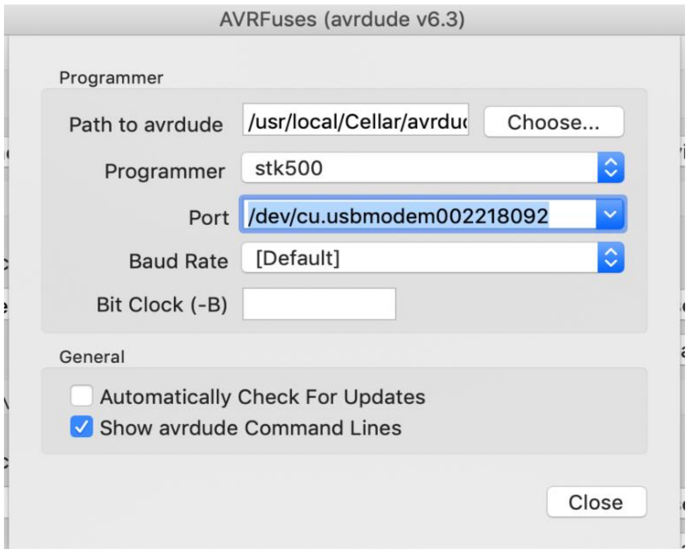
Connected.

Figure 9.8: Vcc output

[1] https://vonnieda.org/software/avrfuses

9.2.1 Settings AVRFuses

Menu: → AVRFuses → Preferences


AVRFuses (avrdude v6.3)

Programmer

Path to avrdude: /usr/local/Cellar/avrdu

Programmer: stk500

Port: /dev/cu.usbmodem002218092

Baud Rate: [Default]

Bit Clock (-B):

General

☐ Automatically Check For Updates

☑ Show avrdude Command Lines

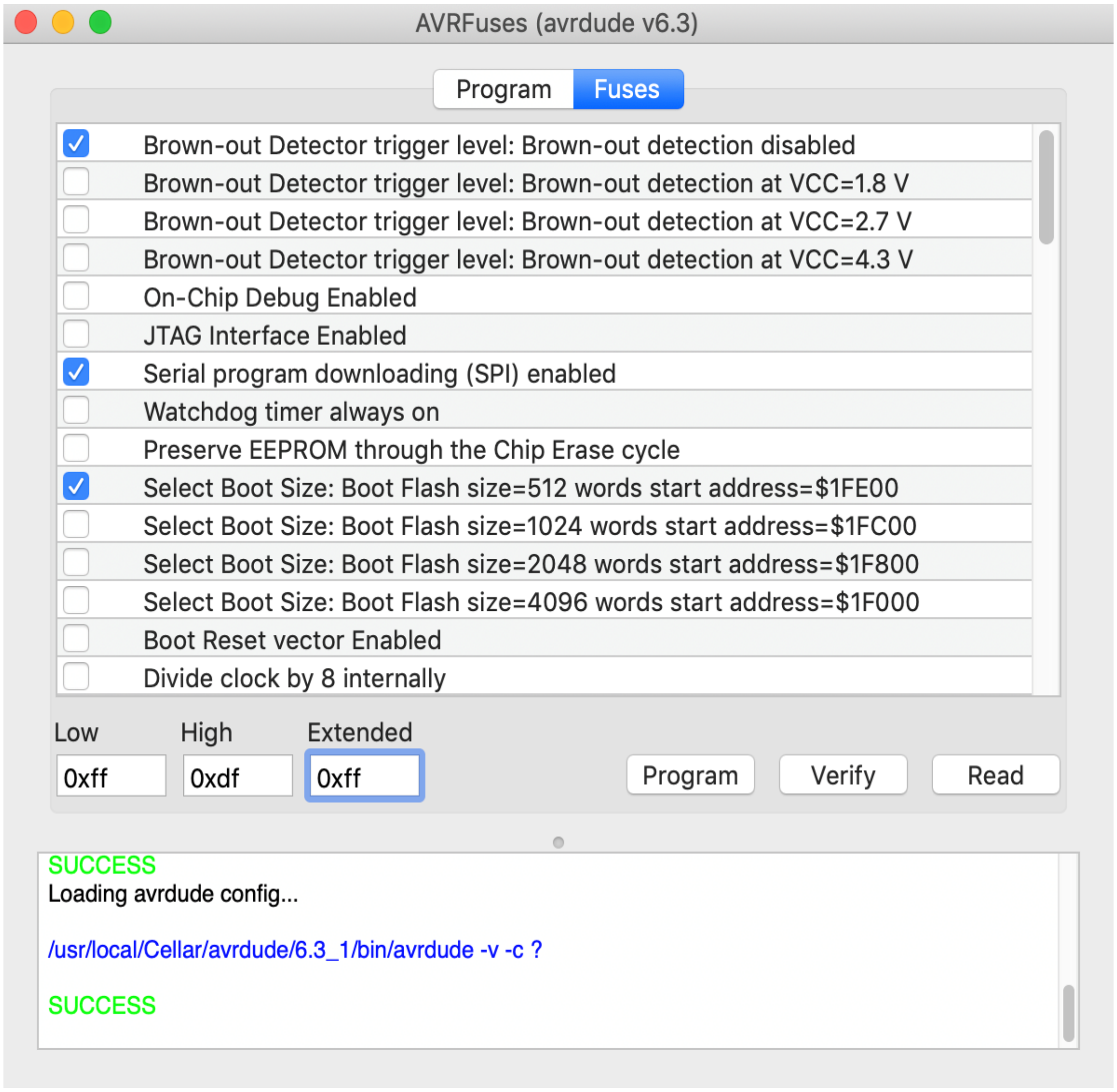
Figure 9.9: AVRFuses

Programmer:	STK500 oder STK500v2
Port:	dev/cu.usbmodemXXXXXXX {select the first „usbmodem“}
Baud Rate:	default
Bit Clock:	{empty, no entry}

9.2.2 Set Fuses

The fuses must be set as follows: Low: 0xf7, High: 0xd7, Extended: 0xff

When entering the values, make sure that the values are also adopted. The ticks change accordingly.



✓	Brown-out Detector trigger level: Brown-out detection disabled
	Brown-out Detector trigger level: Brown-out detection at VCC=1.8 V
	Brown-out Detector trigger level: Brown-out detection at VCC=2.7 V
	Brown-out Detector trigger level: Brown-out detection at VCC=4.3 V
	On-Chip Debug Enabled
	JTAG Interface Enabled
✓	Serial program downloading (SPI) enabled
	Watchdog timer always on
	Preserve EEPROM through the Chip Erase cycle
✓	Select Boot Size: Boot Flash size=512 words start address=$1FE00
	Select Boot Size: Boot Flash size=1024 words start address=$1FC00
	Select Boot Size: Boot Flash size=2048 words start address=$1F800
	Select Boot Size: Boot Flash size=4096 words start address=$1F000
	Boot Reset vector Enabled
	Divide clock by 8 internally

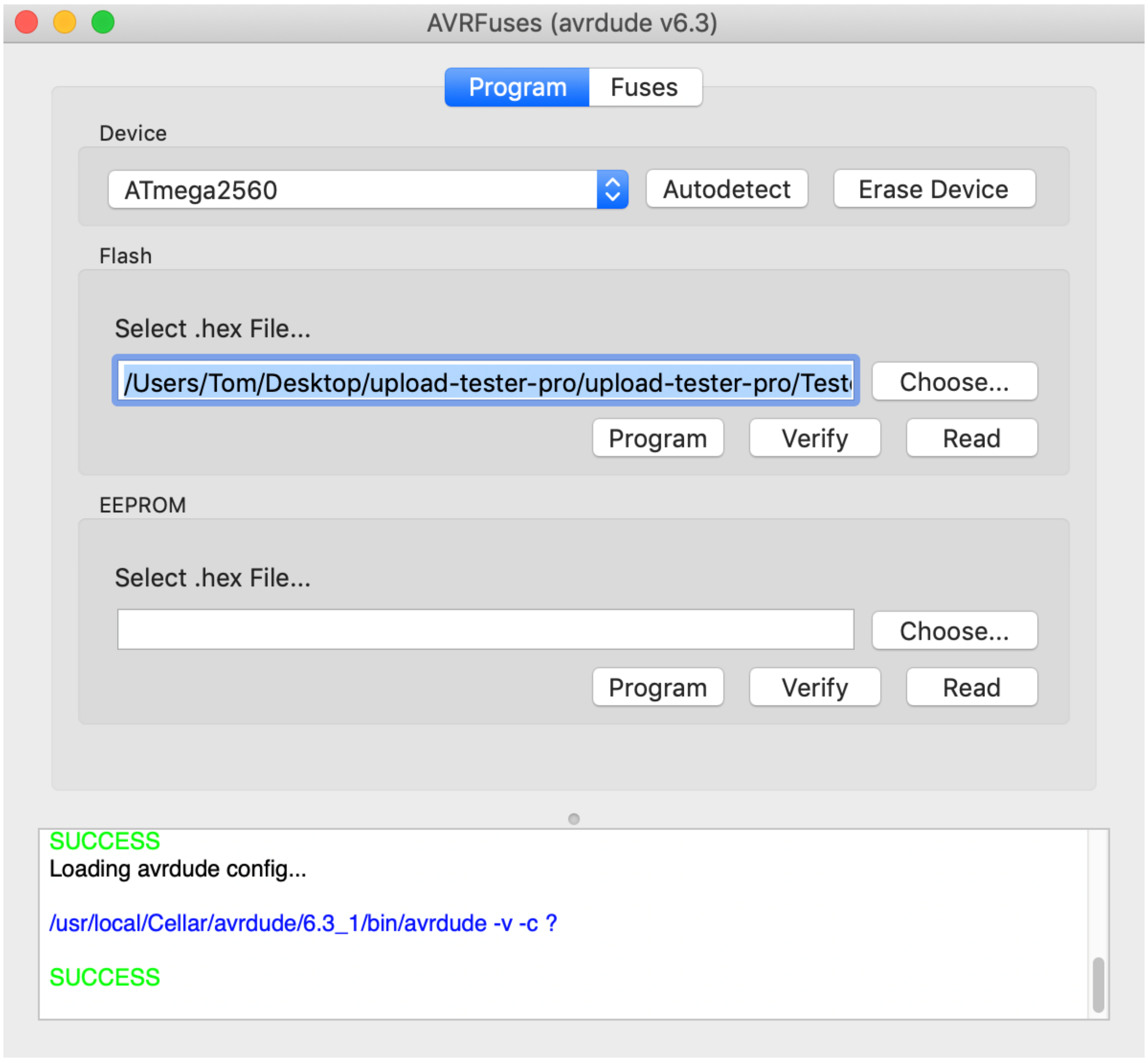
Low	High	Extended
0xff	0xdf	0xff

```
SUCCESS
Loading avrdude config...

/usr/local/Cellar/avrdude/6.3_1/bin/avrdude -v -c ?

SUCCESS
```

Figure 9.10: Setting the fuses with AVRFuses

Write the values into the ATmega2560 with "Program".

9.2.3 Flash Firmware



Device: ATmega2560

Flash, Select .hex File: /Users/Tom/Desktop/upload-tester-pro/upload-tester-pro/Test

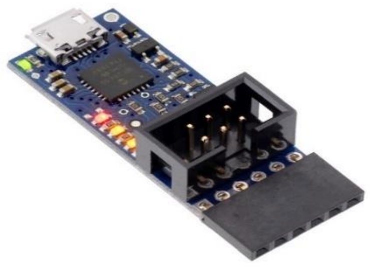
```
SUCCESS
Loading avrdude config...

/usr/local/Cellar/avrdude/6.3_1/bin/avrdude -v -c ?

SUCCESS
```

Figure 9.11: Programming with AVRFuses

• Device:	Select „ATmega2560“
• Flash, Select .hex File:	Select the firmware file
• EEPROM:	{empty, no entry}

Write the values into the ATmega2560 with "Program".

9.3 Pololu USB AVR Programmer v2.1 / Microsoft Windows

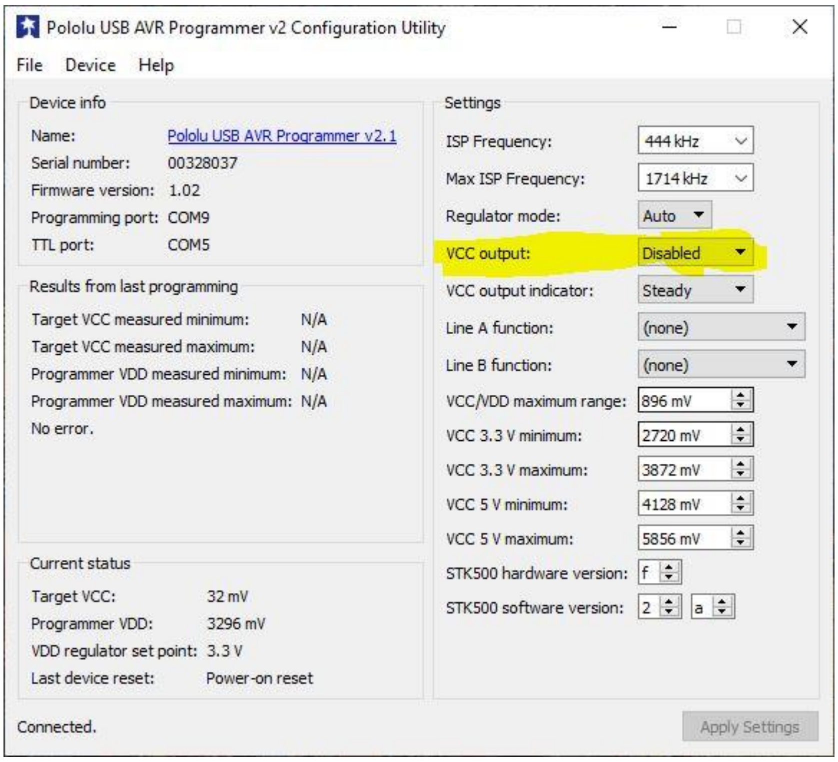


Before you program the Retro Chip Tester Professional, remove the display from the memory tester as follows:

1. Remove the 3 Phillips head screws.
2. The display is connected via a 16 pin connector at the top of the display board to the main board. Using gentle force, lift the display straight up from the main board. Some rocking may be employed to remove the display.
3. Set the removed display and screws aside.
4. Connect the Pololu USB AVR Programmer v2.1 to the RCT 6 pin connector (labeled ISP at the connector bottom) next to the ACTIVE light as follows:
   a. Position the 6 pin cable with the red wire at the top (towards where the display was on the board)
   b. Carefully push the 6 pin connector block on to the pins.
   Note: The connector for the Pololu USB AVR Programmer is indexed with a notch on the board connector, and a tab on the cable connector
   c. Plug the indexed 6 pin cable into the Pololu USB AVR Programmer.

Note: The Pololu Programmer is not recommended to supply the RCT board with voltage. The RCT board should be supplied with +5V via USB or an external power supply.

Figure 9.12: Pololu USB AVR Programmer v2.1



Device info

Name:	Pololu USB AVR Programmer v2.1
Serial number:	00328037
Firmware version:	1.02
Programming port:	COM9
TTL port:	COM5

Results from last programming

Target VCC measured minimum:	N/A
Target VCC measured maximum:	N/A
Programmer VDD measured minimum:	N/A
Programmer VDD measured maximum:	N/A

No error.

Current status

Target VCC:	32 mV
Programmer VDD:	3296 mV
VDD regulator set point:	3.3 V
Last device reset:	Power-on reset

Settings

ISP Frequency:	444 kHz
Max ISP Frequency:	1714 kHz
Regulator mode:	Auto
VCC output:	Disabled
VCC output indicator:	Steady
Line A function:	(none)
Line B function:	(none)
VCC/VDD maximum range:	896 mV
VCC 3.3 V minimum:	2720 mV
VCC 3.3 V maximum:	3872 mV
VCC 5 V minimum:	4128 mV
VCC 5 V maximum:	5856 mV
STK500 hardware version:	f
STK500 software version:	2 a

Connected.

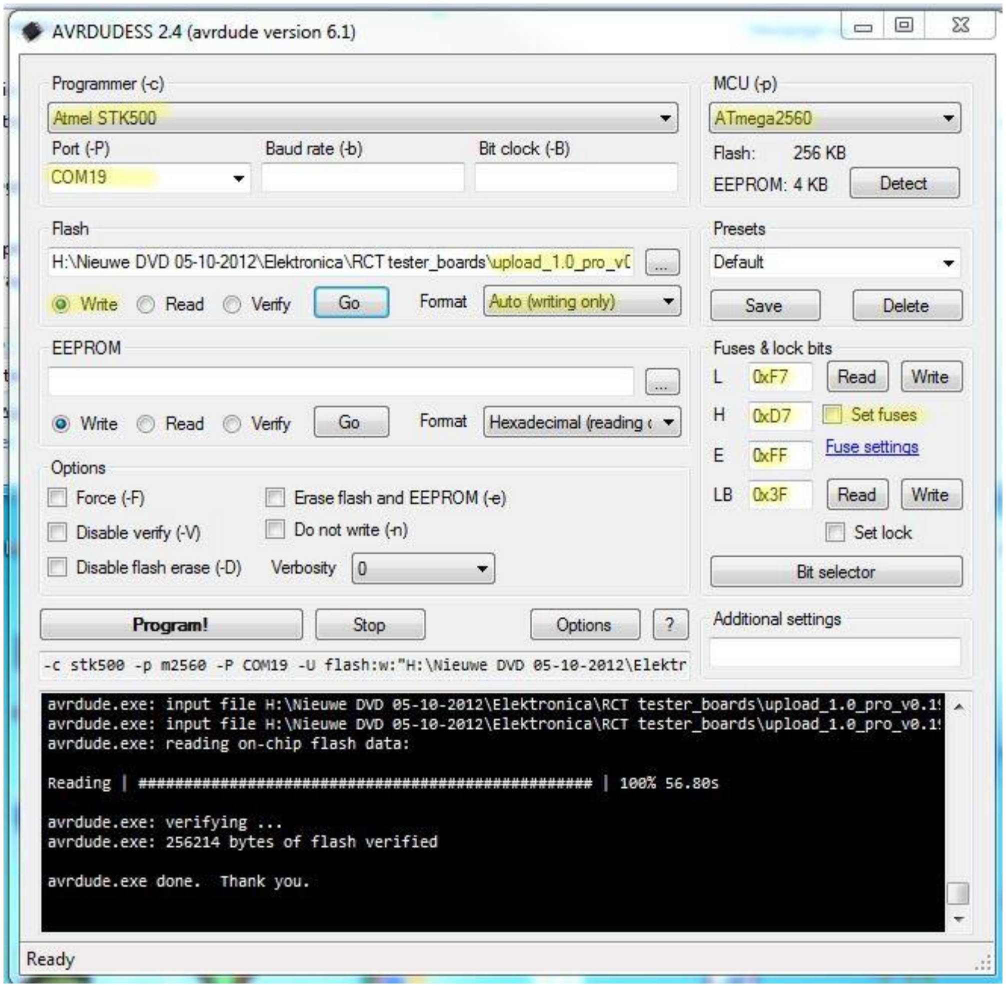
Figure 9.13: Vcc output

Continue with sec. 9.3.1 or sec. 0…

9.3.1 Programming using AVRDUDESS

The operation of AVRDUDESS[2] is relatively easy. Open AVRDUDESS from the Windows Menu.

1. Select the programmer (Atmel STK500 compatible) from the pull-down list under "Programmer (-c)"
2. Select the port used (Pololu creates two virtual ports, usually the lower port should be correct) from the pull-down list under "Ports (-P)"
3. Select the MCU (ATmega2560) from the pull-down list on the upper right portion of the GUI under "MCU (-p)"
4. Select the firmware to be programmed under "Flash" (HEX file) by clicking the button with three ellipses to the right of the "FLASH" header. This will be in your Downloads folder.
5. Select the "Write" checkbox.
6. Using the pull-down menu, select Format "Auto"
7. At the right side of the GUI, enter the Fuses and Lock Bits "L", "H", "E", "LB" as shown in the image below.
8. Check "Set fuses"
9. Press the "Write" button (above "Set Fuses").
10. When the fuses are set, write the firmware by pressing the "Go" button (under "Flash").
11. When the display shows Writing and Verification and AVR DONE, your programming is complete.


AVRDUDESS 2.4 (avrdude version 6.1)

Programmer (-c): Atmel STK500
Port (-P): COM19
Baud rate (-b):
Bit clock (-B):
MCU (-p): ATmega2560
Flash: 256 KB
EEPROM: 4 KB

Flash: H:\Nieuwe DVD 05-10-2012\Elektronica\RCT tester_boards\upload_1.0_pro_v(
[✓ Write] Read Verify
Format: Auto (writing only)

EEPROM:
[✓ Write] Read Verify
Format: Hexadecimal (reading (

Presets: Default

Fuses & lock bits
L: 0xF7
H: 0xD7
E: 0xFF
LB: 0x3F
Set fuses
Set lock

Options
Force (-F)
Disable verify (-V)
Disable flash erase (-D)
Erase flash and EEPROM (-e)
Do not write (-n)
Verbosity: 0

```
-c stk500 -p m2560 -P COM19 -U flash:w:"H:\Nieuwe DVD 05-10-2012\Elektr
```

```
avrdude.exe: input file H:\Nieuwe DVD 05-10-2012\Elektronica\RCT tester_boards\upload_1.0_pro_v0.1!
avrdude.exe: input file H:\Nieuwe DVD 05-10-2012\Elektronica\RCT tester_boards\upload_1.0_pro_v0.1!
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 100% 56.80s

avrdude.exe: verifying ...
avrdude.exe: 256214 bytes of flash verified

avrdude.exe done.  Thank you.
```

Ready

Figure 9.14: AVRDUDESS

[2] https://github.com/zkemble/AVRDUDESS

9.3.2 Programming using the command line

The procedure corresponds to the process described in sec. 4.2 (if necessary set the fuses beforehand).

```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cstk500 -PCOMx
-Uflash:w:Chip-TesterPro-FW-v0.XX.hex:i
```

For `-PCOMx` specify the appropriate port, e.g. `-PCOM9` and adjust the version number of the firmware.

If problems occur, then also set the speed:

```
avrdude.exe -C"avrdude.conf" -v -patmega2560 -cstk500 -PCOMx -b115200
-Uflash:w:Chip-TesterPro-FW-v0.XX.hex:i
```

The programming process can be significantly accelerated by specifying `-B1` (factor 4) or `-B0.5` (factor 8).

9.4 Arduino Uno as ISP programmer

The Arduino can be used as a universal programming adapter to flash individual microcontrollers such as AVRs and PICs. Even if the programming usually works without any problems, this solution is a bit "shaky".

First the code for simulating an ISP programmer has to be loaded into the Arduino Uno:

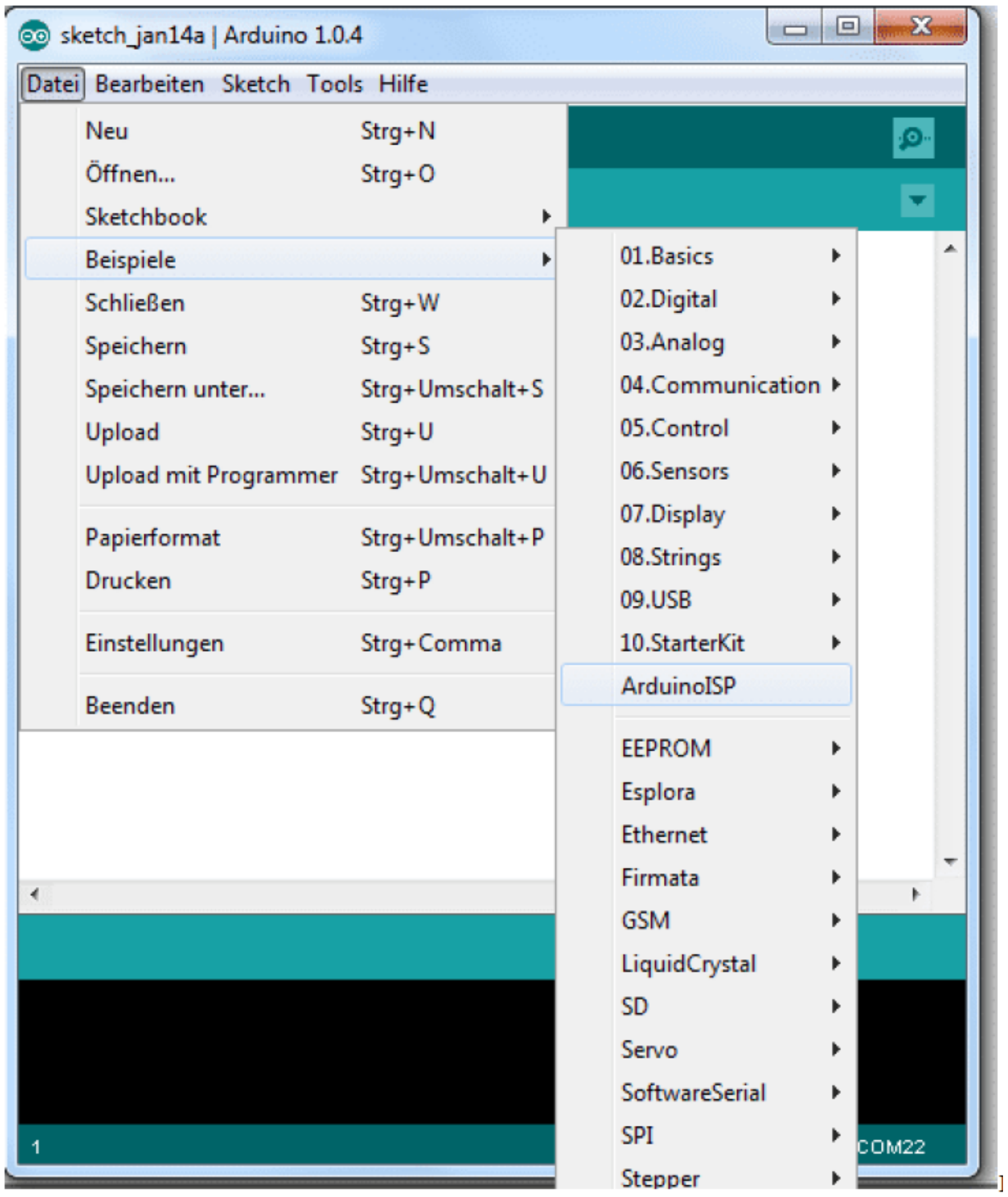
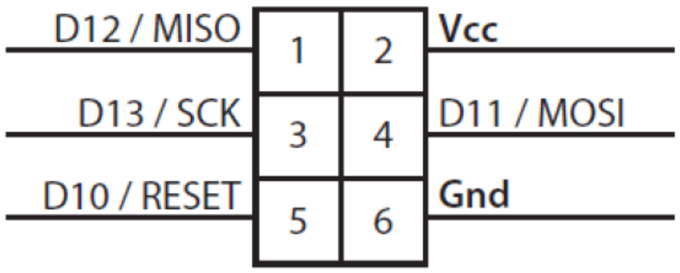


Figure 9.15: Arduino IDE

For the control of MISO, MOSI and SCK the Arduino Uno uses pins 11, 12 and 13, the RST input of the microcontroller controls pin 10. The assignment of a six-pin ISP socket connector for programming an AVR controller in a finished circuit is seen in the picture.



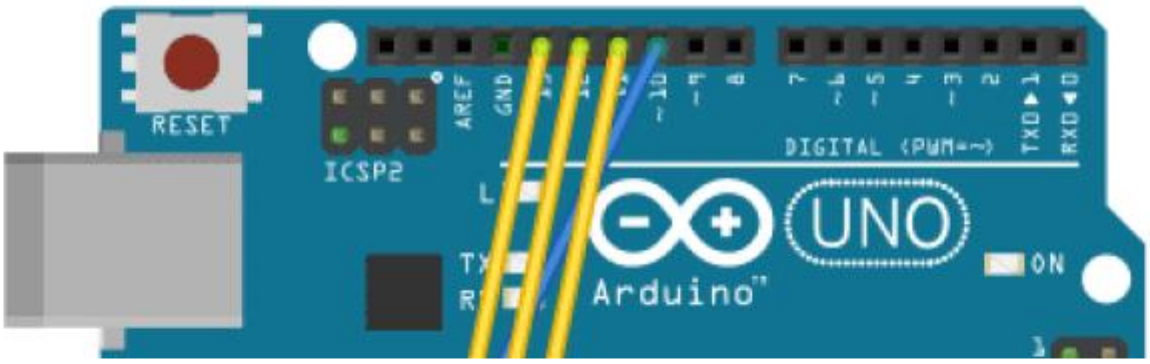
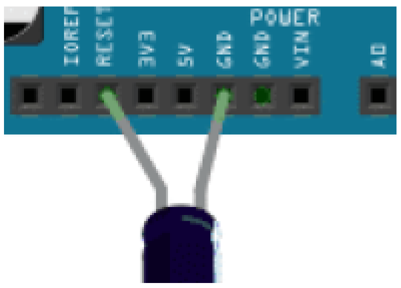


Figure 9.16: Using an Arduino Uno as programmer

For programming it must be ensured that "Programmer> Arduino ISP" is selected.

With the Arduino Uno you have to put a capacitor (electrolytic capacitor with 10µF) between GND and Reset for the next steps. It intercepts the reset pulse that the USB-serial converter on the board of the Arduino Uno sends when starting serial communication. The reset pulse causes the microcontroller on the Arduino board to start its bootloader for reprogramming instead of starting the Arduino sketch.



There are probably two different sketches, so that the Arduino programmer is addressed either as `avrisp` or as `stk500v1`. In both cases the programming is done with AVRDUDE:

1) As programmer `avrisp` is specified:

```
avrdude.exe -C"avrdude.conf" -v -p atmega2560 -c avrisp -P COM3 -b 19200 ...
```

2) As programmer stk500v1 is specified:

```
avrdude.exe -C"avrdude.conf" -v -p atmega2560 -c stk500v1 -P COM3 -b 19200 ...
```

Otherwise, the ATmega2560 is programmed as described in section 4.

9.5 AVRISP-MKII compatible programmer

If you want to use a programmer compatible with the AVRISP-MKII, you must first ensure that AVRdude can communicate with it. If the standard Atmel drivers are installed, communication only works with Atmel Studio, but not with AVRdude.

Some Chinese AVRISP MK II use a CH340 chip for communication with the PC. The CH340 driver must be installed for it to work and COM9 must be used otherwise the AVRISP MKII will not be seen by the PC.

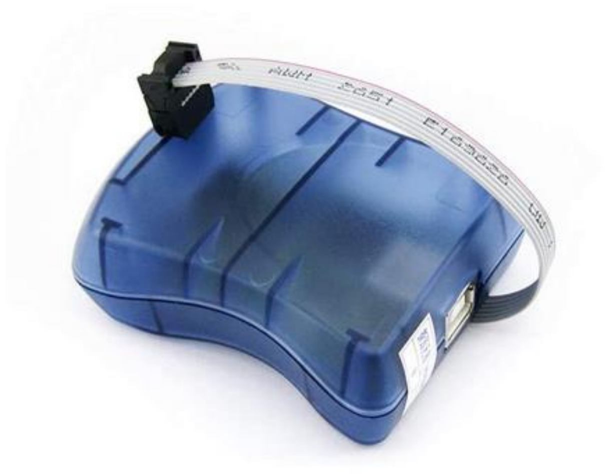
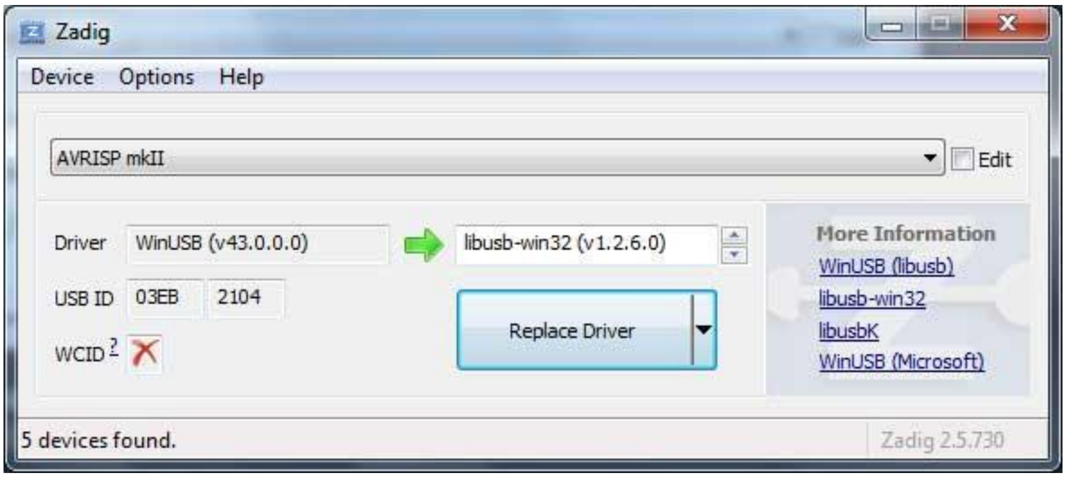


Figure 9.17: AVRISP-MKII compatible programmer

The USB driver used must first be changed. This is done reliably by the free tool ZADIG, which can be downloaded here: https://zadig.akeo.ie/

After the installation, select the option that all devices are listed. Then select “AVRISP mkII” in the list and change the driver to “libusb-win32”. Finally, select "Replace Driver".



Zadig

Device Options Help

AVRISP mkII

Edit

Driver WinUSB (v43.0.0.0) → libusb-win32 (v1.2.6.0)

USB ID 03EB 2104

WCID ?

Replace Driver

More Information
WinUSB (libusb)
libusb-win32
libusbK
WinUSB (Microsoft)

5 devices found.

Zadig 2.5.730

Figure 9.18: ZADIG

If the programmer is reconnected, the new driver becomes active. It is now addressed with AVRdude via the parameter "`-cavrispmkii -Pusb`".

9.6 Microchip MPLABX / PICkit4

The firmware can be updated using a PICkit4 from Microchip. Use microchip's MPLABX IDE software. Note the pickit4 won't supply voltage to the AVR microcontrollers, so use USB to power the RCT. In MPLAB import the hex file, and switch the communication mode of the pickit4 to ISP from JTAG. After this, programming works fine.

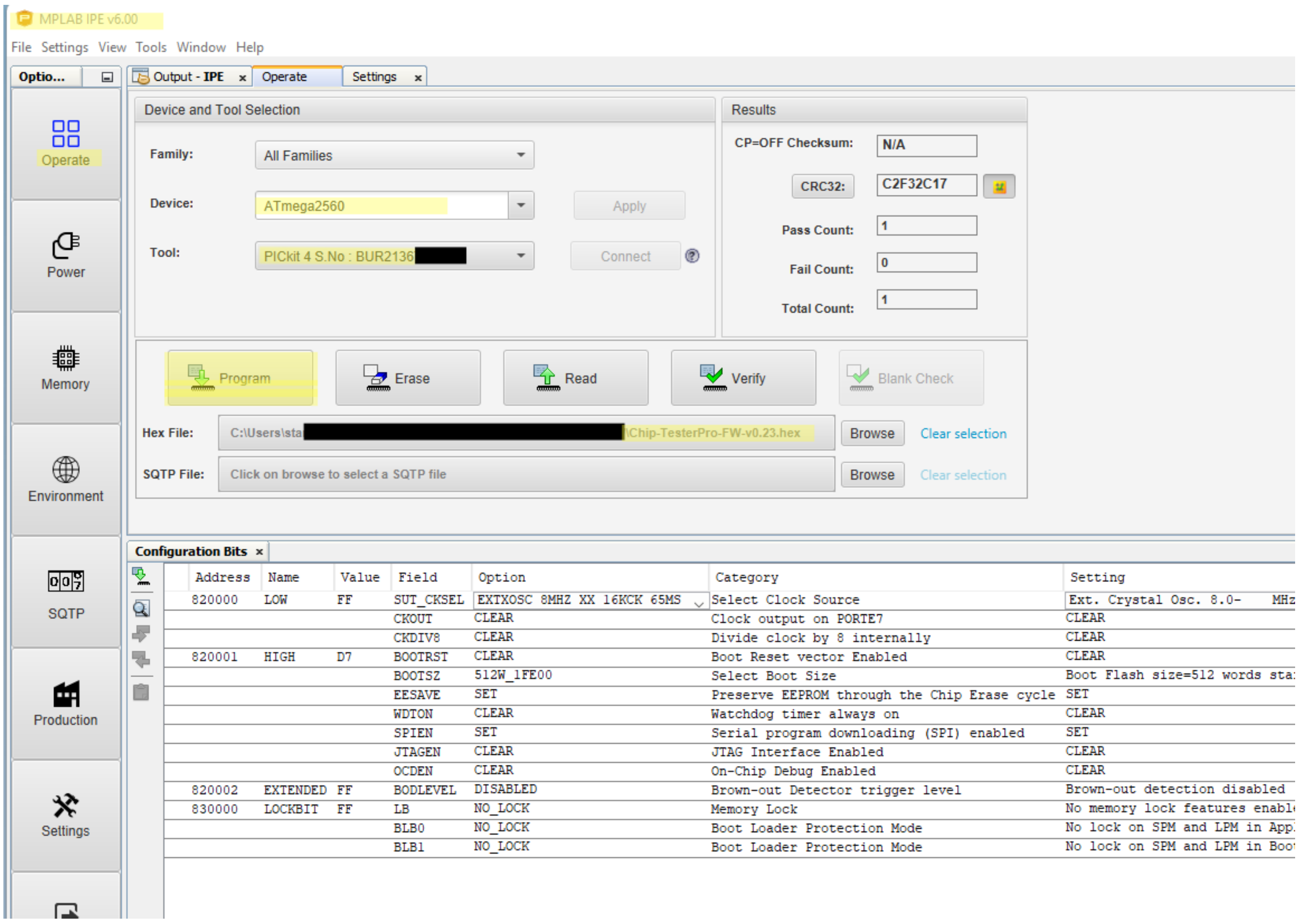
9.7 Microchip MPLABX with IPE Application / PICkit4

You can use the MPLABX companion IPE application (under Windows 11) for downloading precompiled HEX files to the RCT. The IPE application is simpler to use when only a HEX file download is needed.

Note the pickit4 won't supply voltage to the AVR microcontrollers, so use USB to power the RCT. When the ATmega2650 is chosen in the menu, the internal power option is automatically greyed out in the default “power” menu dialog.

Operate tab:

Choose ATmega2560 device, your programmer, the hex file, the configuration bits should be FF, D7, FF, FF (low, high, extended, lockbit).



Device and Tool Selection

Family: All Families

Device: ATmega2560

Tool: PICkit 4 S.No : BUR2136[redacted]

Results

CP=OFF Checksum: N/A

CRC32: C2F32C17

Pass Count: 1

Fail Count: 0

Total Count: 1

Hex File: C:\Users\sta[redacted]\Chip-TesterPro-FW-v0.23.hex

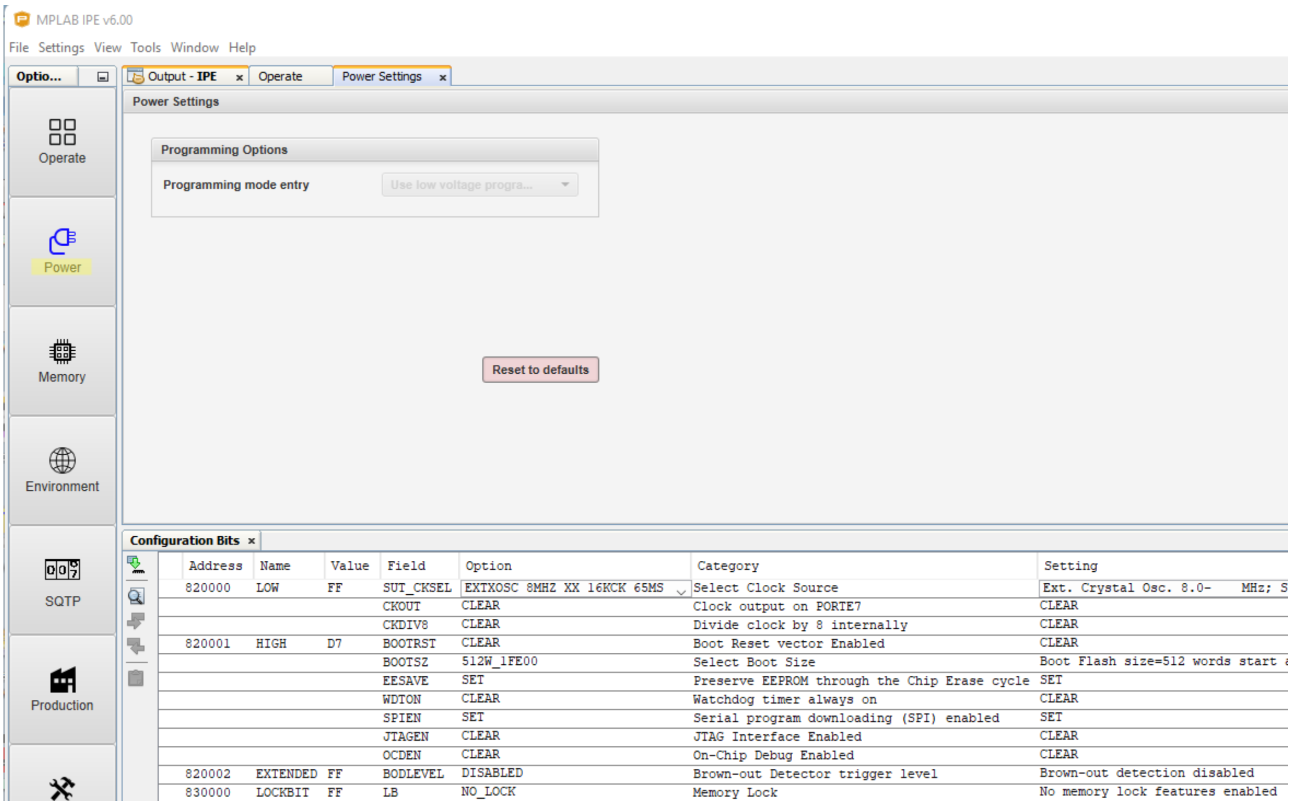
SQTP File: Click on browse to select a SQTP file

Configuration Bits

Address	Name	Value	Field	Option	Category	Setting
820000	LOW	FF	SUT_CKSEL	EXTXOSC 8MHZ XX 16KCK 65MS	Select Clock Source	Ext. Crystal Osc. 8.0- MHz
			CKOUT	CLEAR	Clock output on PORTE7	CLEAR
			CKDIV8	CLEAR	Divide clock by 8 internally	CLEAR
820001	HIGH	D7	BOOTRST	CLEAR	Boot Reset vector Enabled	CLEAR
			BOOTSZ	512W_1FE00	Select Boot Size	Boot Flash size=512 words sta
			EESAVE	SET	Preserve EEPROM through the Chip Erase cycle	SET
			WDTON	CLEAR	Watchdog timer always on	CLEAR
			SPIEN	SET	Serial program downloading (SPI) enabled	SET
			JTAGEN	CLEAR	JTAG Interface Enabled	CLEAR
			OCDEN	CLEAR	On-Chip Debug Enabled	CLEAR
820002	EXTENDED	FF	BODLEVEL	DISABLED	Brown-out Detector trigger level	Brown-out detection disabled
830000	LOCKBIT	FF	LB	NO_LOCK	Memory Lock	No memory lock features enabl
			BLB0	NO_LOCK	Boot Loader Protection Mode	No lock on SPM and LPM in App
			BLB1	NO_LOCK	Boot Loader Protection Mode	No lock on SPM and LPM in Boo

Power tab:

Power is greyed out.



Power Settings

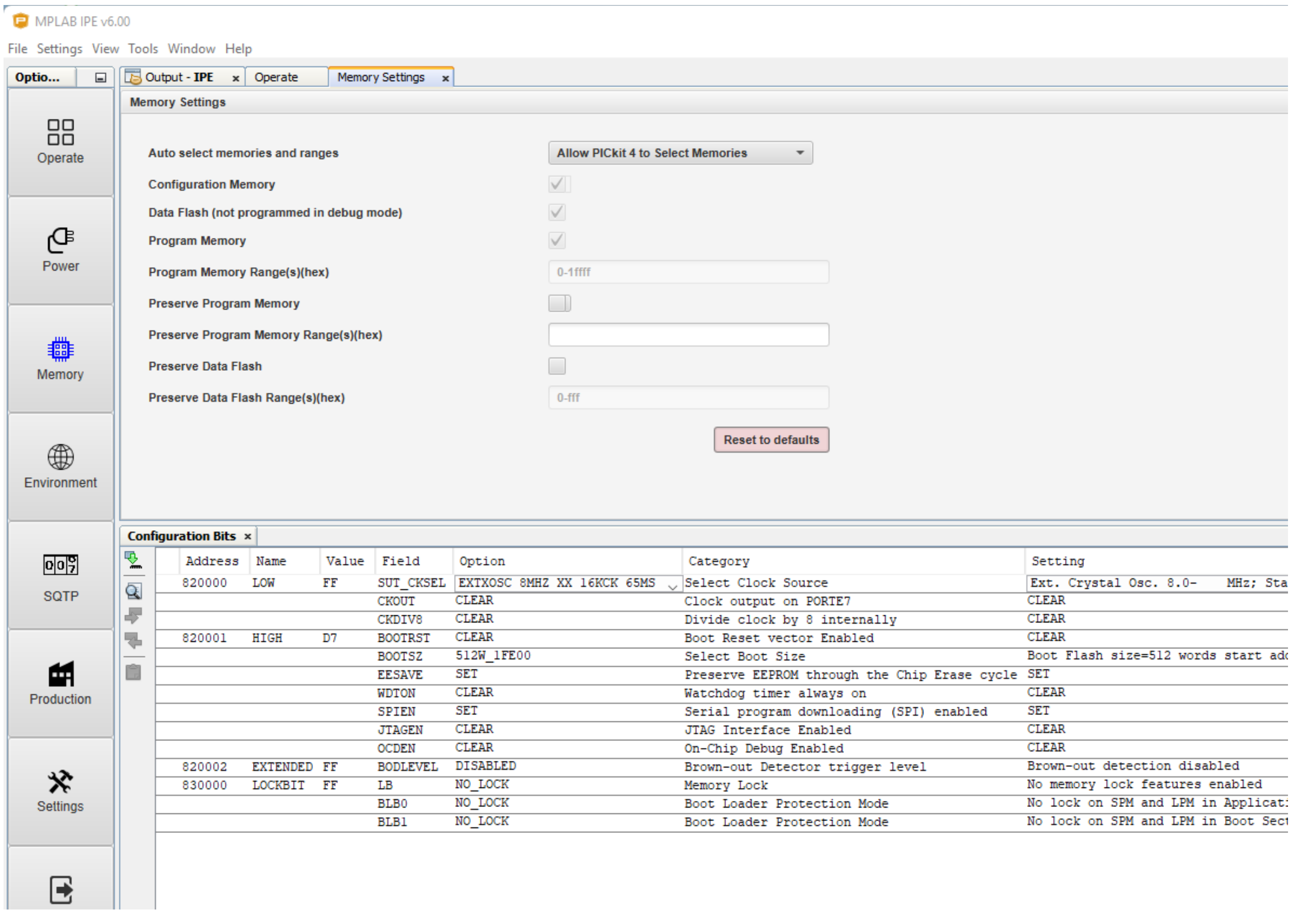
Programming Options

Programming mode entry: Use low voltage progra...

Address	Name	Value	Field	Option	Category	Setting
820000	LOW	FF	SUT_CKSEL	EXTXOSC 8MHZ XX 16KCK 65MS	Select Clock Source	Ext. Crystal Osc. 8.0- MHz; S
			CKOUT	CLEAR	Clock output on PORTE7	CLEAR
			CKDIV8	CLEAR	Divide clock by 8 internally	CLEAR
820001	HIGH	D7	BOOTRST	CLEAR	Boot Reset vector Enabled	CLEAR
			BOOTSZ	512W_1FE00	Select Boot Size	Boot Flash size=512 words start
			EESAVE	SET	Preserve EEPROM through the Chip Erase cycle	SET
			WDTON	CLEAR	Watchdog timer always on	CLEAR
			SPIEN	SET	Serial program downloading (SPI) enabled	SET
			JTAGEN	CLEAR	JTAG Interface Enabled	CLEAR
			OCDEN	CLEAR	On-Chip Debug Enabled	CLEAR
820002	EXTENDED	FF	BODLEVEL	DISABLED	Brown-out Detector trigger level	Brown-out detection disabled
830000	LOCKBIT	FF	LB	NO_LOCK	Memory Lock	No memory lock features enabled

Memory tab:

Select “Allow PICkit 4 to Select Memories”



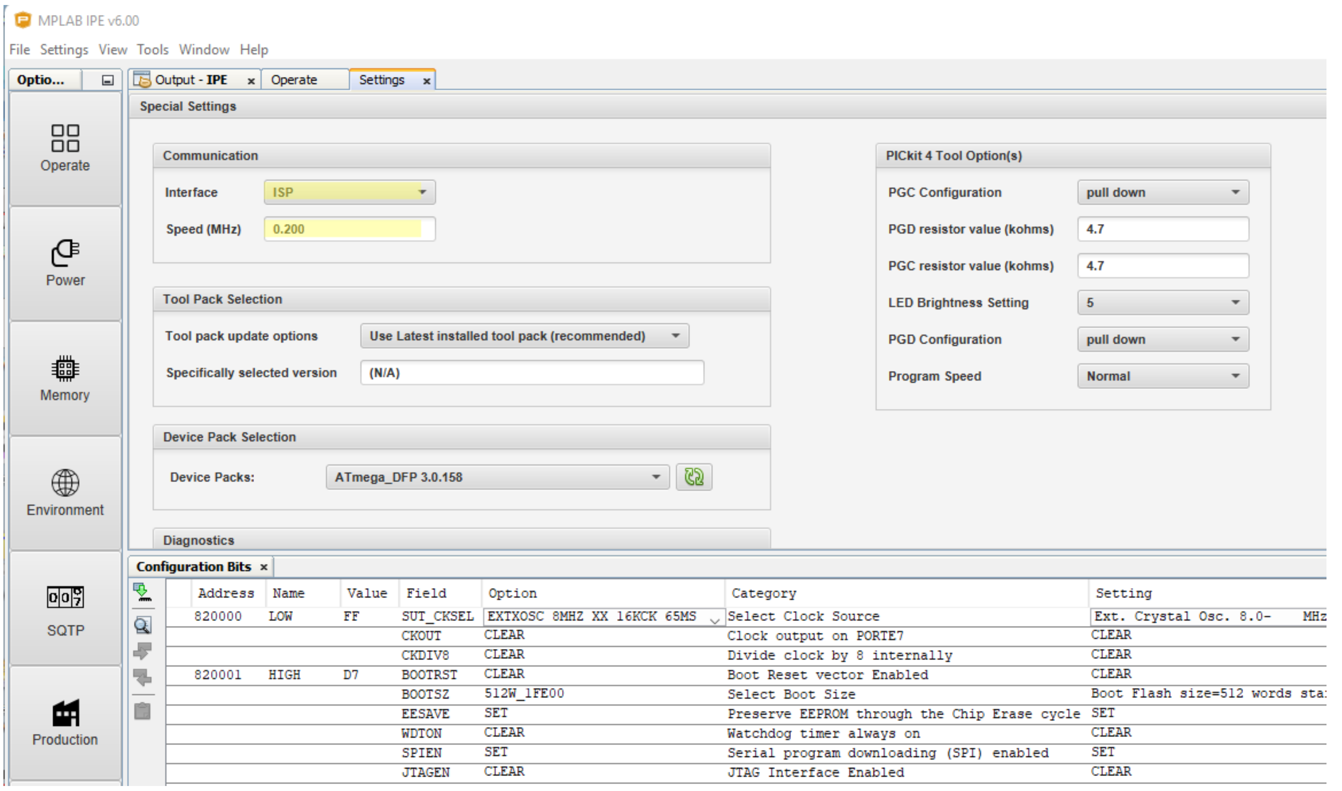
Memory Settings

Auto select memories and ranges	Allow PICkit 4 to Select Memories
Configuration Memory	✓
Data Flash (not programmed in debug mode)	✓
Program Memory	✓
Program Memory Range(s)(hex)	0-1ffff
Preserve Program Memory	☐
Preserve Program Memory Range(s)(hex)	
Preserve Data Flash	☐
Preserve Data Flash Range(s)(hex)	0-fff

Address	Name	Value	Field	Option	Category	Setting
820000	LOW	FF	SUT_CKSEL	EXTXOSC 8MHZ XX 16KCK 65MS	Select Clock Source	Ext. Crystal Osc. 8.0- MHz; Sta
			CKOUT	CLEAR	Clock output on PORTE7	CLEAR
			CKDIV8	CLEAR	Divide clock by 8 internally	CLEAR
820001	HIGH	D7	BOOTRST	CLEAR	Boot Reset vector Enabled	CLEAR
			BOOTSZ	512W_1FE00	Select Boot Size	Boot Flash size=512 words start ad
			EESAVE	SET	Preserve EEPROM through the Chip Erase cycle	SET
			WDTON	CLEAR	Watchdog timer always on	CLEAR
			SPIEN	SET	Serial program downloading (SPI) enabled	SET
			JTAGEN	CLEAR	JTAG Interface Enabled	CLEAR
			OCDEN	CLEAR	On-Chip Debug Enabled	CLEAR
820002	EXTENDED	FF	BODLEVEL	DISABLED	Brown-out Detector trigger level	Brown-out detection disabled
830000	LOCKBIT	FF	LB	NO_LOCK	Memory Lock	No memory lock features enabled
			BLB0	NO_LOCK	Boot Loader Protection Mode	No lock on SPM and LPM in Applicat
			BLB1	NO_LOCK	Boot Loader Protection Mode	No lock on SPM and LPM in Boot Sect

Settings tab:

Choose ISP interface and a speed of 0.200 MHz.



Special Settings

Communication

Interface	ISP
Speed (MHz)	0.200

Tool Pack Selection

Tool pack update options	Use Latest installed tool pack (recommended)
Specifically selected version	(N/A)

Device Pack Selection

Device Packs:	ATmega_DFP 3.0.158

Diagnostics

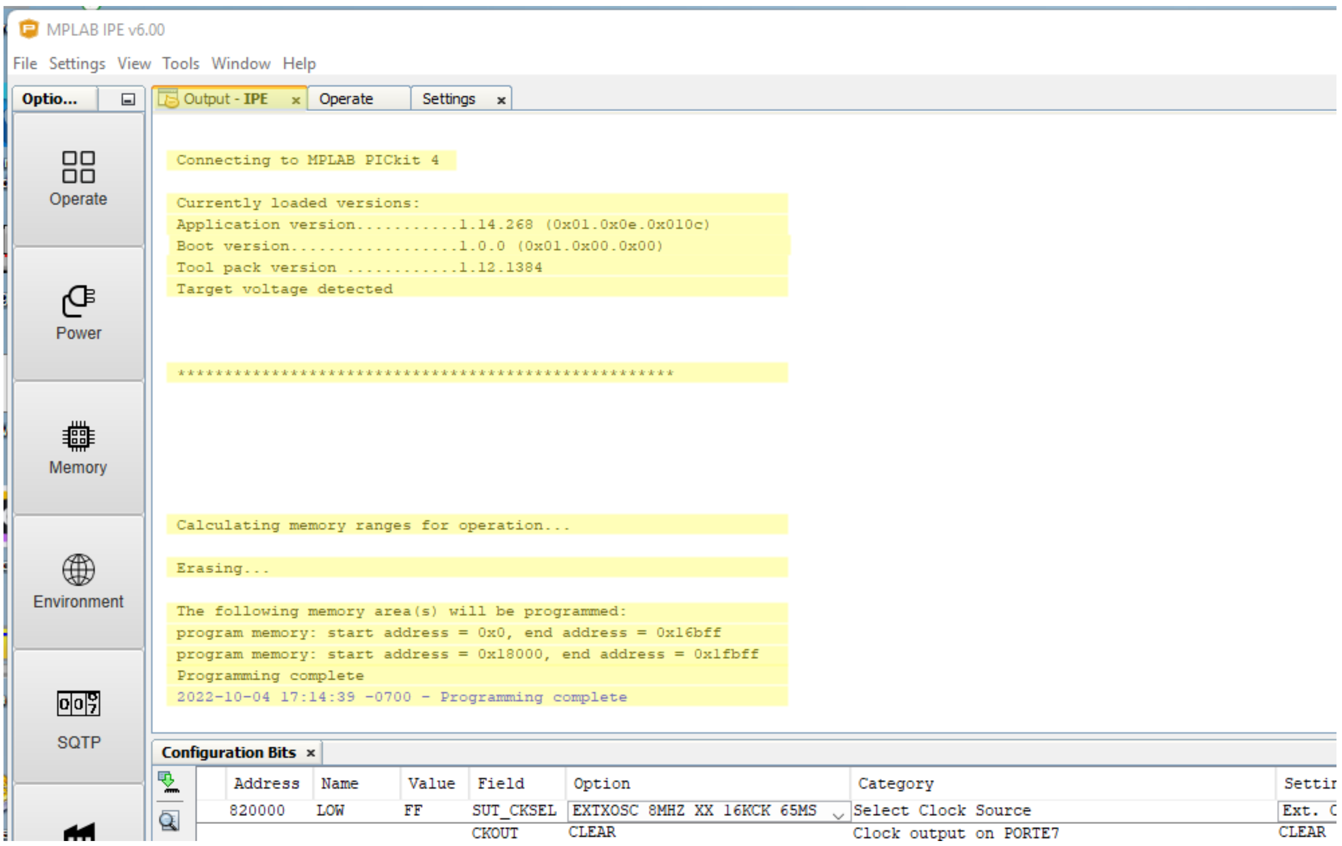
PICkit 4 Tool Option(s)

PGC Configuration	pull down
PGD resistor value (kohms)	4.7
PGC resistor value (kohms)	4.7
LED Brightness Setting	5
PGD Configuration	pull down
Program Speed	Normal

Configuration Bits

Address	Name	Value	Field	Option	Category	Setting
820000	LOW	FF	SUT_CKSEL	EXTXOSC 8MHZ XX 16KCK 65MS	Select Clock Source	Ext. Crystal Osc. 8.0- MHz
			CKOUT	CLEAR	Clock output on PORTE7	CLEAR
			CKDIV8	CLEAR	Divide clock by 8 internally	CLEAR
820001	HIGH	D7	BOOTRST	CLEAR	Boot Reset vector Enabled	CLEAR
			BOOTSZ	512W_1FE00	Select Boot Size	Boot Flash size=512 words sta
			EESAVE	SET	Preserve EEPROM through the Chip Erase cycle	SET
			WDTON	CLEAR	Watchdog timer always on	CLEAR
			SPIEN	SET	Serial program downloading (SPI) enabled	SET
			JTAGEN	CLEAR	JTAG Interface Enabled	CLEAR

Output IPE tab:



```
Connecting to MPLAB PICkit 4

Currently loaded versions:
Application version...........1.14.268 (0x01.0x0e.0x010c)
Boot version..................1.0.0 (0x01.0x00.0x00)
Tool pack version ............1.12.1384
Target voltage detected



*****************************************************



Calculating memory ranges for operation...

Erasing...

The following memory area(s) will be programmed:
program memory: start address = 0x0, end address = 0x16bff
program memory: start address = 0x18000, end address = 0x1fbff
Programming complete
2022-10-04 17:14:39 -0700 - Programming complete
```

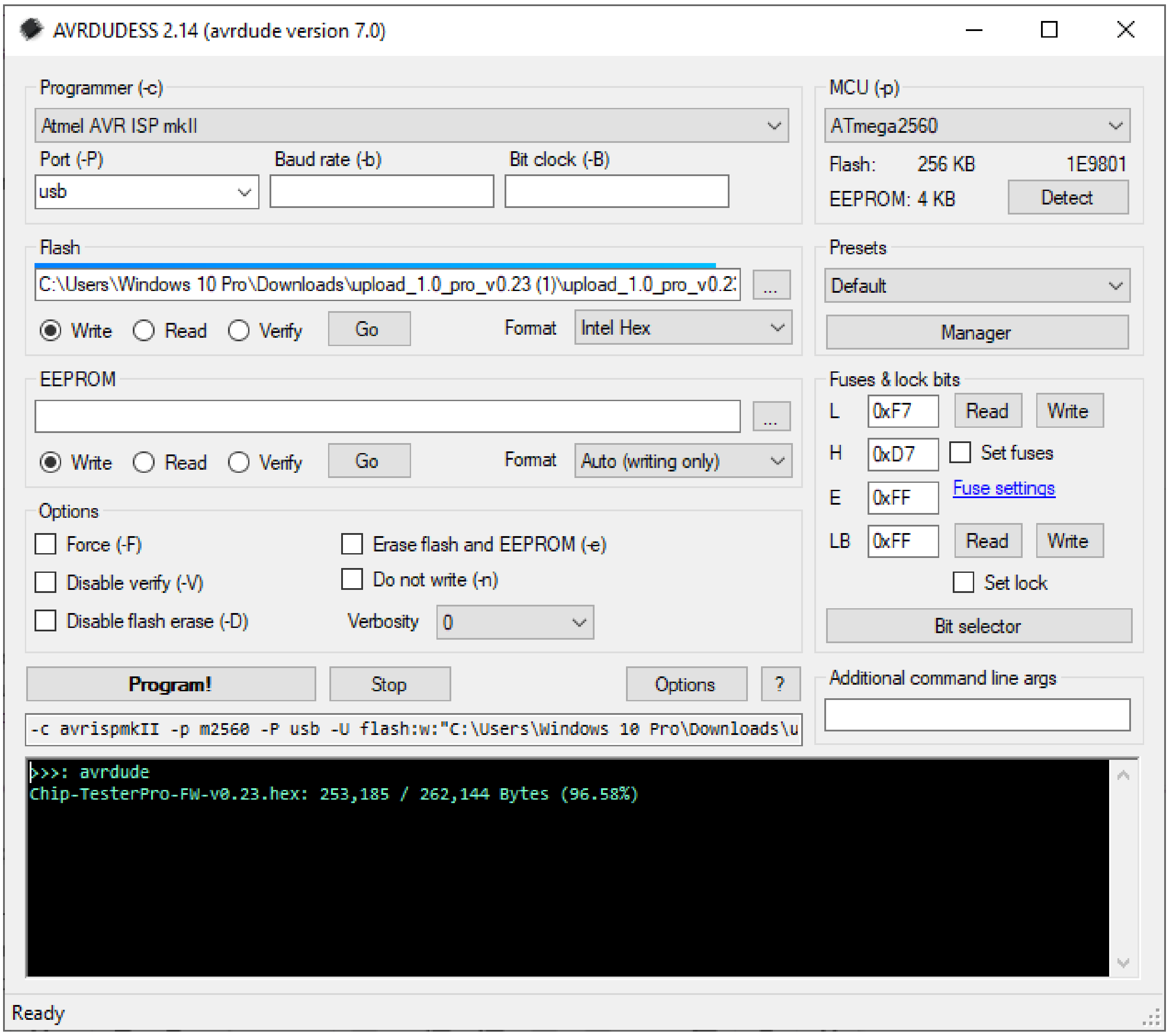
Configuration Bits

Address	Name	Value	Field	Option	Category	Settir
820000	LOW	FF	SUT_CKSEL	EXTXOSC 8MHZ XX 16KCK 65MS	Select Clock Source	Ext. C
			CKOUT	CLEAR	Clock output on PORTE7	CLEAR

9.8 OLIMEX AVR-ISP-MK2

The OLIMEX AVR-ISP-MK2 programmer is able to power the RCT board (it has two jumpers that can be configured, one to power on or off and the other to select 5v or 3.3v).

You can use AVRDUDESS 2.14 with AVRDUDE v7.0 (because it allows the -P to be configured USB instead of a COM port).



AVRDUDESS 2.14 (avrdude version 7.0)

Programmer (-c): Atmel AVR ISP mkII
Port (-P): usb
Baud rate (-b):
Bit clock (-B):

MCU (-p): ATmega2560
Flash: 256 KB 1E9801
EEPROM: 4 KB

Flash: C:\Users\Windows 10 Pro\Downloads\upload_1.0_pro_v0.23 (1)\upload_1.0_pro_v0.2
[✓ Write] Read Verify
Format: Intel Hex

Presets: Default

EEPROM:
[✓ Write] Read Verify
Format: Auto (writing only)

Fuses & lock bits
L: 0xF7
H: 0xD7
E: 0xFF
LB: 0xFF
Set fuses ☐
Set lock ☐

Options
☐ Force (-F)
☐ Erase flash and EEPROM (-e)
☐ Disable verify (-V)
☐ Do not write (-n)
☐ Disable flash erase (-D)
Verbosity: 0

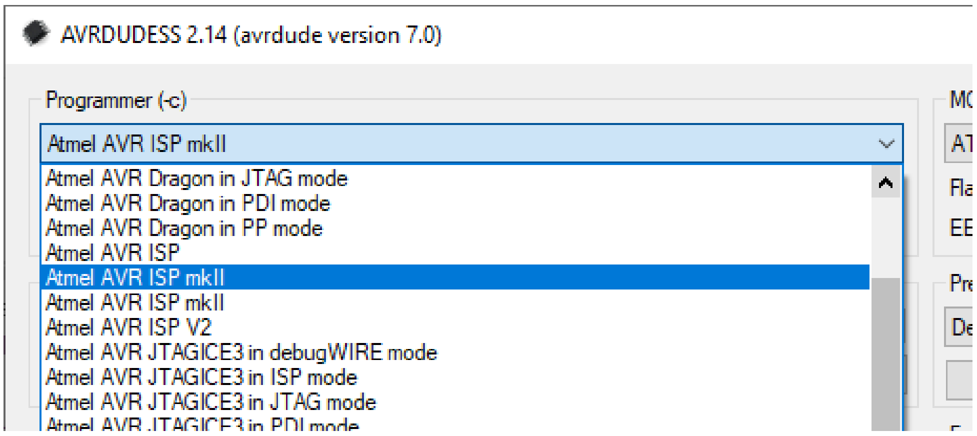
Additional command line args:

```
-c avrispmkII -p m2560 -P usb -U flash:w:"C:\Users\Windows 10 Pro\Downloads\u
```

```
>>>: avrdude
Chip-TesterPro-FW-v0.23.hex: 253,185 / 262,144 Bytes (96.58%)
```

Ready

The programmer “-c” option should select the first "ATMEL AVR ISP mkII" selection from the list shown because the installed driver shows the device name "avrispmkII".



AVRDUDESS 2.14 (avrdude version 7.0)

Programmer (-c)

Atmel AVR ISP mkII

- Atmel AVR Dragon in JTAG mode
- Atmel AVR Dragon in PDI mode
- Atmel AVR Dragon in PP mode
- Atmel AVR ISP
- Atmel AVR ISP mkII
- Atmel AVR ISP mkII
- Atmel AVR ISP V2
- Atmel AVR JTAGICE3 in debugWIRE mode
- Atmel AVR JTAGICE3 in ISP mode
- Atmel AVR JTAGICE3 in JTAG mode
- Atmel AVR JTAGICE3 in PDI mode

To verify that the programmer is detecting the ATMega2560 chip the “-p” option list box (ATmega2560) must be configured and then the "Detect" button pressed.

If the chip is detected then the flash area can be configured by selecting the new firmware HEX file to be uploaded to the chip, selecting "Write" and the format "Intel HEX".

Before updating the new Firmware, disconnected the programmer from the board and tested pressing the "GO" button on the Flash area, only to see if the programming command was correct:

```
    avrdude -c avrispmkII -p m2560 -P usb
    -U flash:w:"C:\<...>\Chip-TesterPro-FW-v0.23.hex":i
```

If all is correct then I reconnected the PCB board again and pressed "Go" to write the Firmware.

9.9 Overview of graphical interfaces for AVRDUDE

If you want, you can also use a graphical user interface to program the ATmega. These recommendations are given without any further support:

AVR8 Burn-O-Mat: a GUI for avrdude (Windows, Mac, Linux)

www.brischalle.de/avr8_burn-o-mat_avrdude_gui/

AVRDUDESS – A GUI for AVRDUDE (Windows, Mac, Linux)

https://blog.zakkemble.net/avrdudess-a-gui-for-avrdude/

AVRDUDESHELL (Russisch, Englisch)

http://matrex-notes.blogspot.com/search/label/AVRDUDESHELL

10 Appendix: Tweaks

This chapter contains some collected ideas.

10.1 Extension of the display connection

If the board is to be installed in a case, it may be necessary to extend the connection of the display using a ribbon cable.

The trivial method uses a 16-pin. ribbon cable for connecting the display to the board (or 12-pin if you do not connect the unused connections).

Alternatively, the following variants are conceivable.

10.1.1 Extension via IDE or floppy cable

Instead of the 16-pin socket become a 16-pin header fitted (or a "pin header to pin header" adapter plugged into the populated pin socket). The display can then be connected using an IDE or floppy ribbon cable. Since the sockets are in two rows (and one or four pins wider), make sure that the cable is plugged in correctly.

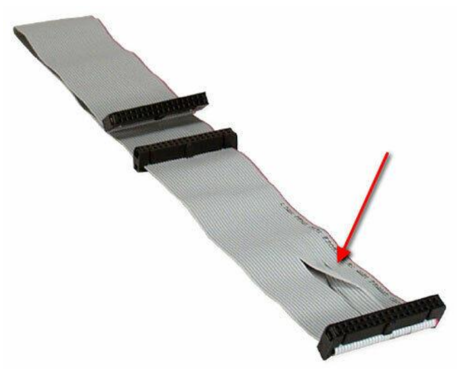


Figure 10.1: Floppy Cable

The flipped side of a floppy cable must be cut off (see arrow).

10.1.2 Extension via Jumper Cables

The display can easily be extended with a jumper cable (male-female).

Figure 10.2: Jumper Cable (male-female)

10.1.3 Saving connections

If the display is extended, up to four cables can be saved, so that only eight cables are required. For this purpose, the brightness and contrast control are relocated directly to the display.

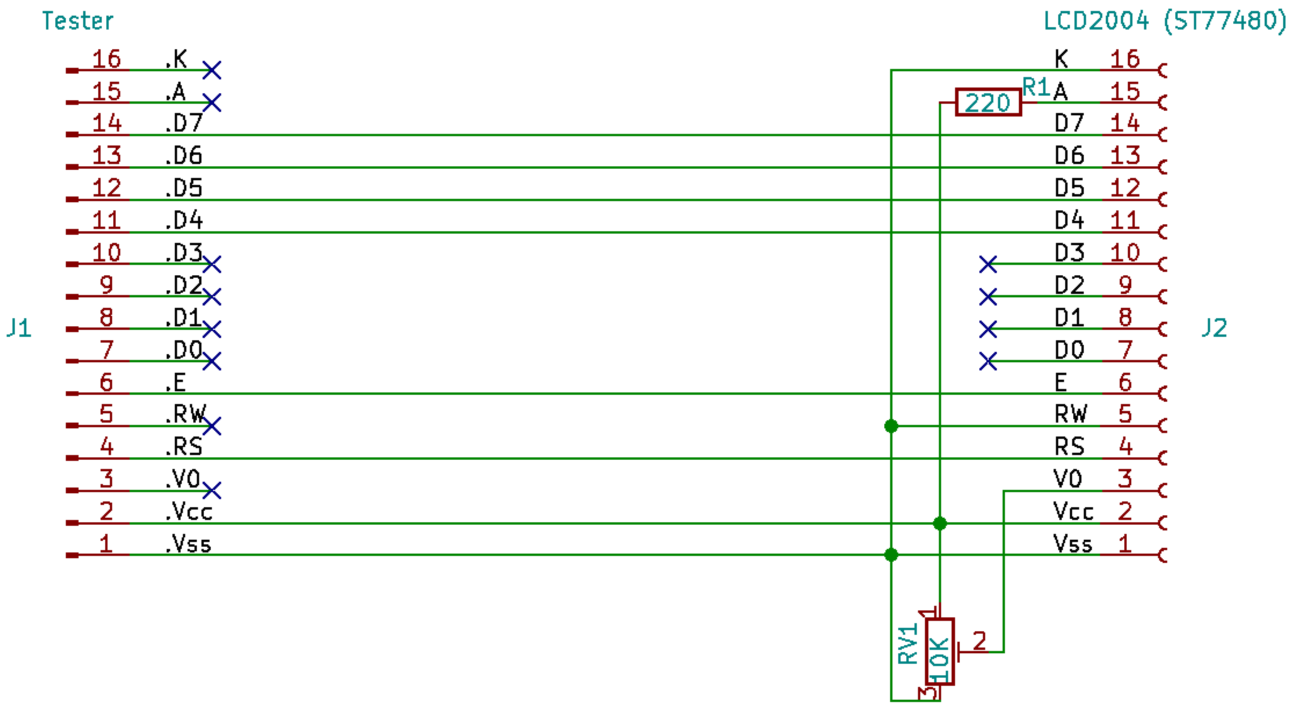


Figure 10.3: Wiring of the display

The 220 Ohm resistor is the series resistor for the display lighting. Usually. this resistance can be eliminated. The contrast is adjusted with the potentiometer/trimmer.

10.2 Easily accessible volume control

If it is necessary to adjust the volume frequently, the potentiometer can be soldered to the back of the circuit board. This makes it accessible without having to dismantle the display.

10.3 Transfer of data via WiFi

If several ROMs are to be dumped, plugging in the SD card can be a bit cumbersome. Direct access to the SD card via WiFi would be more practical.

The Chinese manufacturer of 3D printers ShenZhen BigTree Technology (https://www.bigtree-tech.com/) has developed an inexpensive WiFi adapter for SD cards for this purpose.

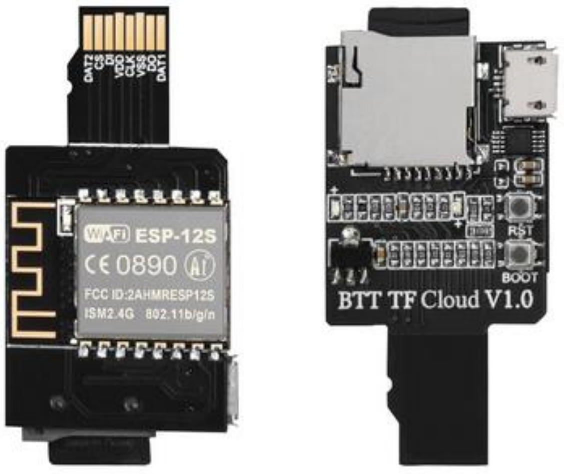


Figure 10.4: TF Cloud V1.0 adapter

Instructions and firmware are available on GitHub (https://github.com/bigtreetech/BTT-SD-TF-Cloud-V1.0). The adapter TF-Cloud V1.0 (Micro SD-Card) can be ordered e.g. on AliExpress for less than 10 EUR (https://www.aliexpress.com/item/4001031573171.html).

The adapter is configured by creating a `SETUP.INI` file on the SD card. This file contains two lines:

```
SSID=wlan-ssid
PASSWORD=password
```

The name of your own WLAN is entered as `wlan-ssid` and the password of the WLAN as `password`. Then the memory card is inserted into the "TF-Cloud" adapter and this is inserted into the SD micro card adapter.

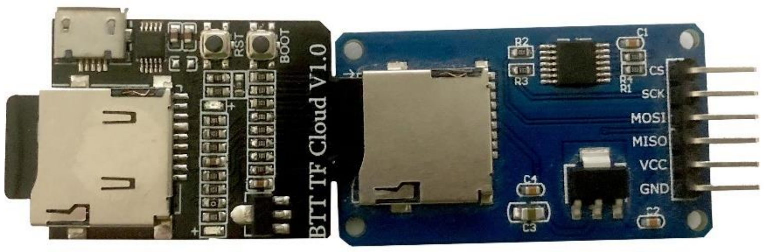
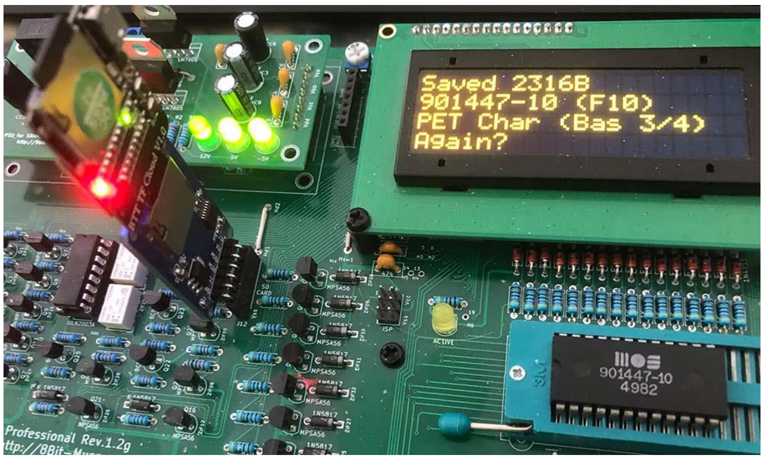


Figure 10.5: TF Cloud V1.0 Adapter (left), Micro-SD card adapter (right)

If you want, you can also connect the TF-Cloud to a computer with a USB cable. To do this, a driver for the CH340 chipset must also be installed. The adapter can then be reached via a virtual serial interface. With a serial monitor (115200 bps, e.g. via the Arduino software or any terminal program), debug messages from the adapter can be displayed, e.g. the assigned IP address is displayed at the beginning. If necessary, press RESET once on the adapter board beforehand.

But it is easier to look in the router to see which IP address the adapter has been assigned. For me, the adapter registered with the router with "WiFi-SD-Card-3DPrinter".



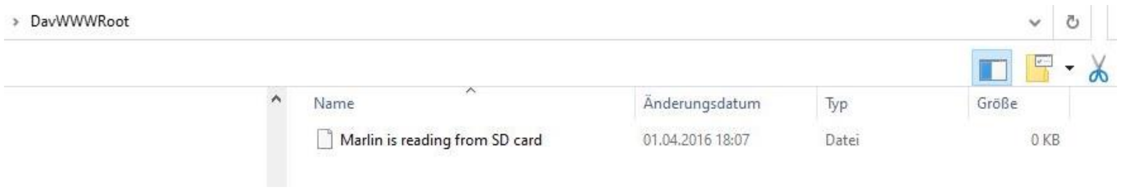
```
Saved 2316B
901447-10 (F10)
PET Char (Bas 3/4)
Again?
```

Figure 10.6: TF Cloud V1.0 adapter in use

The memory card is then accessed via Windows Explorer by entering:

```
\\IP-Adresse\DavWWWRoot
```

If the memory card is still in use, the following message appears:

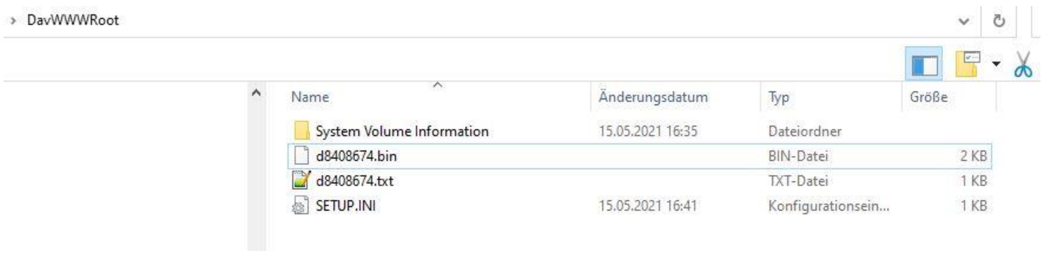


DavWWWRoot

Name	Änderungsdatum	Typ	Größe
Marlin is reading from SD card	01.04.2016 18:07	Datei	0 KB

Figure 10.7: TF Cloud V1.0 adapter in use

otherwise the content of the SD card is displayed.



DavWWWRoot

Name	Änderungsdatum	Typ	Größe
System Volume Information	15.05.2021 16:35	Dateiordner	
d8408674.bin		BIN-Datei	2 KB
d8408674.txt		TXT-Datei	1 KB
SETUP.INI	15.05.2021 16:41	Konfigurationsein...	1 KB

Figure 10.8: Contents of the SD card

The data can now be easily copied.

10.4 Adding a FUNC/TOP button (only up to rev.1.2i)

Due to some inquiries, the firmware from v.19 onwards includes the option of adding an additional button. Instead of pressing SELECT and JUMP at the same time, this button can then be used. Usually, it is used to jump back one menu level from a submenu.

Adding makes sense if the tester is built into a case. The button is connected to the soldering eye 19 / H6 (see picture):

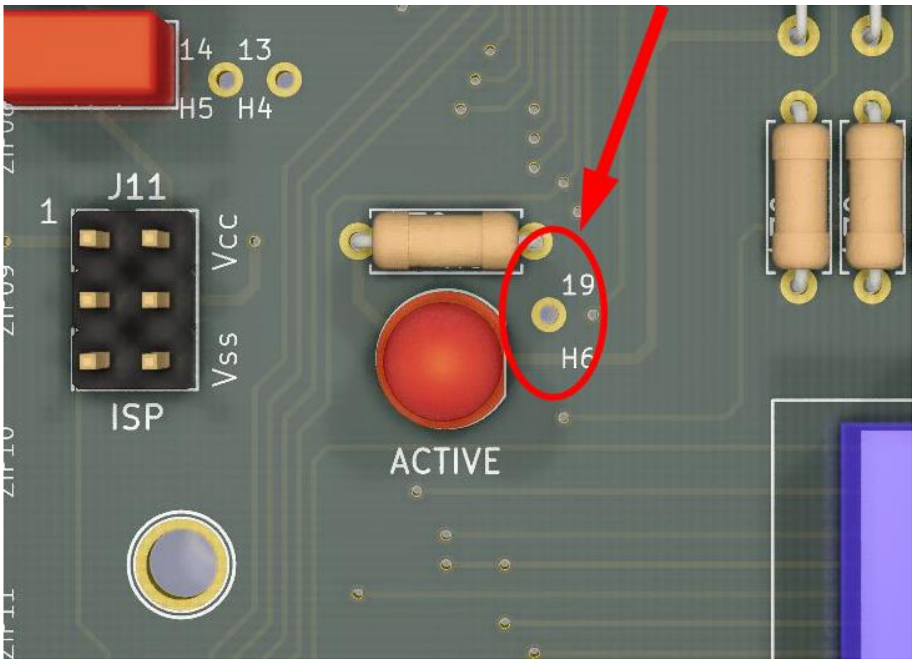


Figure 10.9: Connecting the push button (19/H6)

The button is simply connected to GND and this soldering eye. Ideally, the cable is soldered from below and laid on the back of the circuit board. GND can, for example, be used from the pin socket between the display and the DC/DC module. If you want to try the function first, you can try it out first with a plug-in cable (plug one end into the GND of the connector socket and briefly tap the soldering eye with the other end).

In addition, the function marked in yellow is then available:

IC menu / IC submenu:

SELECT BACK	JUMP FWRD	FUNC TOP	OK	Function
X				Previous chip
	X			Next chip
X	X			Jump to exit[3]
		X		Jump to main menu[4] or enter submenu
			X	Test selected chip
			keep pressed	Alternative mode in some cases

[3] in submenu only
[4] in submenu only