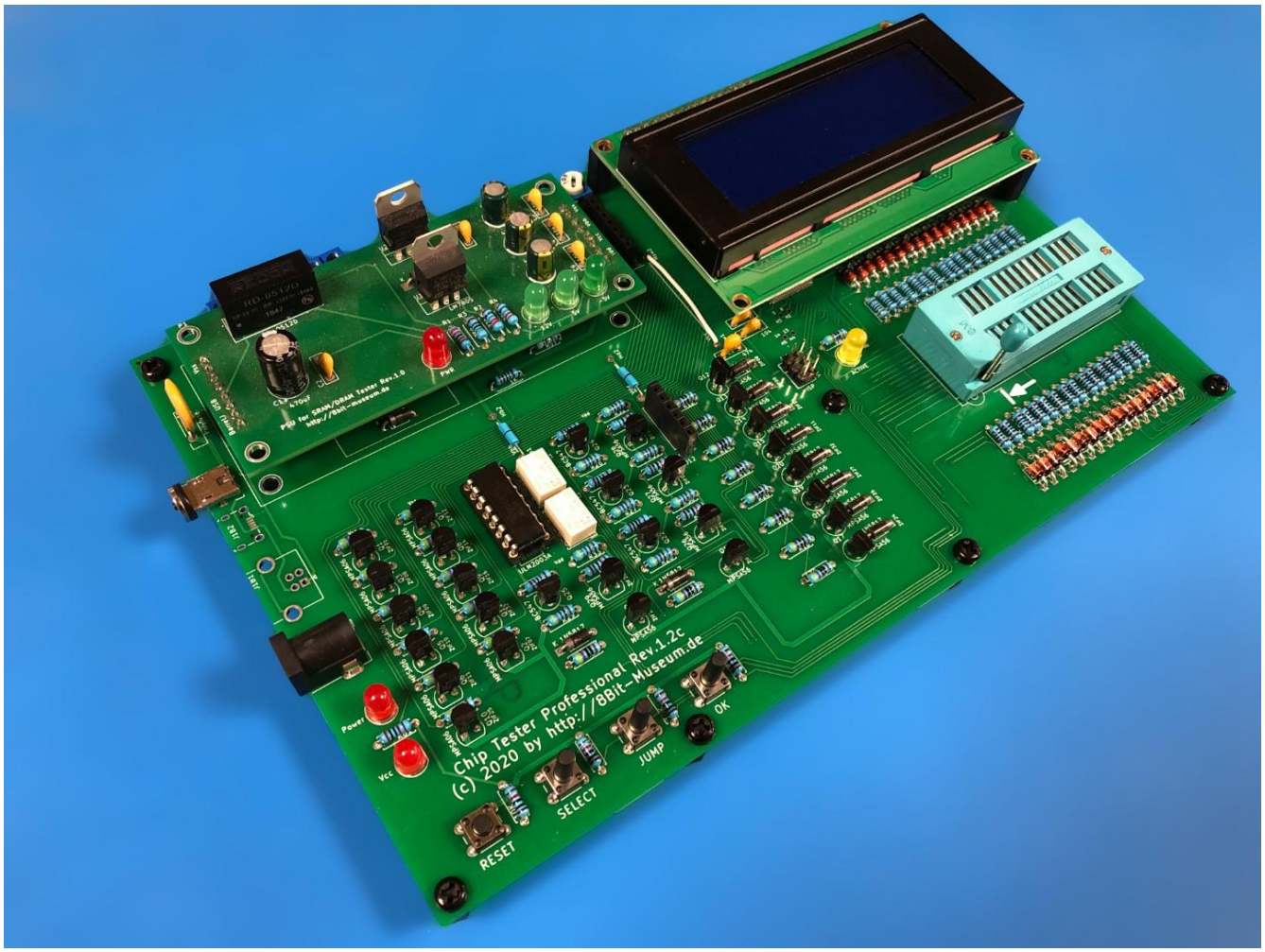
Version: 2022-11-09

Retro Chip Tester Professional

http://8bit-museum.de



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 note:

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 section 10.

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.

Warning notices are marked accordingly.

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!

There are some good "learning videos" for soldering the chip in the TQFN-100:

- 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

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 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 (products).

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

1.1 Overview

The Retro Chip Tester Pro was developed in order to test "old" memory chips from the 1970s and 1980s, which are often no longer recognized by today's programming devices.

Supported memory chips include (plus all pin-compatible types):

SRAMs (Examples)	
Size	**Chips (examples)**
4x4	74170, 74670
4x8	TC4036, TC4039
8x2	74172
16x1	7481 / 7484 (mit Adapter)
16x4	3101, 7489, 74189, 74219
16x9	74F211, 74F311
16x12	74F213, 74F313
32x8	74LS218, 74LS318
64x1	4505
64x4	74C910
64x9	82S09, 93419
128x8	6810
256x1	8216, 2116, 8217, 2700, 2701, 3106, 3107, 93410
256x4	2101A, 2111, 2112A, 74920, 74921, 6561
256x8	81C50, 81C51, 81C52, GTE3539, 8155, 8156
256x9	82S10, 82S12
512x4	D2113
512x8	81C54
1k x 1	2102, 8102, 2115, 2511
1k x 4	2114, 2148, 2149, 4045, 5114, 6514, C214, U224
1k x 4	6550
1k x 8	4118, 4801, 8185
2k x 1	HM6303H/L
2k x 8	2016, 2416, 4016, 4802, 4812, 6116, 6512, TC5516
4k x 1	2147, MK4104
4k x 4	7C168, 6168, 5416, P4C168, P4C169
8k x 8	2064, 2464, 6264, 2465
8k x 9	P4C163
16k x 1	8167, 6267, 6167, 2167, P4C167
16k x 4	P4C188, P4C198
32k x 8	20256, 61256, 62256, 71256, 43256, 43257
32k x 9	CY7C188
64k x 1	P4C187
64k x 4	P4C1258, P4C1281, P4C1298
64k x 8	61512, 24512
128k x 8	621024, 431000, P4C1023, 6173081, 638100, 63C1024
128k x 9	CY7C1088
256k x 1	P4C1257
256k x 4	P4C1026
512k x 8	AS6C4008, P4C1048, F7447APC
1024k x 1	P4C107

FIFO RAMs (Examples)	
Size	**Chips (examples)**
16 x 4	40105, 74LS222, 74LS224, 74LS227, 74LS228, 74LS232
16 x 5	74S225, 74LS229, 74LS233
64 x 4	74LS234, 74LS236, 74F413, 74HC7403
64 x 5	74LS235
256 x 9	7200, 7400 PLCC
512 x 9	7201, 7401 PLCC
1k x 9	7202, 7402 PLCC
2k x 9	7203, 7403 PLCC
4k x 9	7204, 7404 PLCC

NOVRAMs (Examples)	
Size	**Chips (examples)**
64 x 4	X22C10
256 x 4	X22C12
512 x 8	X20C04, X20C05
2k x 8	X20C16, X20C17

DRAMs (Examples)	
Size	**Chips (examples)**
1k x 1	MK4008
4k x 1	2104A, MK4015, MK4027, 7027, 2107
8k x 1	4108-x0/x1, MK4115-x0/x1, 2108H/L, 5298A/B
8k x 4	4408NLT/NLB
16k x 1	4116, 2117, 6116, 8116, 416, 2116, 3716, U256
16k x 1	2118, K565RU6
16k x 4	4416, 2620
32k x 1	3732H (4532-L4), 3732L (4532-L3)
32k x 1	4132 (Modul 1, Modul 2), MK4332 (Modul 1, Modul 2)
64k x 1	4164, 2600, K565RU5, 8264, 3764
64k x 4	4464, 41464, 50464
256k x 1	41256, 53256, 81256, MT1259
256k x 4	44256, 514256
1024k x 1	41024, 411000
256k x 8	256 kB SIMM / SIPP (30-pin)
256k x 1	256 kB SIMM / SIPP (30-pin) Parity only
1024k x 8	1024 kB SIMM / SIPP (30-pin)
1024k x 1	1024 kB SIMM / SIPP (30-pin) Parity only
64k x 4	ZIP 20
256k x 1	ZIP 16
256k x 4	ZIP 20
1024k x 1	ZIP 20
1024k x 4	ZIP 20

ROMs / PROMs / EPROMs (Examples)	
Type	**Chips (examples)**
2k x 8	6540
(P)ROM	2308, 2316A, 2316B, 2332, 2364, 23128, 23256, 23512, 231000/231001, 232000, 234000, RO-3-2513, 6670, 2513 (with adapter)
EPROMs	2704, 2708, 2716, TMS2716, 2732, 2764, 27128, 27256, 27512, 271001, 272001, 274001, 1702 (with adapter), 4204/5204 (with adapter) , TMS2716, TMS2564, CDP18U42
C64	2x 8k Cartridges
VC20	8k to 64k Cartridges
VCS/2600	2k, 4k, 8k, (16k) Cartridges

bipolar ROMs / PROMs (Examples)	
Type	**Chips (examples)**
32 x 8	7488, 74188, 74288
256 x 4	74187, 74287, 74387
256 x 8	74S271, 74S371, 74S470, 74S471
512 x 4	74S270, 74S370, 74S570, 74S571
512 x 8	74S472, 74S473, 74S474, 74S475
1k x 4	74S476, 74S477, 74S572, 74S573
1k x 8	74S478, 74S479
2k x 4	82S184, 82S185
2k x 8	82S190, 82S191
4k x 4	82S195
4k x 8	82S321
8k x 8	82S641

Logic Chips	
Type	**Chips**
74xxx	7400, 01, 01(H), 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 51(LS), 52, 53, 53(H), 54, 54(H), 54(LS), 55(H), 55(LS), 56, 57, 58, 60, 61, 62, 63, 64, 65, 68, 69, 70, 71(H), 71(L), 72, 73, 74, 75, 76, 76(LS), 77, 78(H), 78(L), 78(LS), 79, 80, 82, 83, 85, 85(C), 86, 86(L), 87, 90, 91, 92, 93, 93(C), 93(L), 94, 95, 95(C), 95(L), 96, 97, 98, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 118, 119, 120, 121, 122, 123, 125, 126, 128, 130, 131, 132, 133, 133+, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 221, 230, 231, 237, 238, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 251, 253, 256, 257, 258, 259, 260, 261, 265, 266, 269, 273, 275, 276, 278, 279, 280, 283, 284, 285, 286, 290, 292, 293, 294, 295, 298, 299, 322, 323, 347, 348, 350, 351, 352, 353, 354, 355, 356, 357, 363, 364, 365, 366, 367, 368, 373, 374, 375, 376, 377, 378, 379, 381, 382, 384, 385, 386, 388, 390, 393, 395, 396, 398, 399, 405, 412, 416(S), 422, 423, 425, 426, 432, 436, 437, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 456, 461, 465, 466, 467, 468, 490, 518, 519, 520, 521, 522, 524, 526, 527, 528, 533, 534, 537, 538, 539, 540, 541, 543, 544, 545, 546, 547, 547(F), 548(F), 560, 561, 563, 564, 566, 567, 568, 569, 573(LS), 574, 575, 576, 577, 579, 580, 583, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 604, 605, 606, 607, 620, 621, 622, 623, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 651, 652, 653, 654, 657, 666, 667, 668, 669, 671, 672, 673, 674, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 696, 697, 698, 699, 740, 741, 744, 748, 756, 757, 758, 759, 760, 762, 763, 793, 794, 795, 796, 797, 798, 800, 802, 804, 805, 808, 810, 821, 822, 823, 824, 825, 826, 832, 841, 842, 843, 844, 845, 846, 848, 850, 851, 857, 867, 869, 873, 874, 876, 878, 879, 880, 885, 900(ALS), 901(C), 902(ALS), 902(C), 903(ALS), 903(C), 904(C), 906(C), 914(C), 915(C), 940, 941, 990, 991, 992, 993, 994, 995, 996, 3037, 7001, 7002, 7014, 7032, 7266, 8541, 9034, 9035, 9114, 9115, 9134, 9135, 9240, 9244, 9245

Logic Chips	
Type	**Chips**
Misc	75121, 75122, 75123, 75124, 75125, 75127, 75138, 75140, 75154, 75160, 75172, 75173, 75189, 75450, 75451, 75452, 75453, 75454, 75460, 75461, 75462, 75463, 75464, 75466, 75467, 75468, 75469, 75470, 75471, 75472, 75473, 75474, 75491, 75492, 75494, 75497, 75498, 7707 MOS, 7708 MOS, 7709 MOS, 7711 MOS, 7712 MOS, 7713 MOS, 7714 MOS, 7715 MOS, 80C95, 80C96, 80C97, 80C98, 81LS95, 81LS96, 81LS97, 81LS98, 82C19, 8205, 8212/3212, 8216/3216, 8226/3226, 8233, 8234, 8235, 8241, 8242, 8250, 8251, 8252, 8263, 8264, 8266, 8267, 8270, 8271, 8282, 8283, 8286, 8287, 8290, 8291, 8415A, 8455A, 8470A, 8471A, 8480A, 8481A, 8490A, 8708 MOS, 8713 MOS, 8808A, 8815A, 8816A, 8829A, 8840A, 8848A, 8855A, 8870A, 8875A, 8880A, 8881A, 8885A, 8890A, 8891A, 8H16, 8H70, 8H80, 8H90, 8T10, 8T13, 8T14, 8T22, 8T23, 8T24, 8T26/MC6880, 8T28/MC6889, 8T37, 8T38, 8T93, 8T94, 8T95/MC6885, 8T96/MC6886, 8T97/MC6887, 8T98/MC6888, 8T125, 8T245, Am25LS07, Am25LS08, Am25LS09, Am25S10, Am25LS14, Am25LS15, Am25LS22, Am25LS23, Am25LS2518, Am25LS2519, Am25LS2521, Am7303, Am7304, Am7307, Am7308, AmZ8121, Am8303, Am8304, Am8307, Am8308, Am93S10, Am93S16, Am9341, DM8090, DM8091, DM8092, DM8093, DM8094, DM8095, DM8096, DM8097, DM8098, DM8121, DM8123, DM8131, DM8160, DM8530, DM8532, DM8533, DM8560, DM8563, DM8570, DM8590, DM8830, DM8837/7837, DM8838/7838, DM9002, DM9003, DM9004, DM9009, DM9012, DM9016, DM9024, DM9300, DM9301, DM9310, DM9314, DM9316, DM9322, DM9368, DM9370, DS1630, DS1631, DS1632, DS1633, DS1634, DS3630, DS3631, DS3632, DS3633, DS3634, DS3662, DS8640/7640, DS8641/7641, DS8810/7810, DS8811/7811, DS8812/7812, DS8819/7819, DS8833/7833, DS8835/7835, DS8837/7837, DS8838/7838, BA12003, CA3045, CA3046, CA3081, CA3082, CA3083, CA3086, CA3161, L20x, L70x, QS3384, TD6208x, TD6278xRev8, ULN200x, ULN201x, ULN202x, ULN2064, ULN2074/76, ULN280x, ULN282x, UDN6118, UDN298xRev8, LMx39, uA741, FCJ121Rev, D345, D346, D492, D718, K155IE1, KR559IP1, KR559IP2, SNG4xRev5, SNG6xRev5, SNG9xRev5, SNG13xRev5, SNG14xRev5, SNG15xRev5, SNG16xRev5, SNG19xRev5, SNG22xRev5, SNG23xRev5, SNG24xRev5, SNG26xRev5, SNG32xRev5, SP302A, SP304A, SP305A, SP306A, SP314A, SP317A, SP334A, SP337A, SP357A, SP358A, SP370A, SP374A, SP375A, SP377A, SP380A, SP381A, SP384A, SP387A, SP391A, CBM251641-02, CBM906114-01

Logic Chips	
Type	**Chips**
4xxx	4000, 01, 02, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 40, 41, 42, 43, 44, 48, 49, 50, 51, 52, 53, 54, 55, 56, 59, 60, 63, 66, 67, 68, 69, 70, 71, 72, 73, 75, 76, 77, 78, 81, 82, 85, 86, 93, 94, 95, 96, 97, 98, 99, 014, 097, 098, 101, 102, 103, 104, 106, 107, 109, 110, 116, 147, 160, 161, 162, 163, 174, 175, 192, 193, 194, 240, 244, 245, 257, 373, 374, 511, 4316, 4428, 4501, 02, 03, 04, 06, 07, 08, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 22, 26, 28, 29, 30, 31, 32, 38, 39, 43, 55, 56, 58, 60, 61, 72, 74, 84, 85, 98, 4723, 4724, 4929, 4930, 4931, 4934, 4935, 49700, 49701, 49702, 49703, 49704, 49705, 49710, 49711, 49713, 49714
DDR DLxx	000, 002, 003, 004, 008, 010, 011, 014, 020, 021, 030, 032, 037, 038, 040, 051, 074, 086, 090, 093, 112, 123, 132, 155, 164, 175, 192, 193, 194, 251, 253, 257, 259, 374, 540, 541, 299
UdSSR 153-555	AP3, AP4, AP5, AP6, AP9, AP10(K1102), AP10, AP11P6(K1102), AP12P6(K1102), AP13P6(K1102), AP14P6(K1102), AP14, AP15, AP16, AP17, AP24, AP26, ID1, ID3, ID4, ID5, ID6, ID7, ID10, ID14, ID18, ID22, ID24, IE1, IE2, IE4, IE5, IE6, IE7, IE8, IE9, IE10, IE11, IE12, IE13, IE14, IE15, IE16, IE17, IE18, IE19, IE20, IK2, IM1, IM2, IM3, IM5, IM6, IM7, IP1(KR559), IP2(KR559), IP2, IP3, IP4, IP5, IP6, IP7, IP8, IP9, IR1, IR2, IR5, IR8, IR9, IR10, IR11, IR12, IR13, IR15, IR16, IR22, IR23, IR24, IR25, IR27, IR28, IR29, IR30, IR33, IR34, IR35, IR37, IR38, IR40, IR41, IR42, IR43, IR52, IV1, IV2, IV3, KP1, KP2, KP5, KP7, KP11, KP12, KP13, KP14, KP15, KP16, KP17, KP18, KP19, KP20, KT1(K1010), LA1, LA2, LA3, LA4, LA6, LA7, LA8, LA9, LA10, LA11, LA12, LA13, LA16, LA18P6(K1102), LA19, LD1, LE1, LE2, LE3, LE4, LE5, LE6, LE7, LE8, LE11, LI1, LI2, LI3, LI4, LI5P6(K1102), LI6, LI7, LI9, LL1, LL2P6(K1102), LL3, LN1, LN2, LN3, LN4, LN5, LN6, LN7, LP5, LP8, LP9, LP10, LP11, LP12, LP13, LP14, LR1, LR3, LR4, LR9, LR10, LR11, LR13, PP4, PR6, PR7, SP1, SP2, TL1, TL2, TL3, TM2, TM5, TM7, TM8, TM9, TM10, TR2, TV1, TV6, TV9, TV10, TV11, TV15
UdSSR 561 CMOS	ID1, ID6, ID7, ID23, IE4, IE5, IE8, IE9, IE10, IE11, IE14, IE16, IE19, IE20, IE21, IM1, IP2, IP6, IR2, IR6, IR9, IR14, KP1, KP2, KP3, KP4, KP5, KT1, KT3, LA7, LA8, LA9, LA10, LE5, LE6, LE10, LI2, LN1, LN2, LN3, LP1, LP2, LP4, LP14, LS2, PR1, PU2, PU3, PU4, PU6, PU7, PU9, SA1, TL1, TM2, TM3, TR2, TV1, UM1
Misc	DIS1417, DL1414, DL2416Rev1, HP730x, HP7340, TIL306/307, TIL308/309, TIL311, VQB76, LED1x7Seg., LED1x7Seg., LED1x7Seg., LED1x7Seg., LED1x7Seg., LED1x7Seg., LED2x7Seg., LED2x7Seg., LED2x7Seg., LED2x7Seg., LED2x7Seg., LED2x7Seg., LED2x7Seg., LED2x7Seg., LED3x7Seg., LED3x7Seg., LED4x7Seg., LED4x7Seg., 8x8DotMatrix, 8x8DotMatrix

To test east types, the corresponding west counterpart can also be selected.

Some tests were implemented according to data sheets. The overview list for logic chips gives an overview of which tests have been verified. If a "false negative" is reported for a logic chip, i.e. a known good chip tests as defective, please report this to the developer as an error.

Logic Chips	
PAL	Readout of PALs (not „registered”, only purely combinatorial types), e.g. PAL10H8, PAL12H6, PAL14H4, PAL16H2, PAL10L8, PAL12L6, PAL14L4, PAL16L2, PAL16C1

Programmable Chips	
EPROM	2708 (adapter required)
EPROM	2716/2516 (adapter required)
EPROM	TMS2716 (adapter required)
EPROM	2532 (adapter required)
EPROM	2732 (adapter required)
EPROM	2564 (adapter required)

Other features:

- Over 120 different SRAMs (>700 on the comparison list), several FIFO RAMs, and over 30 different DRAMs (>250 on the comparison list) can be tested.
- You can define twelve SRAMs, three DRAMs, and three ROMs yourself.
- Over 80 different ROMs, PROMs and EPROMs can be read out (>650 on the comparison list).
- ROMs can be identified via the CRC32 (external database files contain approx. 400,000 ROMs).
- Manufacturer and type for more than 420 EPROMs can be identified.
- Six EPROMs can be programmed.
- Over 1200 logic ICs can be checked.
- Search for a chip using a built-in database.
- The contents of the supported ROMs / PROMs / EPROMs can be saved on an SD memory card (a suitable SD card adapter is required).
- PALs/GALs can be read out combinatorially (with restrictions).
- All pins have overvoltage protection up to 40V and are short-circuit proof.

Since the tester uses a fairly cheap ATmega2560, some minor compromises had to be made. The known problems and limitations are discussed in Section 11. Even if not all errors can be identified, the results should be sufficiently valid for most applications.

1.2 Let’s go

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

Components required:

First of all, getting the components is important. The document "BOM - ENGLISH.pdf" is available for this purpose (not public available). It 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.

Assembling and programming:

The most important document is this manual. The assembly is discussed in the first chapters:

- Chapter 2: Assembly
- Chapter 3: Power supply
- Chapter 4: Programming the Tester (fuses and firmware)
- Chapter 14: Appendix: Examples of programming the ATmega

There are also additional files for this:

- 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.

The other chapters deal with the first start-up, features, configuration, own experiments, possible problems and troubleshooting, and the design of the Tester. They should be read briefly at least once.

2 Assembly

The parts lists of the individual versions of the boards are in a separate document. The components are usually all labeled (“104” = 100nF, “224” = 220nF, “22” = 22pF).

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.

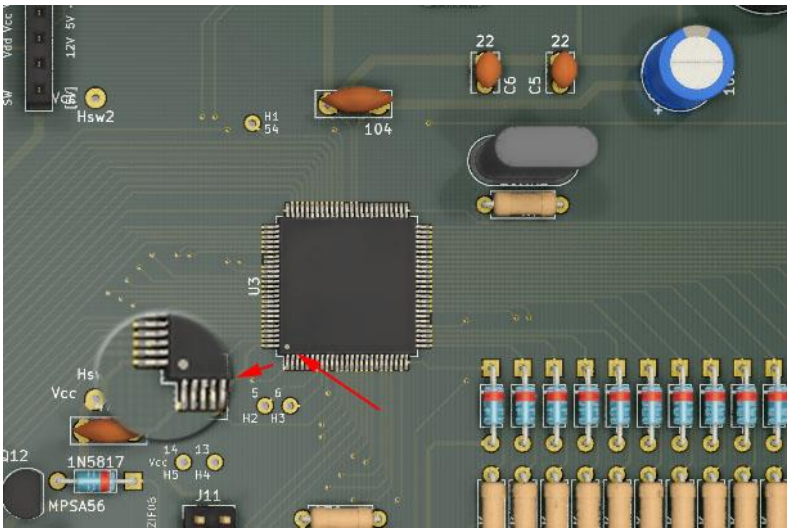


Figure 2.1: Orientation of the ATmega2560

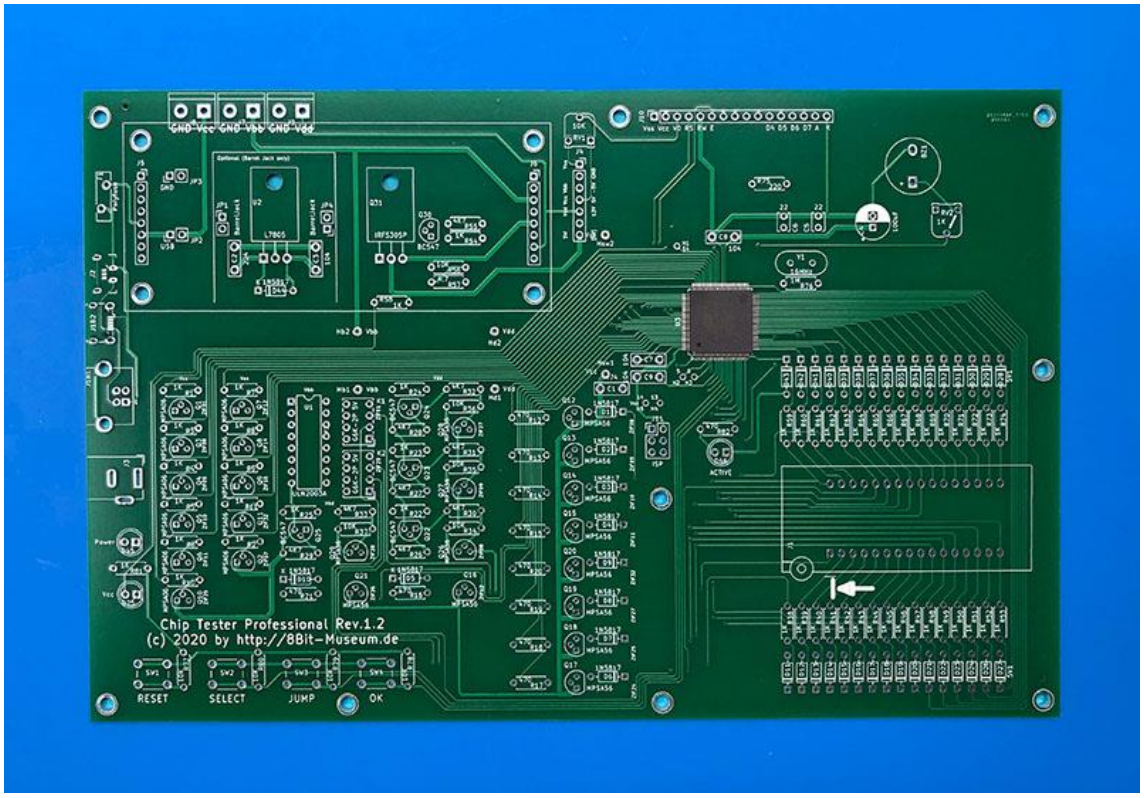


Figure 2.2: PCB equipped with the ATmega2560

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.

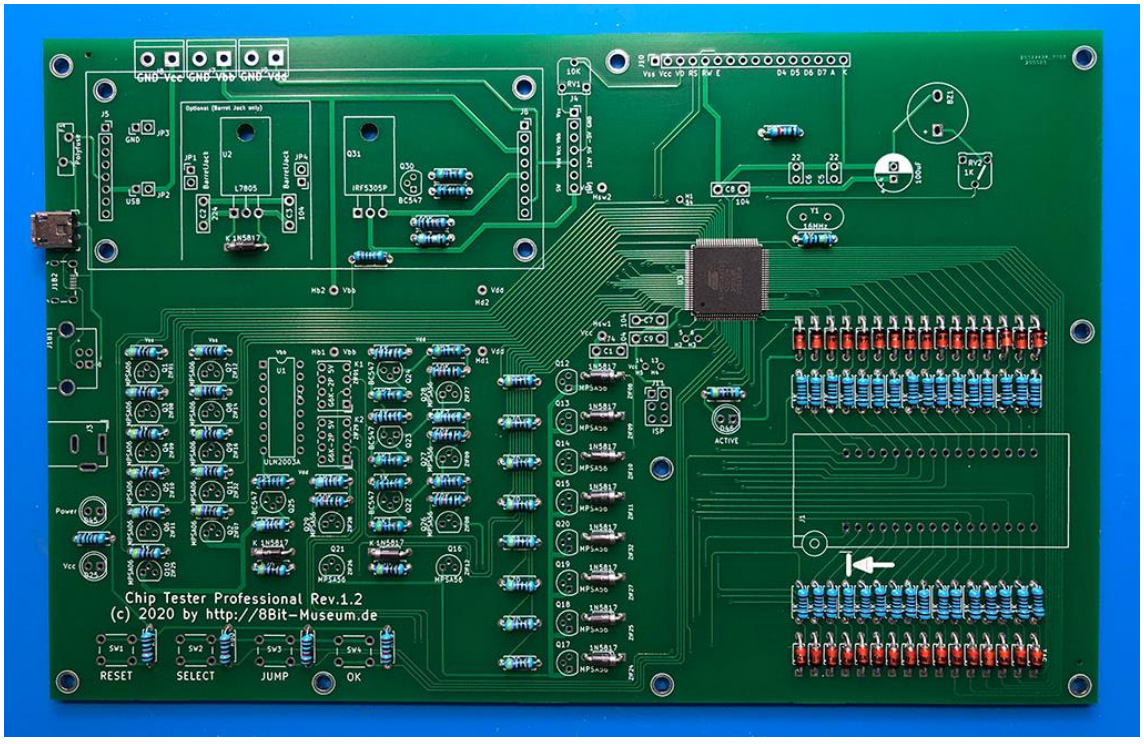


Figure 2.3: PCB equipped with Z-diodes and resistors

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!

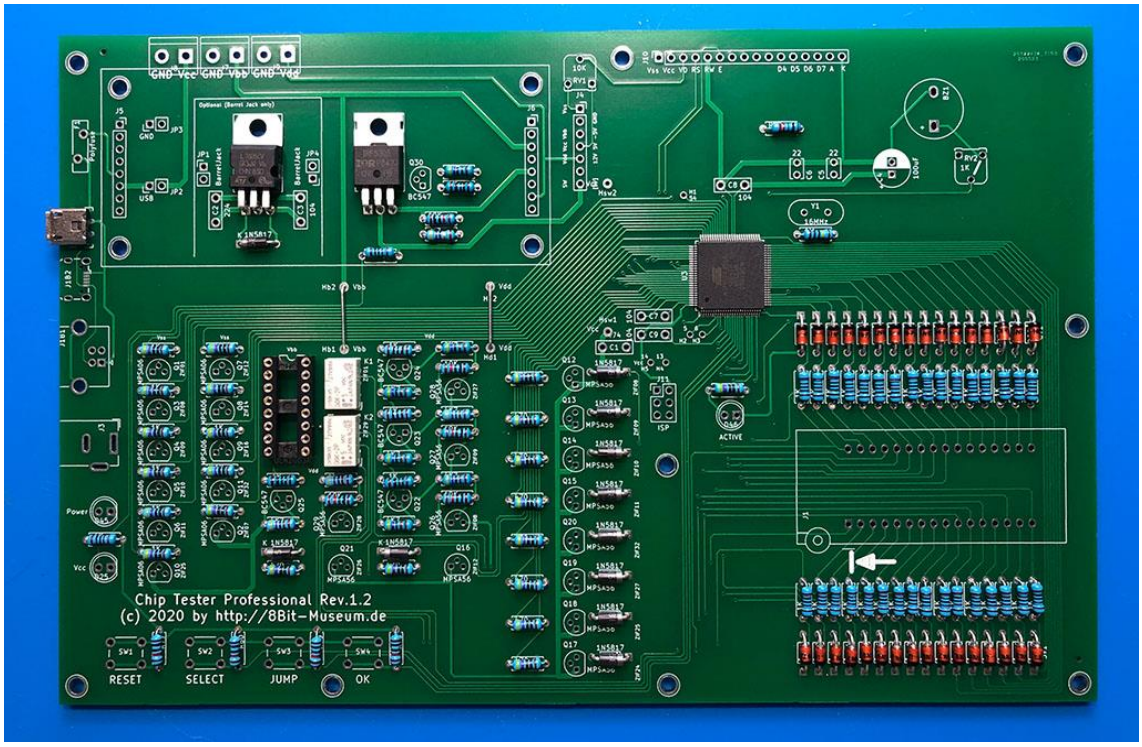


Figure 2.4: PCB equipped with relays and voltage regulators

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

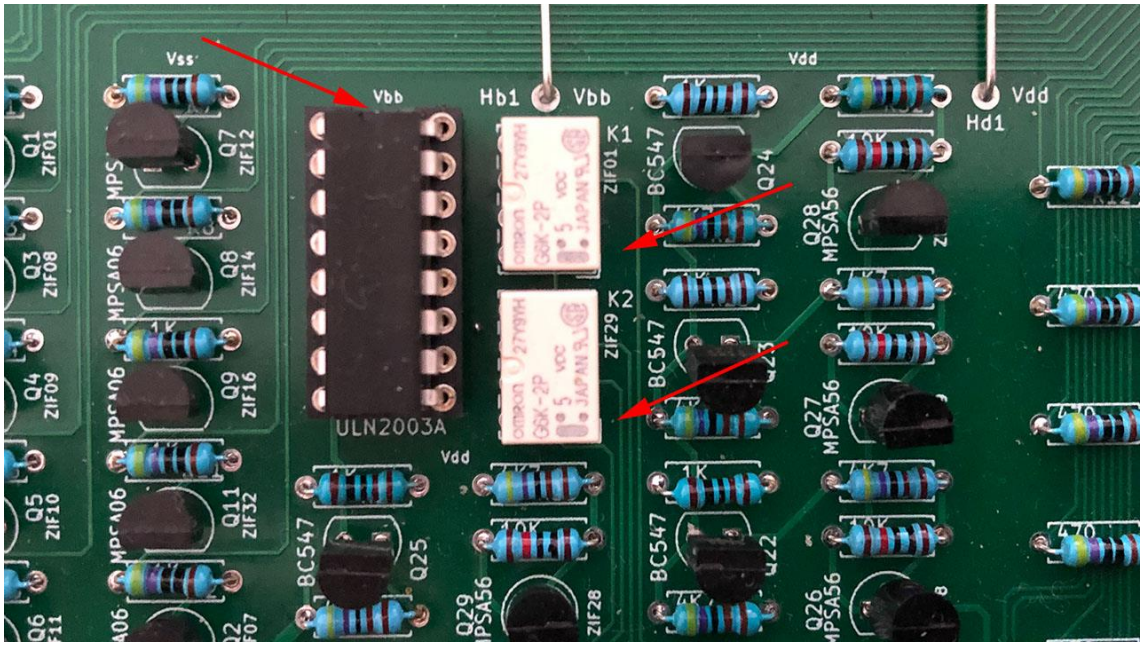


Figure 2.5: Detail of PCB equipped with relays and IC.

The wire jumpers again in detail:

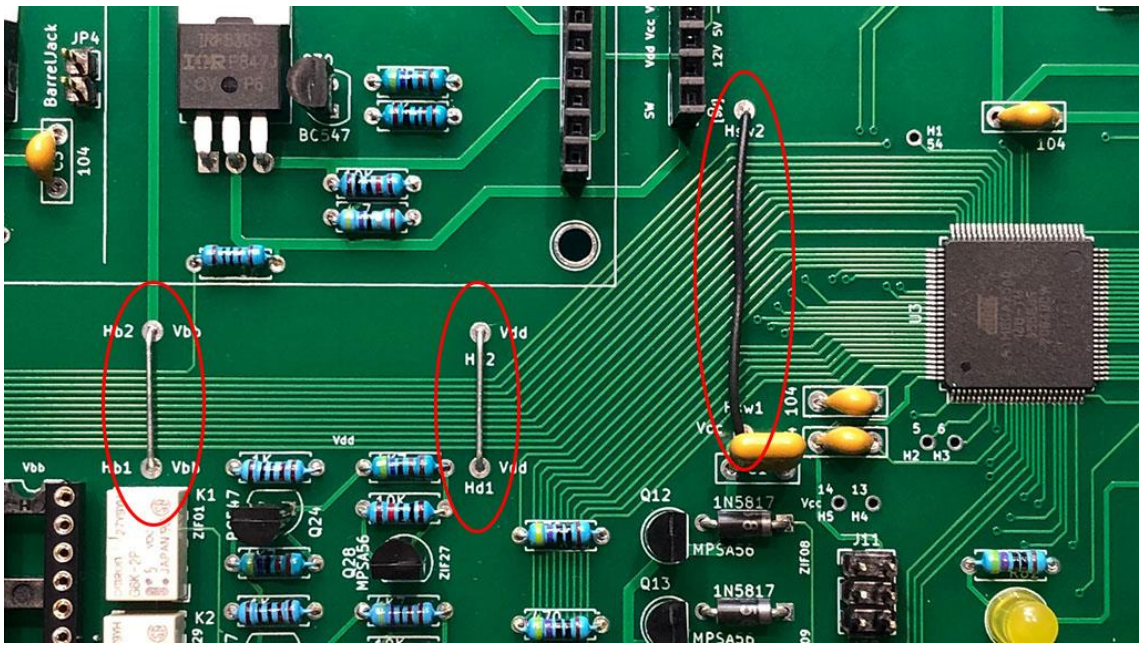


Figure 2.6: Wire jumpers

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!

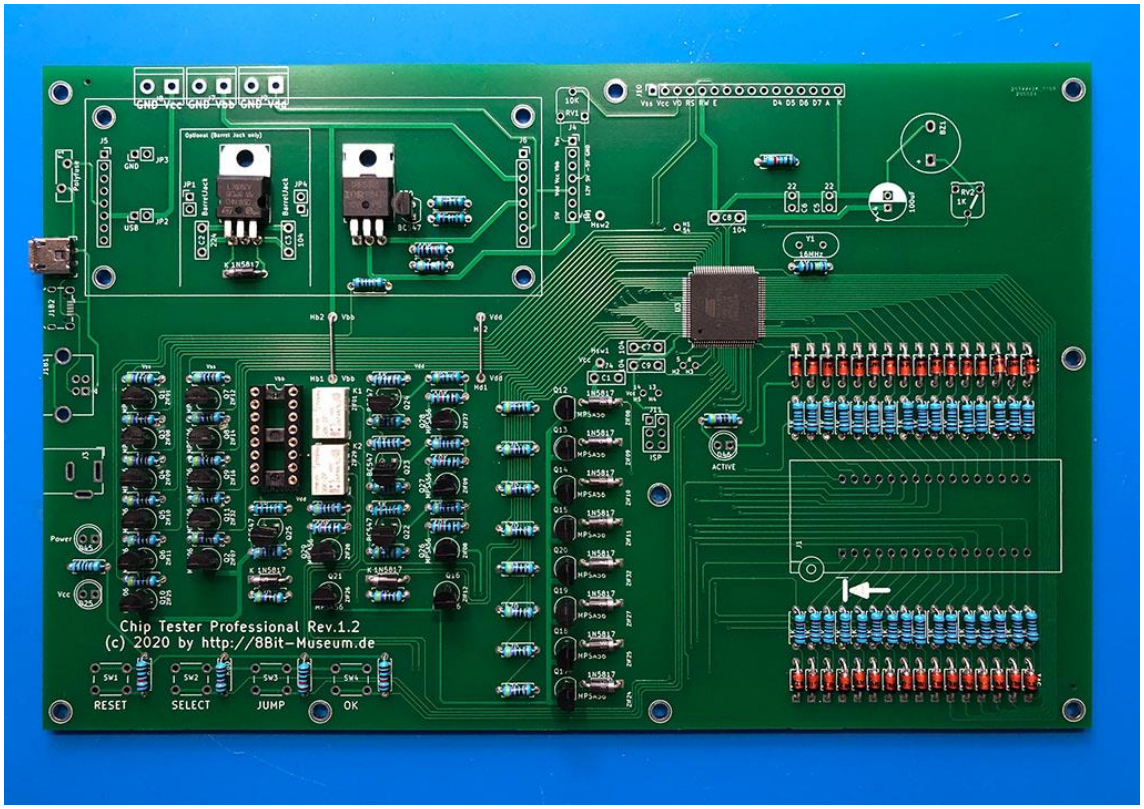


Figure 2.7: PCB equipped with the transistors

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:

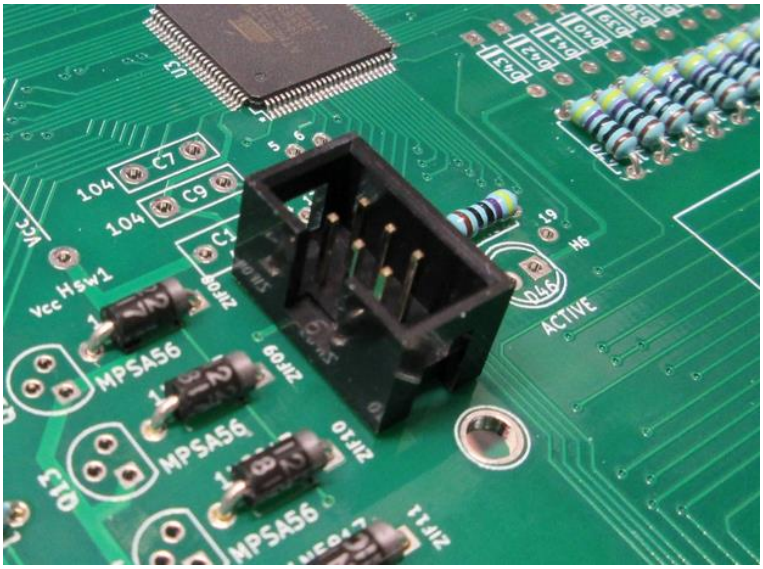


Figure 2.8: Shrouded Header (orientation)

The assembled board looks like this:

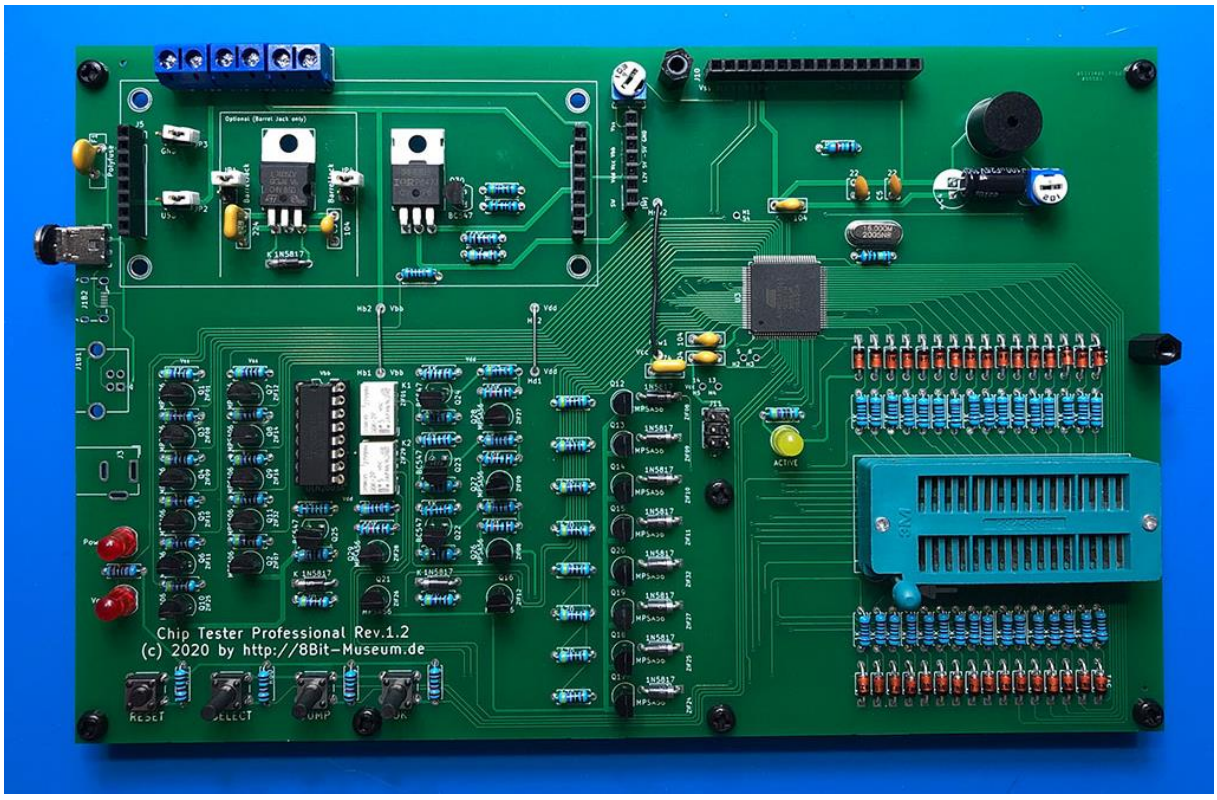


Figure 2.9: The assembled board

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

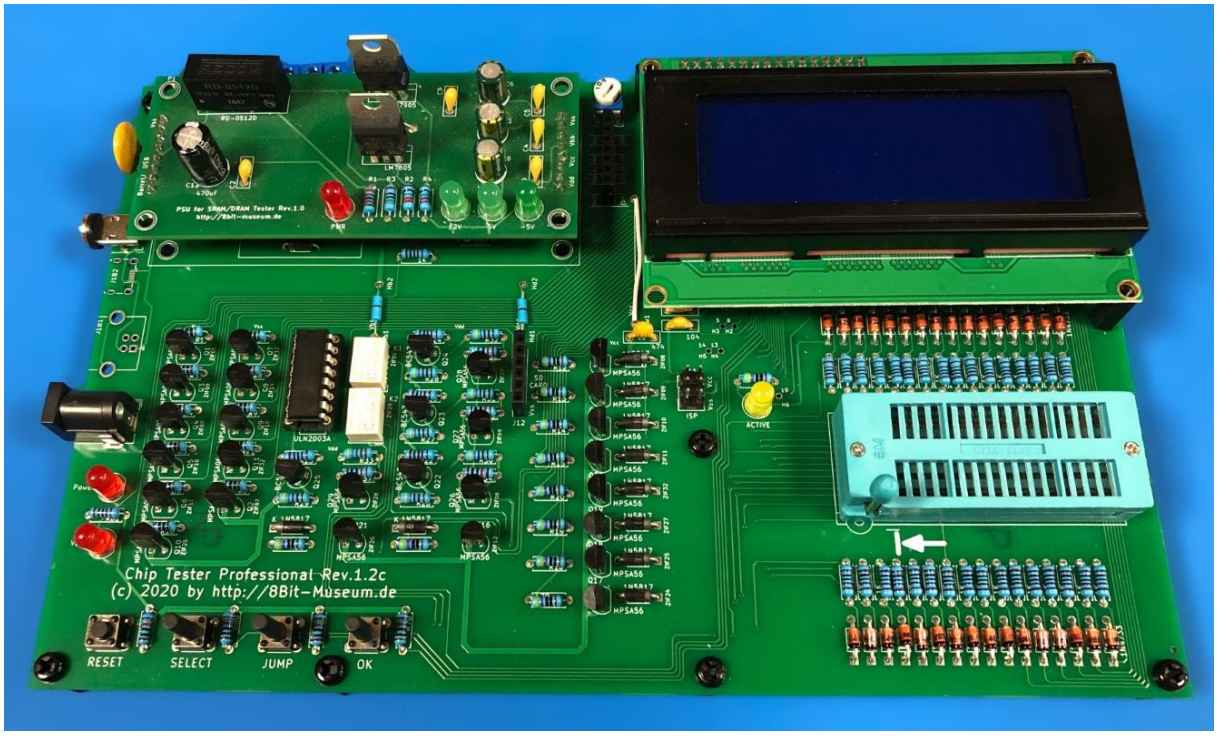


Figure 2.10: The fully assembled PCB

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:

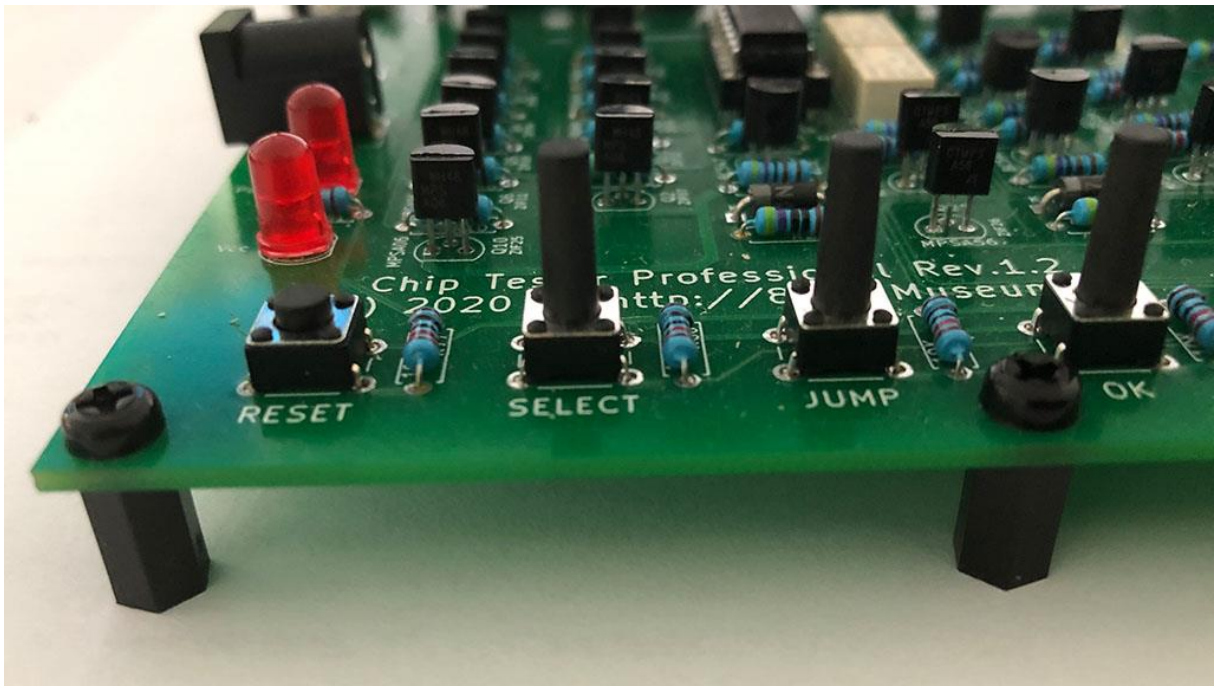


Figure 2.11: Assembly of feet

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

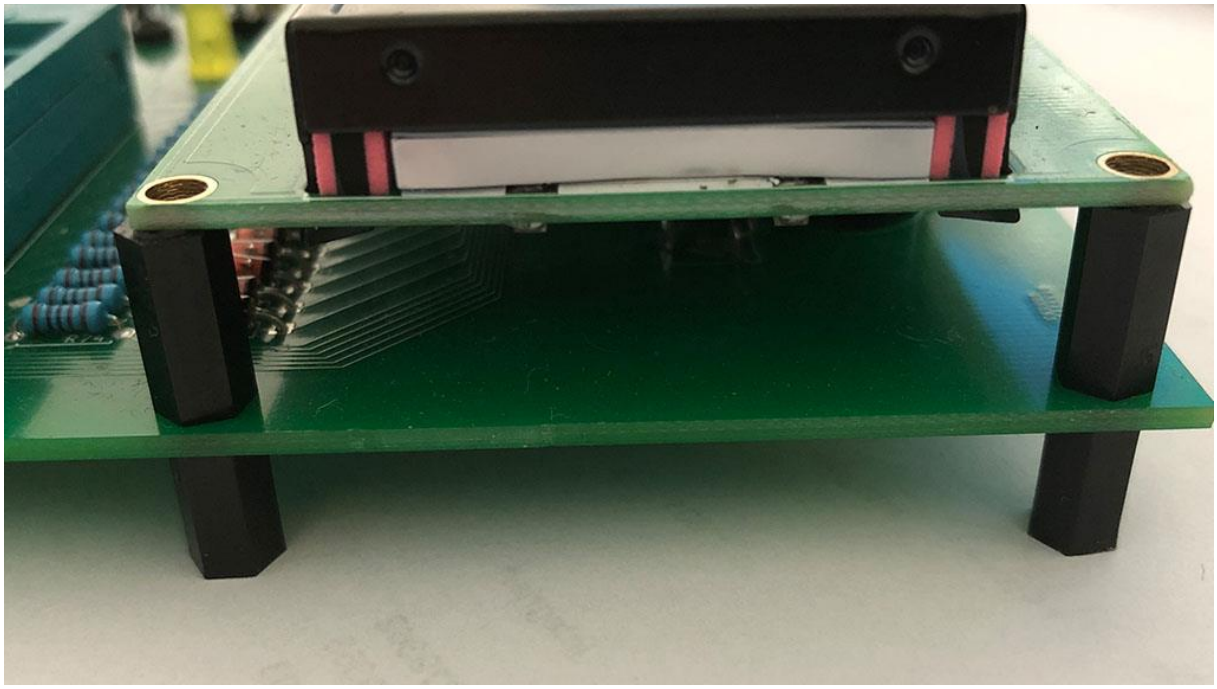


Figure 2.12: Assembly of the display

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).

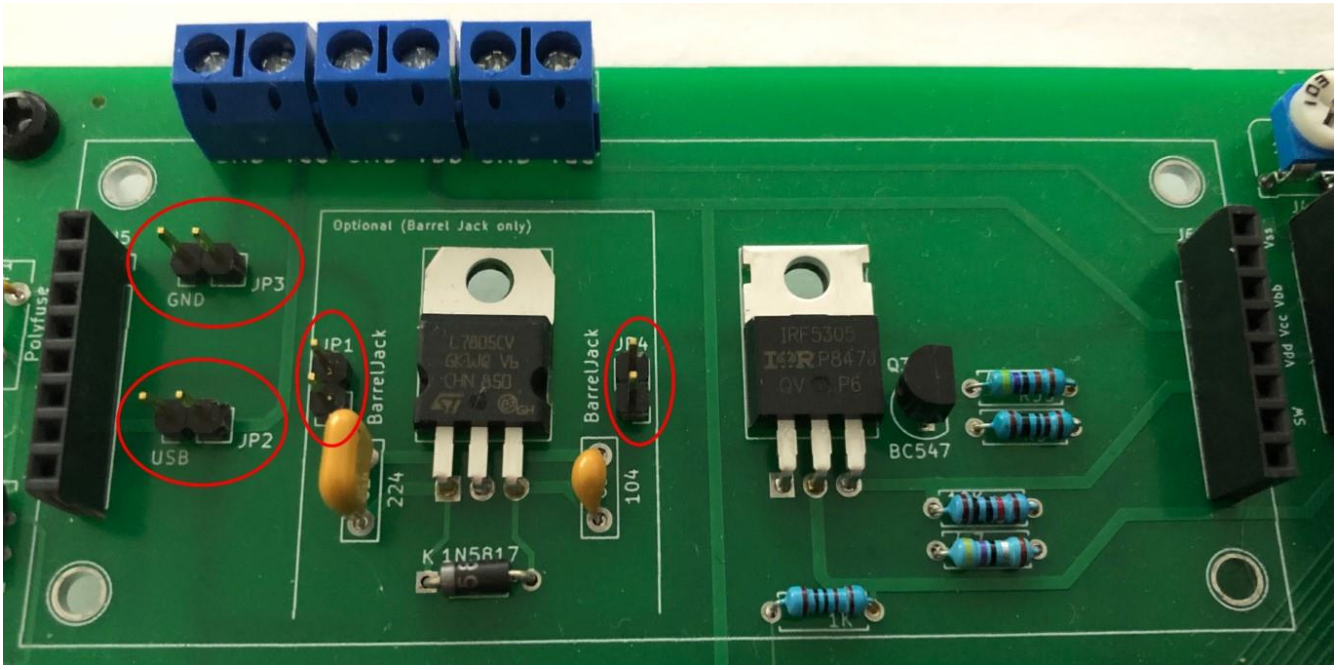


Figure 3.2: All jumpers open

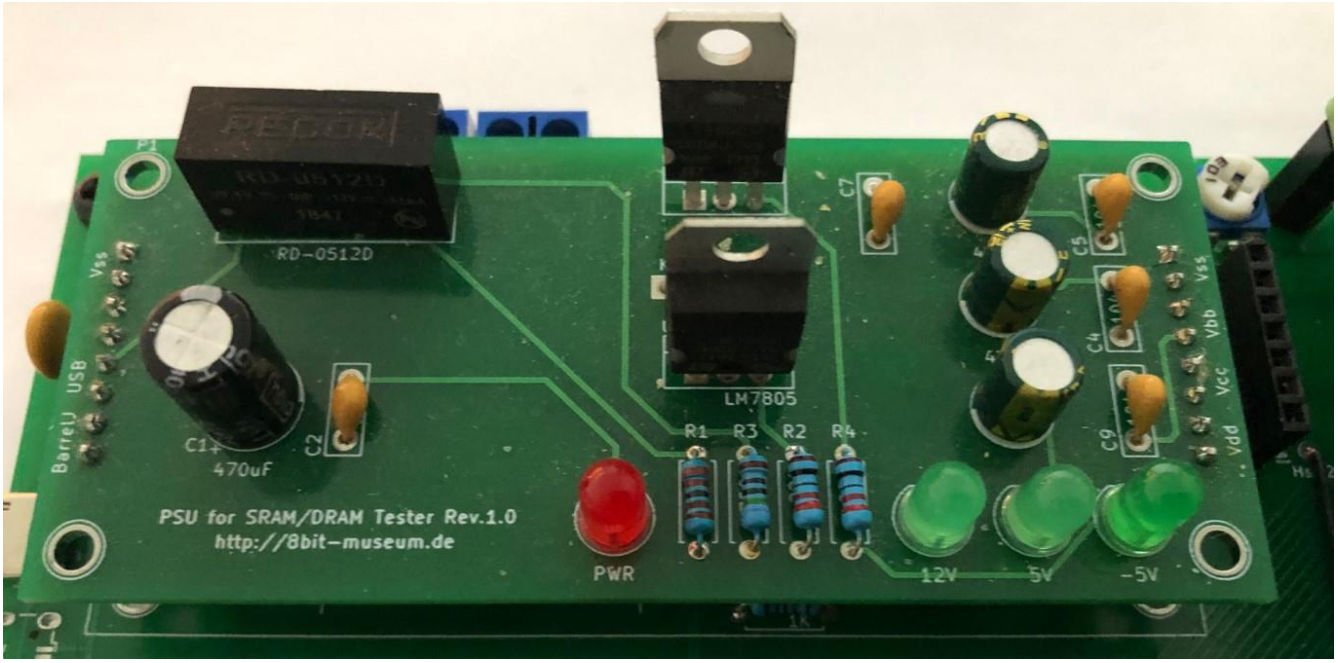


Figure 3.3: Plugged DC/DC module

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.

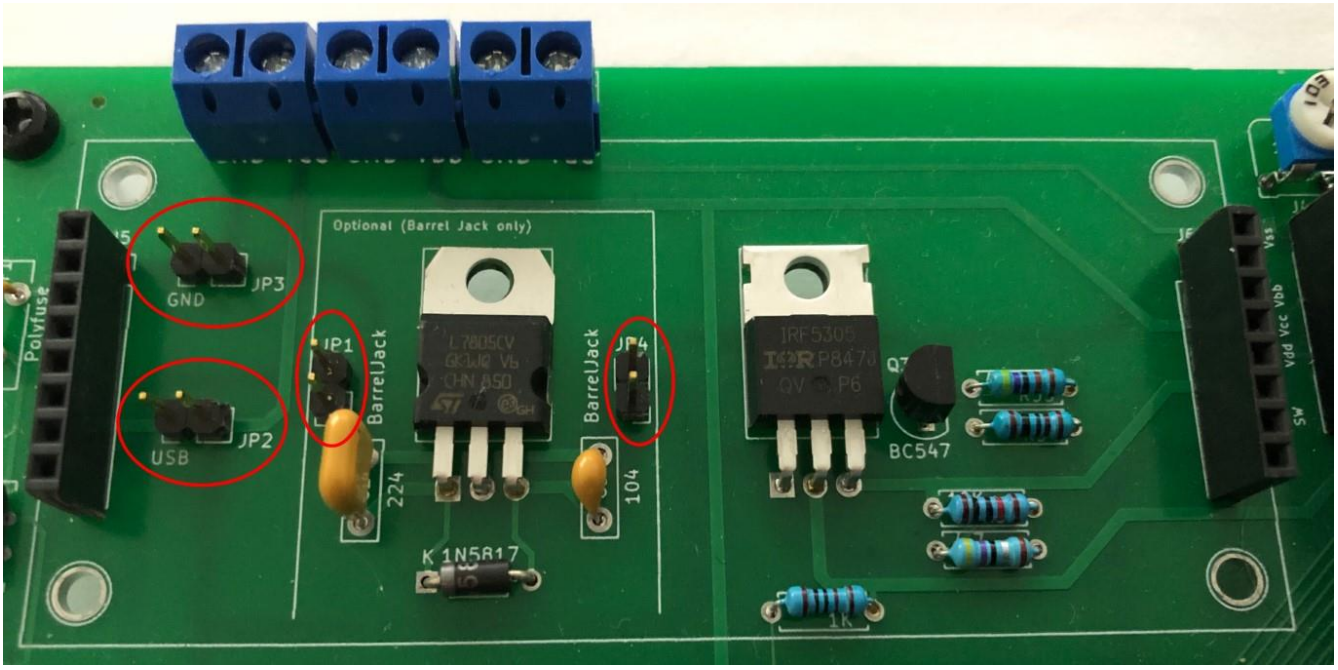


Figure 3.4: All jumpers open

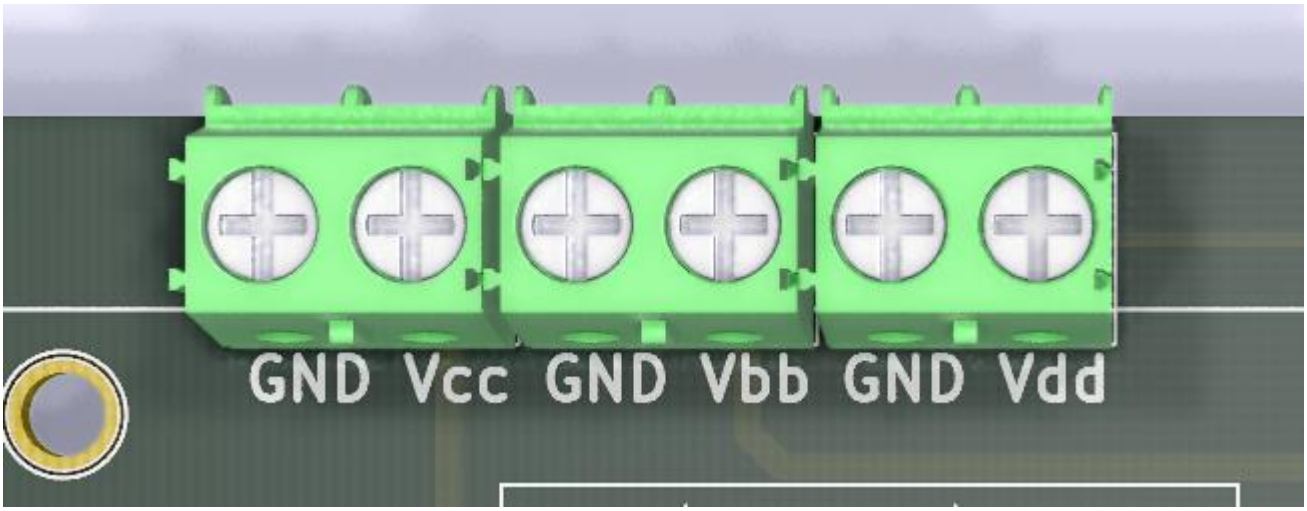


Figure 3.5: Assignment of the screw terminals (Vcc = 5V, Vbb = -5V, Vdd = 12V)

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.

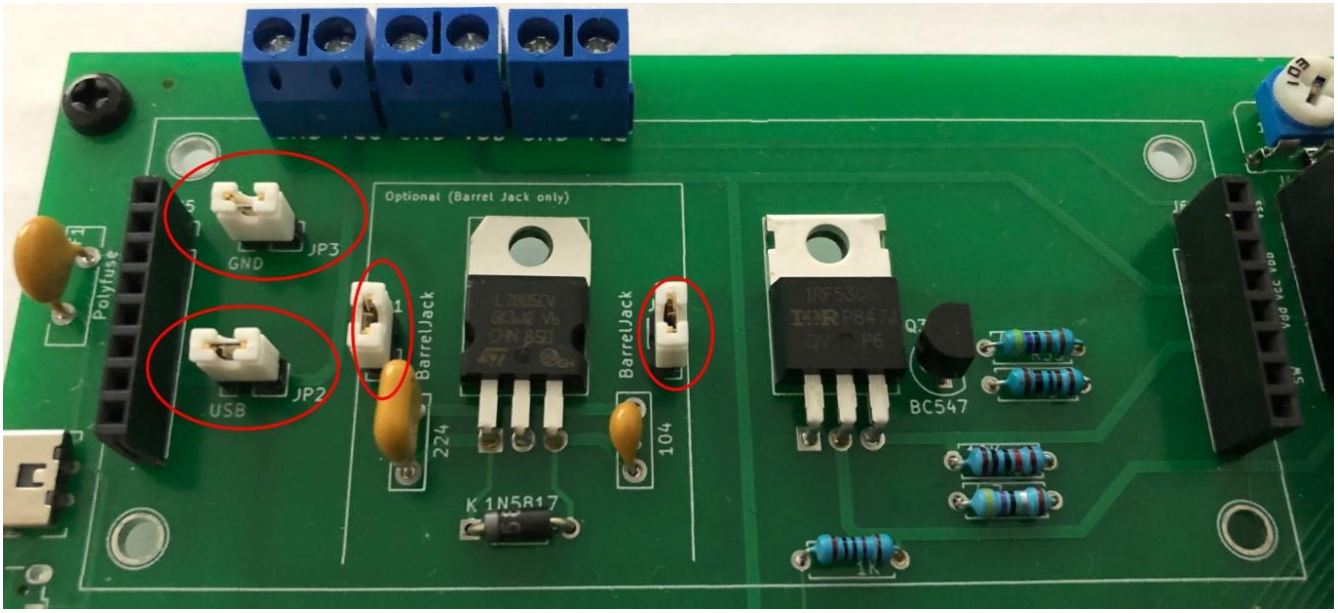


Figure 3.6: All jumpers closed

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

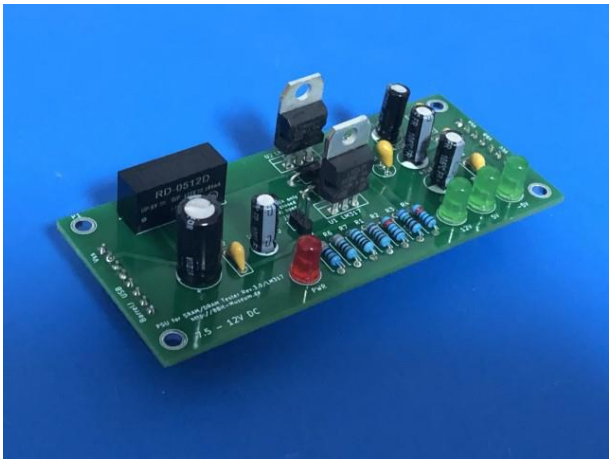
This option uses a RECOM DC/DC controller. No heat is generated even after long use and the output voltage is properly regulated. This power supply is the standard power supply for the chip tester.

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)

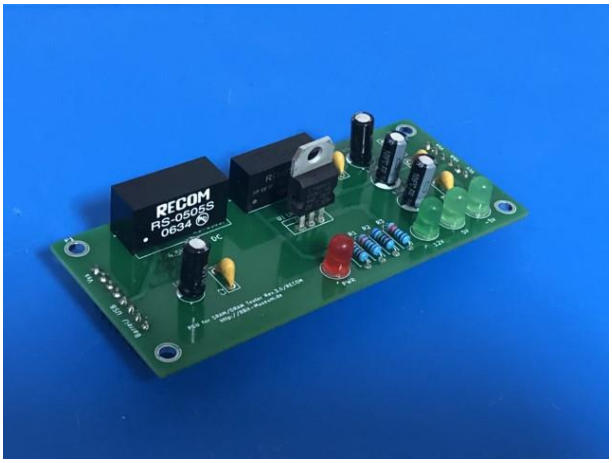
Like option #1, the second option uses a RECOM controller, but instead of the fixed voltage controller an LM317 for the 5V supply voltage is used. By setting a jumper, the supply voltage can be raised from 5V up to 5.3V. This may be necessary for special tests, e.g.

- 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 14 there are further instructions for some programming devices. Suitable programmers are listed in sec. 13.1.4 and sec. 13.1.5!

Please check the warning notices in section 10 if programming is done with the DC/DC converter plugged in (only “Alternative #2” with Recom module).

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. 13.1.4 and sec. 13.1.5!

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.

4.1 Programming the Fuses

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=ijMVng-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.

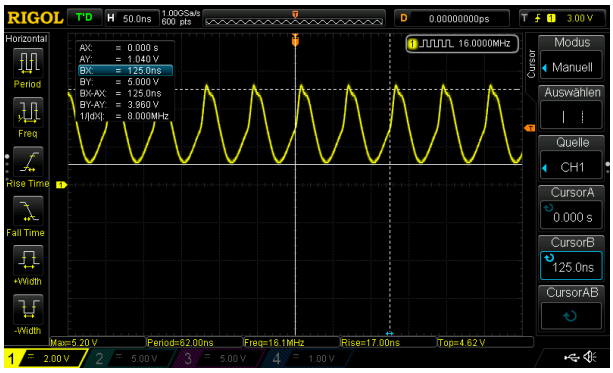
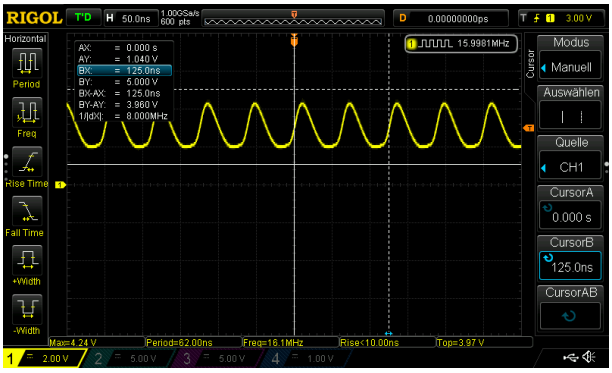


Figure 4.2: a “good“ crystal



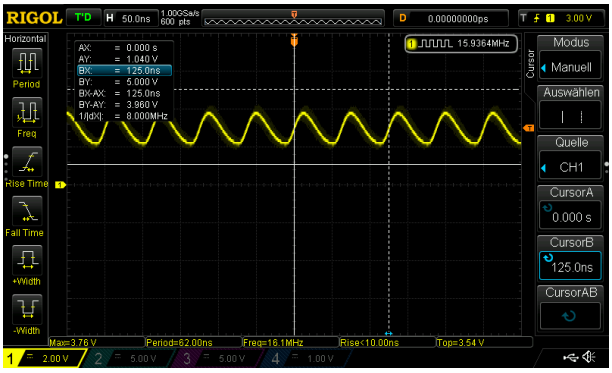
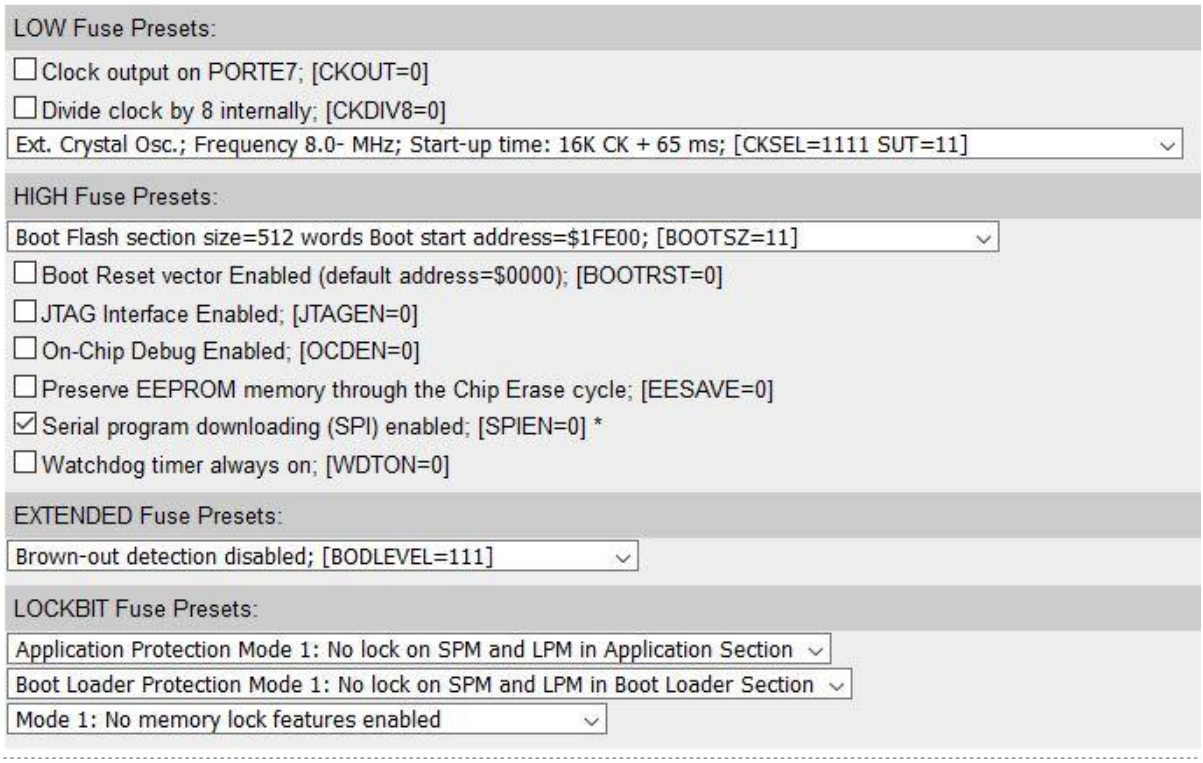


Figure 4.3: two “not so good“ crystals

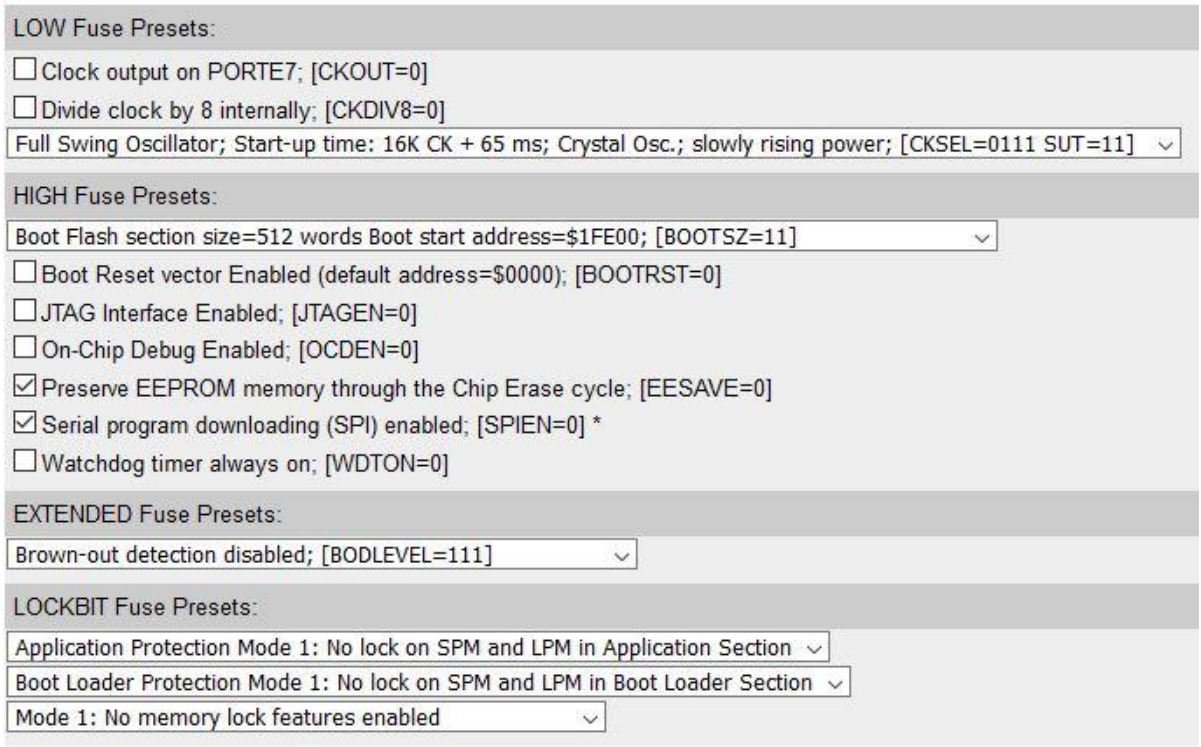
In the pictures above, three different crystals were used in the same oscillator circuit.

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



4.2 Programming the Firmware

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.13.1).

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 13.

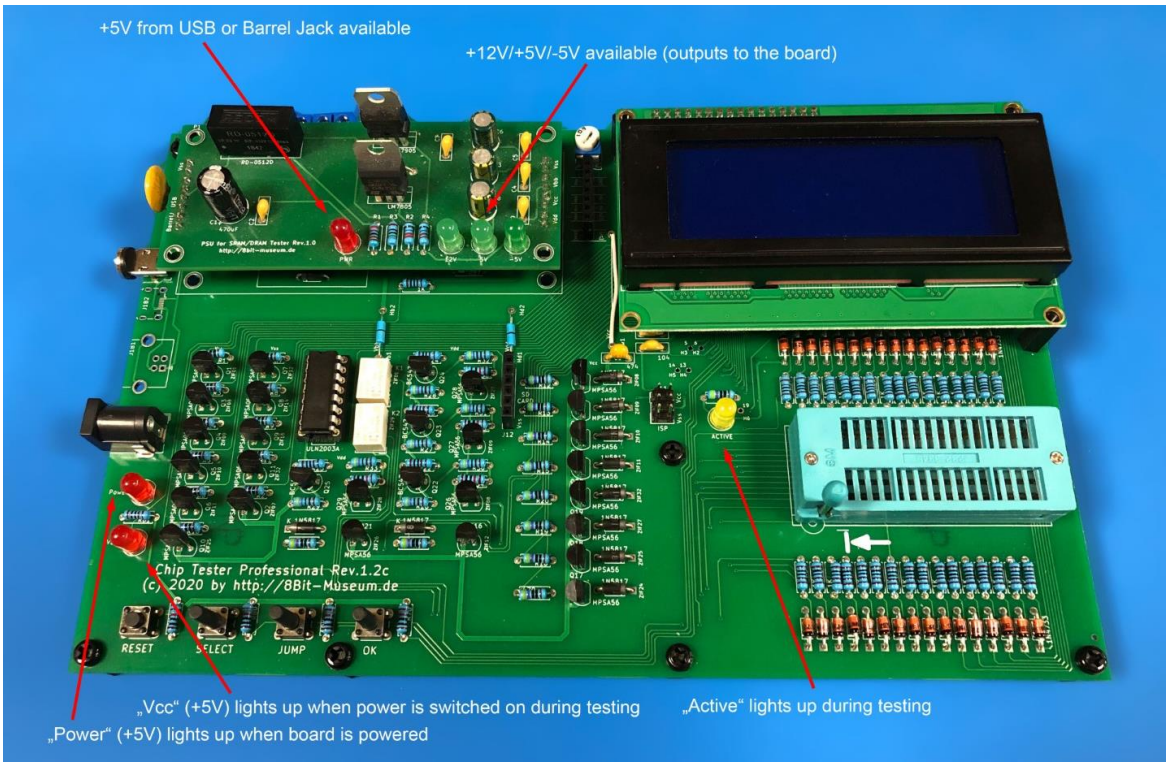
Is the display lit, but no text appears?

- 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 13.



"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 Testing Chips

6.1 Key assignment

The buttons are partly assigned multiple times.

main menu / sub menu:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				Previous chip
	X			Next chip
X	X			Jump to exit[1]
			X	Jump to main menu[2] or enter submenu
		X		Test selected chip
		keep pressed		Alternative mode in some cases

TTL/CMOS sub menu:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				Previous chip
keep pressed				Fast backward
	X			Next chip
	keep pressed			Fast forward
X	X			Jump to exit / End Loop-Mode
			X	Exit menu / End Loop-Mode
		X		Test selected chip

[1] in submenu only
[2] in submenu only

Search menu:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				Next chip (same prefix)
	X			Next character
X	X			Back to start
			X	Back to start
		X		Jump to test entry

Config menu:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				Previous entry
	X			Next entry
X	X			Jump to exit
			X	Jump to exit
		X		Select

PAL menu:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				Move cursor right
	X			Change within the group [I] = Input / [O] = Output / [-] = Ignore or [G] = GND / [V] = Vss
X	X			Change group [IO-] and [GV]
			X	Change group [IO-] and [GV]
		X		Start

TIP:

Instead of pressing the (long) buttons, it is easier to tilt them slightly to the side.
In particular, two buttons can easily be triggered at the same time.

6.2 The menu

The main menu has grown quite large due to continued development. For this reason, the chips and/or functions have been arranged into a hierarchical menu so that selection of items is easier. In the main menu you can change between:

Main Menu
SRAMs common
SRAMs uncommon
SRAMs cypress
DRAMs common
DRAMs uncommon
FIFO
EPROM
PROM/ROM 1
PROM/ROM 2
LOGIC
CUSTOM
PROGRAMMING
SERVICE

Figure 6.1: Main menu

The menu only specifies a "category" which can be entered to access other chips in that category. The compatibility list shows which setting is specifically correct for the exact chip you want to test.

Menu (long form)	Menu (short form)
SRAM (common)	
2102 - 1k x 1	2102
2115 - 1k x 1	2115
2114 - 1k x 4	2114
6550 - 1k x 4	6550
2142 - 1k x 4	2142
4118A - 1k x 8	4118A
2147 - 4k x 1	2147
xx16 - 2k x 8	xx16
xx64 - 8k x 8	xx64
xx256 - 32k x 8	xx256
xx512 - 64k x 8	xx512
xx1024 - 128k x 8	xx1024

Menu (long form)	Menu (short form)
SRAM (uncommon)	
3101 - 16 x 4	3101
74219 - 16 x 4	74219
4505 - 64 x 1	4505
74910 - 64 x 4	74910
93419 - 64 x 9	93419
6810 - 128 x 8	6810
3106 - 256 x 1	3106
93410 – 256 x 1	93410
2101 - 256 x 4	2101
2111 - 256 x 4	2111
2112 - 256 x 4	2112
74920 - 256 x 4	74920
74921 - 256 x 4	74921
6561 - 256 x 4	6561
81C50 - 256 x 8	81C50
81C51 - 256 x 8	81C51
81C54 - 512 x 8	81C54
82S208 - 256 x 8	82S208
82S210 - 256 x 9	82S210
82S212 - 256 x 9	82S212
8185 - 1k x 8	8185
74170/74670 – 4 x 4	74170/670
TC4039 – 4 x 8	4039
7481/7484 - 16 x 1	7481/84

Menu (long form)	Menu (short form)
SRAM (cypress)	
x150 - 1k x 4	x150
x167- 16k x 1	x167
x168 - 4k x 4	x168
x1681 - 4k x 4	x1681
x170 - 4k x 4	x170
x187 - 64k x 1	x187
x188 - 16k x 4	x188
x198 - 16k x 4	x198
x198A - 16k x 4	x198A
x1981 - 16k x 4	x1981
x423 - 256 x 4	x423
x1026 - 256k x 4	x1026
x1257 - 256k x 1	x1257
x1258 - 64k x 4	x1258
x1281 - 64k x 4	x1281
x1298 - 64k x 4	x1298
x1299 - 64k x 4	x1299
xx257 – 32k x 8	xx257

Menu (long form)	Menu (short form)
DRAM (common)	
2104 - 4k x 1	2104
2107 - 4k x 1	2107
4116 - 16k x 1	4116
2118 - 16k x 1	2118
4416 - 16k x 4	4416
4164 - 64k x 1	4164
4464 - 64k x 4	4464
41256 - 256k x 1	41256
44256 - 256k x 4	44256
41024 - 1024k x 1	41024

Menu (long form)	Menu (short form)
DRAM (uncommon)	
xx32L - 32k x 1	xx32L
xx32H - 32k x 1	xx32H
2108L - 8k x 1	2108L
2108H - 8k x 1	2108H
4108-x0 - 8k x 1	4108-x0
4108-x1 - 8k x 1	4108-x1
5298A - 8k x 1	5298A
5298B - 8k x 1	5298B
4408NLT – 8k x 4	4408NLT
4408NLB – 8k x 4	4408NLB
4132 M1 - 32k x 1	4132 M1
4132 M2 - 32k x 1	4132 M2
4332 M1 - 32k x 1	4332 M1
4332 M2 - 32k x 1	4332 M2
41128 TOP - 64k x 1	41128 T
41128 BOT - 64k x 1	41128 B
SIMM30 - 256k x 8	SIMM
SIMM30-P - 256k x 1	SIMM P.
SIMM30 - 1024k x 8	SIMM
SIMM30-P - 1024k x 1	SIMM P.
ZIP20 - 64k x 4	ZIP64
ZIP16 - 256k x 1	ZIP256
ZIP20 - 256k x 4	ZIP256
ZIP20 - 1024k x 1	ZIP1024
ZIP20 - 1024k x 4	ZIP1024
4008 – 1k x 1	4008

Menu (long form)	Menu (short form)
FIFO RAMs	
40105 (16 x 4)	40105
74222/27 (16 x 4)	74222/27
74224/28/32 (16 x 4)	74224/28
74225 (16 x 5)	74225
74229/33 (16 x 5)	74229/33
74234/36 (64 x 4)	74234/36
74235 (64 x 5)	74256
747404 (64 x 5)	747404
7200 (256 x 9)	7200
7201 (512 x 9)	7201
7202 (1024 x 9)	7202
7203 (2048 x 9)	7203
7204 (4096 x 9)	7204
7200 (256 x 9) PLCC32	7200 PLCC
7201 (512 x 9) PLCC32	7201 PLCC
7202 (1024 x 9) PLCC32	7202 PLCC
7203 (2048 x 9) PLCC32	7203 PLCC
7204 (4096 x 9) PLCC32	7204 PLCC

Menu (long form)	Menu (short form)
EPROMs	
2716 - 2k x 8	2716
2732 - 4k x 8	2732
2764 - 8k x 8	2764
27128 - 16k x 8	27128
27256 - 32k x 8	27256
27512 - 64k x 8	27512
271001 - 128k x 8	271001
272001 - 256k x 8	272001
274001 - 512k x 8	274001
1702 - 256 x 8	1702
2704 - 512 x 8	2704
2708 - 1k x 8	2708
TMS2716 – 2k x 8	TMS2716
TMS2564 – 8k x 8	TMS2564
VCS/2600 Cart	VCS/2600
ES 2764-27128	2764+
ES 27256-27512	27256+

Menu (long form)	Menu (short form)
ROMs / PROMs 1	
2308 - 1k x 8	2308
2316A - 2k x 8	2316A
2316B - 2k x 8	2316B
2332 - 4k x 8	2332
2364(24) - 8k x 8	2364(24)
2364(28) - 8k x 8	2364(28)
23128 - 16k x 8	23128
23256 - 32k x 8	23256
23512 - 64k x 8	23512
231000 - 128k x 8	231000
232000 - 256k x 8	232000
234000 - 512k x 8	234000
6540 - 2k x 8	6540
RO-3-2513 – 64x8x5	2513
6670 – 128x7x5	6670
2804 – 512 x 8 EEPROM	2804

Menu (long form)	Menu (short form)
ROMs / PROMs 2	
7488/188/288 - 32x8	7488
74186 - 64x8	74186
4187/287/387 - 256x4	74187
74S271/470 - 256x8	74S271
74S270/570 - 512x4	74S270
74S472/473 - 512x8	74S472
74S474/475 - 512x8	74S474
74S476/572 – 1k x4	74S476
74S478/479 – 1k x8	74S478
82S184/185 – 2k x4	82S184
82S190/191 – 2k x8	82S190
82S195 – 4k x4	82S195
82S321 – 4k x8	82S321
82S641 – 8k x8	82S641

Menu (long form)	Menu (short form)
LOGIC	
7400+ Logic Tests	
74300+ Logic Tests	
74600+ Logic Tests	
40xx Logic Tests	
Misc Logic Tests 1	
Misc Logic Tests 2	
USSR TTL Logic	
USSR CMOS Logic	
Misc Tests	
PAL 20pin	PAL
PAL 24pin	PAL

Menu (long form)	Menu (short form)
CUSTOM SRAMs / DRAMs	
12 x SRAMs	
3 x DRAMs	
3 x ROMs	

Menu (long form)	Menu (short form)
PROGRAMMING	
2708	
2716 / 2516	
TMS2716	
2532	
2732	
2564	

Menu (long form)	Menu (short form)
SERVICE	
Search	
Info	
Enter Config	
Selftest	

6.3 Searching a Chip

The menu shows only one chip representing several chips of the same type. The correct menu entry for a special chip can be obtained from the comparison list or via the search function. The database currently contains around 750 chips and, once selected, allows you to jump directly to the correct menu entry.

Example: Finding the chip HM50464

Select the first character by pressing SELECT.	
If the first character ("H") is correct, press JUMP once.	
The cursor moves to the second character. Now select the second character with SELECT.	
If the second character ("M") is correct, press JUMP once.	
The cursor moves to the third character. Now select the third character with SELECT.	
The chip "HM50464" is displayed. Select the chip with OK, the correct menu entry is then displayed.	
The test process can now be started with OK.	

The database is not complete.
If a type is not listed, it usually helps to check the datasheet.

iPhone / iPad users:

Do not open the file from an email (Javascript will not be executed in this case), but save it on the device (or in a Dropbox, Google Drive etc.) and then open it.

An HTML file is also provided, which should help to find the correct test setting. The database is much larger than the one built into the tester.

After opening the file, the following website is displayed:

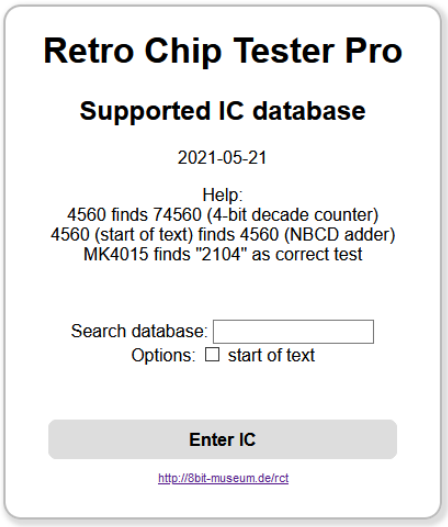


Figure 6.2: Start screen

After entering an IC designation, the correct test setting is displayed after "Select:". With an “MK4015” this is the test “2104”.

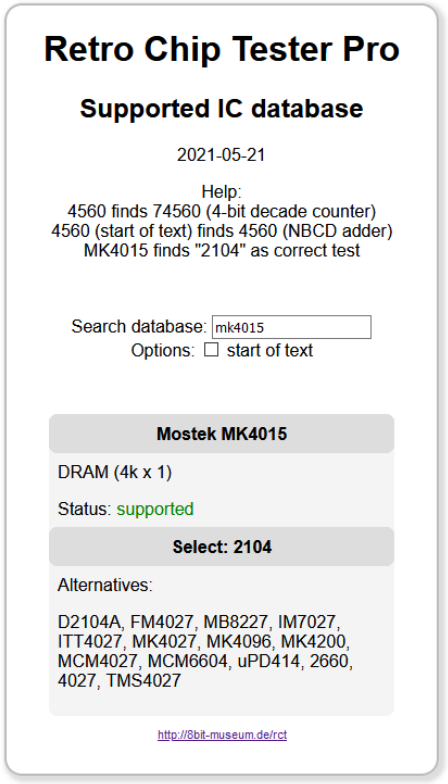


Figure 6.3: Looking up MK4015

Attention: The "alternatives" can all be tested with this test, but are not necessarily suitable as replacements in the real system.

6.4 Testing of SRAMs and DRAMs

6.4.1 Inserting chips

All chips are inserted left-justified. There are a few exceptions, please note the instructions for the TTL chips.

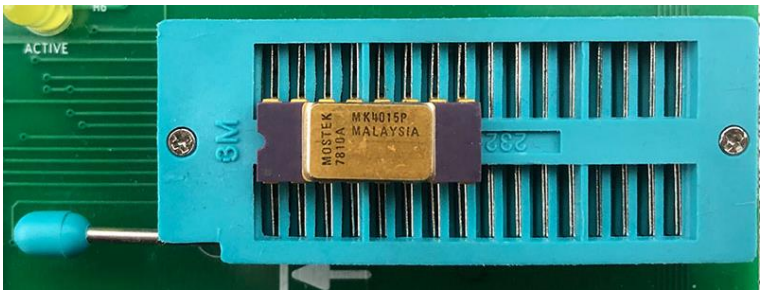


Figure 6.4: Correct position of the chip

If a "Pos x" is displayed after the type designation, the chip must be inserted at "Position x" in the socket. The chips are normally used left aligned (corresponds to "Pos 1"). Since some chips do not have the power supply pins in the usual locations (e.g. 7475, 7490, 7492, 7493), these chips must be positioned in the ZIF socket at an offset position. For example "Pos 4" means that three rows must remain free on the left side of the chip and pin#1 of the chip to be tested must be set at ZIF socket position 4.

Figure 6.5: Chip inserted at position 4 (7490)

Before starting a test the Tester checks whether an SRAM or DRAM is inserted. In rare cases this check can fail. If no chip is detected the message "`Insert <IC type>`" is displayed. In this case the check can be switched off in the configuration.

6.4.2 Selecting the current chip

The menu allows the selection of the chip to be tested. Due to the limited space, only one "main type" is displayed. The compatibility list for SRAMs / DRAMs provides information as to which menu item and which switch setting is correct in individual cases (example: an Am2148 is identical to a 2114 (1k x 4), i.e. menu selection "2114".

Figure 6.6: Example: Selection of a “2114” SRAM resp. “4116” DRAM

If the type cannot be read directly from the label, the memory module can be searched for using the search function. Alternatively, a comparison list for SRAMs, DRAMs and ROMs can be downloaded from the website, which lists the correct setting for many memory modules.

6.4.3 Output of the test result

If a test is successful, `passed` is displayed. If a test fails, the defective memory cell (e.g. `A:13F4`) and the expected and read value are displayed (e.g. `5>F` = value `1010` expected, `1111` read).

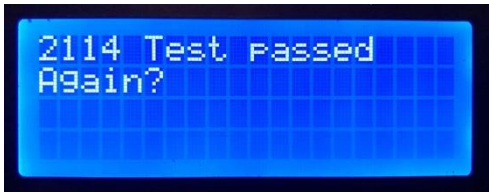


Figure 6.7: Successful test of a 2114 SRAM

Figure 6.8: Failed tests of a 6550 (expected 0x0000, read 0x0010) at address 0x02C0 and 4116 (expected 1, read 0) at address 0x3840

Important:

The Chip Tester detects defective memory cells very reliably. If the same memory module is tested several times and the error always occurs at the same memory cell, then it is very likely that the memory module is defective.

Unfortunately, there are also errors that are not so clear:

When a memory module is tested several times and the error occurs at different memory cells, then it is entirely possible that the memory is too slow for the test (i.e. the access time has increased over the years), that the voltage level at the output no longer meets the specification, or the inputs require a higher current that the tester cannot provide (max. 8mA). Section 11 addresses other possible problems.

These memory modules can still function in a 80's home computer. Increased access times is usually not required for very old electronics. Even with signal levels that no longer correspond exactly to the specification (these cannot be adhered to precisely, see also sec. 11), a computer can still function properly (at least for a certain period of time). However, the tester will report an error if the components are slow or the signal level is insufficient.

It is not tested whether a component still works in a particular computer, but whether the specified limits are respected.

The test duration is approximately:

8k x 8 SRAM (6264)	
Pattern Tests (0-Test, 1-Test, etc.)	approx. 5s / Test
Random Test	approx. 7s
March Y	approx. 20s
March U	approx. 40s

128k x 8 SRAM (62512)	
Pattern Tests (0-Test, 1-Test, etc.)	approx. 30s / Test
Random Test	approx. 55s
March Y	approx. 2:00min
March U	approx. 3:30min

16k x 1 DRAM (4116)	
Pattern Tests (0-Test, 1-Test, etc.)	approx. 1,5s / Test
Random Test	approx. 4,5s
March Y	approx. 8s
March U	approx. 13s

1024k x 8 SIMM/SIPP	
Pattern Tests (0-Test, 1-Test, etc.)	approx. 1:30min / Test
Random Test	approx. 5:00min
March Y	approx. 11:30min
March U	approx. 21:00min

6.4.4 Saving the test result

The content read during a test can be saved on an SD memory card. To enable this

- a card adapter with a suitable memory card must be connected,
- **SD Card Support** must be switched on in the configuration („**SD Card: 1**“),
- the **Dump Function** must be activated in the configuration („**Alt. mode: 1**“).

If the memory test is started with a **long press on the OK button**, the file `<chip>.<xxx>` is generated, where "`xxx`" is a consecutive number. This means that several identical chip types can be tested one after the other.

When the tests are written to the SD card, in the event of an error the test process is not aborted immediately. All tests are carried out and only at the end the first error found is displayed.

The file contains the entire chip contents from all tests in a row. With a 6550 SRAM (1 kByte x 4) the file is structured as follows:

0x0000	0-Test (0000)
0x0400	5-Test (0101)
0x0800	A-Test (1010)
0x0C00	F-Test (1111)
0x1000	count+ (0,1,2,3,4…)
0x1400	count- (f, e, d, c, …)
0x1800	Random(16) read
0x1C00	Random(16) written

With a 1-bit memory, a few tests are omitted, so we get the following layout: 0-Test (0), 1-Test (1), 01-Test (0101), 10-Test (1010), Random(2).

With an 8-bit memory, 0x55, 0xAA, 0xFF are used in the pattern test and Random(256).

Examples of a correct 6550 SRAM:

```
00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

0-Test (0000)

```
00000400 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
00000410 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
00000420 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
00000430 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
```

5-Test (0101)

```
00000800 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
00000810 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
00000820 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
00000830 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
```

A-Test (1010)

```
00000c00 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f
00000c10 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f
00000c20 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f
00000c30 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f
```

F-Test (1111)

```
00001000 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00001010 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00001020 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00001030 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
```

count+

```
00001400 0f 0e 0d 0c 0b 0a 09 08 07 06 05 04 03 02 01 00
00001410 0f 0e 0d 0c 0b 0a 09 08 07 06 05 04 03 02 01 00
00001420 0f 0e 0d 0c 0b 0a 09 08 07 06 05 04 03 02 01 00
00001430 0f 0e 0d 0c 0b 0a 09 08 07 06 05 04 03 02 01 00
```

count-

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00001800	0c	09	05	08	05	0d	0a	0a	05	01	0d	0b	0c	0a	06	05
00001810	0c	03	01	02	05	0e	00	0d	0f	0b	0e	01	03	0c	0d	0e
00001820	0b	06	06	0f	09	05	0b	04	03	08	0c	0d	08	0d	06	06
00001830	0c	0a	0a	07	02	01	05	00	0b	04	06	08	00	02	0a	0e
00001840	0d	00	0c	08	0c	02	0e	0d	0c	00	00	0e	0c	0f	08	03
00001850	0d	0e	03	0e	09	02	0e	05	06	04	03	05	01	09	00	06
00001860	05	0d	0b	0d	07	03	00	02	07	03	00	0a	05	08	06	0a

Random(16) read

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00001c00	0c	09	05	08	05	0d	0a	0a	05	01	0d	0b	0c	0a	06	05
00001c10	0c	03	01	02	05	0e	00	0d	0f	0b	0e	01	03	0c	0d	0e
00001c20	0b	06	06	0f	09	05	0b	04	03	08	0c	0d	08	0d	06	06
00001c30	0c	0a	0a	07	02	01	05	00	0b	04	06	08	00	02	0a	0e
00001c40	0d	00	0c	08	0c	02	0e	0d	0c	00	00	0e	0c	0f	08	03
00001c50	0d	0e	03	0e	09	02	0e	05	06	04	03	05	01	09	00	06
00001c60	05	0d	0b	0d	07	03	00	02	07	03	00	0a	05	08	06	0a

Random(16) written

Examples of defective 6550 SRAMs:

```
00000800 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0a 0a
00000810 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
00000820 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
00000830 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
```

Error in A-Test

```
00001000 01 01 03 03 05 05 07 07 09 09 0b 0b 0d 0d 0f 0f
00001010 01 01 03 03 05 05 07 07 09 09 0b 0b 0d 0d 0f 0f
00001020 01 01 03 03 05 05 07 07 09 09 0b 0b 0d 0d 0f 0f
00001030 01 01 03 03 05 05 07 07 09 09 0b 0b 0d 0d 0f 0f
00001040 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00001050 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
```

Error in count+ Test

```
000000f0 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b
00000100 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b
00000110 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b
00000120 0b 0b 0b 0b 0b 09 0b 0b 0b 0b 0b 0b 0b 09 09 09
00000130 09 09 09 09 09 09 09 09 0b 09 01 09 09 09 09 09
00000140 01 01 09 09 01 01 01 01 01 01 01 09 01 01 01 01
00000150 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000160 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000170 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000180 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
00000190 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
000001a0 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
000001b0 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
000001c0 01 01 01 01 01 01 01 01 01 01 01 01 01 00 01 01
000001d0 01 01 00 00 00 00 00 00 01 01 00 00 00 00 00 01
000001e0 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Error in 0-Test

Examples of a defective 2112A SRAM:

Random data was written to memory, but 16 consecutive bytes are identical. This indicates that the address line A4 is damaged internally, i.e. is always LOW. The previous tests could not detect this error (not even the count test, since the memory has a width of 4 bits and the count pattern is repeated after 16 bytes).

```
00000700 08 0a 0c 0a 02 01 05 09 07 0d 0e 02 0b 08 0c 02
00000710 01 07 09 0e 01 0b 08 09 00 08 08 00 0f 08 00 0d
00000720 05 02 09 07 05 05 01 0e 0e 03 05 05 00 09 0b 06
00000730 04 0f 0a 07 09 01 07 0e 04 02 01 0e 0e 05 06 05
00000740 07 00 05 03 00 0e 07 05 05 0e 04 0a 06 0c 03 03
00000750 06 0f 0d 0f 02 0b 08 06 05 0e 05 05 05 08 0a 06
00000760 0c 00 0b 01 0c 0f 0d 0b 0a 02 06 05 06 08 02 01
00000770 08 02 01 02 0f 04 01 0b 00 09 05 0a 01 0e 0d 03
00000780 0a 0b 0f 0b 09 05 09 08 0f 0e 0f 0a 08 04 0e 05
00000790 09 07 02 09 0b 08 05 07 08 0d 00 0d 0a 0b 01 0f
000007a0 06 0f 07 03 04 0a 09 0c 02 04 00 0c 03 08 00 06
000007b0 02 0c 07 0b 0b 06 03 07 07 00 0d 0b 08 0b 0d 0c
```

Written data in the Random(16) test

```
00000600 01 07 09 0e 01 0b 08 09 00 08 08 00 0f 08 00 0d
00000610 01 07 09 0e 01 0b 08 09 00 08 08 00 0f 08 00 0d
00000620 04 0f 0a 07 09 01 07 0e 04 02 01 0e 0e 05 06 05
00000630 04 0f 0a 07 09 01 07 0e 04 02 01 0e 0e 05 06 05
00000640 06 0f 0d 0f 02 0b 08 06 05 0e 05 05 05 08 0a 06
00000650 06 0f 0d 0f 02 0b 08 06 05 0e 05 05 05 08 0a 06
00000660 08 02 01 02 0f 04 01 0b 00 09 05 0a 01 0e 0d 03
00000670 08 02 01 02 0f 04 01 0b 00 09 05 0a 01 0e 0d 03
00000680 09 07 02 09 0b 08 05 07 08 0d 00 0d 0a 0b 01 0f
00000690 09 07 02 09 0b 08 05 07 08 0d 00 0d 0a 0b 01 0f
000006a0 02 0c 07 0b 0b 06 03 07 07 00 0d 0b 08 0b 0d 0c
000006b0 02 0c 07 0b 0b 06 03 07 07 00 0d 0b 08 0b 0d 0c
```

Read data in the Random(16) test

Hints:

For functions that require selection by pressing the OK button for a long time, those functions cannot be accessed as long as the dump function is activated in the configuration.

Using the dump function changes the timing when reading out the chip (it is read out more slowly due to the necessary access to the SD memory card). As a result, the result can differ from the result without a dump function.

In the case of DRAMs, the result can be falsified if the access to the SD memory card is too slow and a refresh of the DRAM is no longer guaranteed.

If the dump function is activated and March-Y/U is also activated in the configuration, the March-Y/U test will not be done.

6.4.5 Identifying Page Mode, Nibble Mode and Static Column Mode in DRAMs

If the identification of the DRAM features is activated in the configuration, the tester tries to recognize whether the memory module supports the Page Mode (PM), Nibble Mode (NM) and, if applicable, Static Column Mode (SC).

Figure 6.9: DRAM supporting the Page Mode

6.5 Special memory modules

6.5.1 Testing SIMM and SIPP modules

It is possible to test 256 kByte and 1 Mbyte SIMM and SIPP modules.

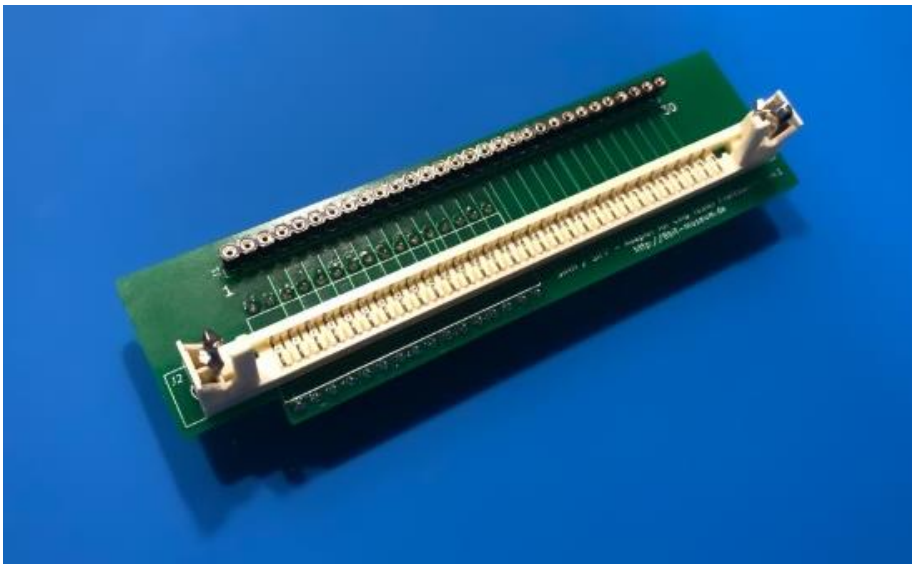


Figure 6.10: Adapter for SIMM und SIPP

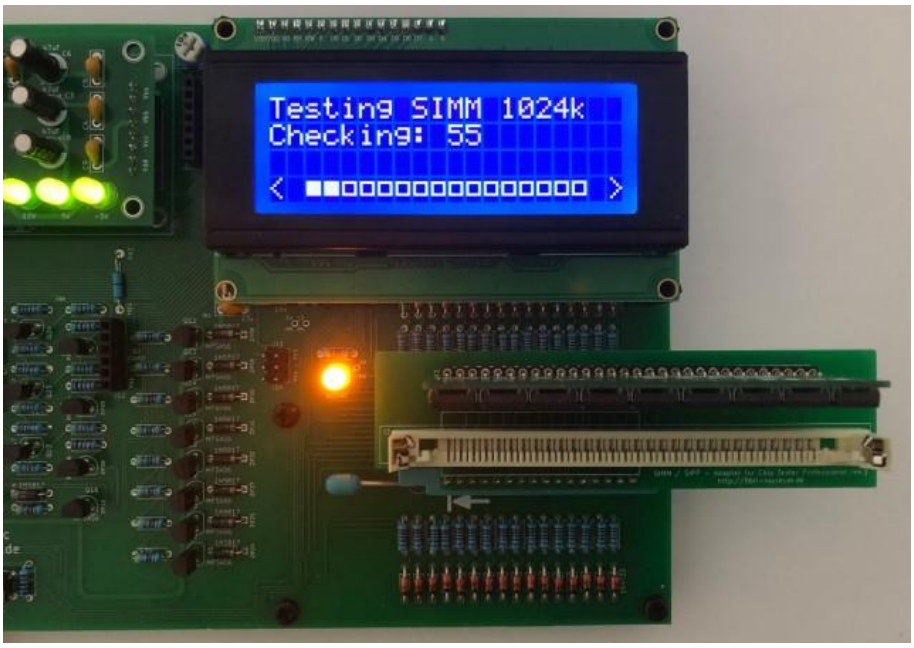


Figure 6.11: Adapter in socket

The memory is plugged into the adapter (either as SIMM or SIPP) and the adapter is inserted left-justified in the socket. There are two tests for each memory size, e.g. "SIMM 1024k x 8" and "SIMM-P 1024k x 1". The reason for this is that these modules were produced as 8-bit and 9-bit modules (i.e. with a parity bit). “SIMM NNN x 8” tests the actual module, “SIMM-P NNN x 1” tests the parity bit that may be present.

The modules with parity bits can be recognized by their three or nine chips.

Without an adapter, the modules can be connected to the tester in other suitable ways.

Figure 6.12: SIPP adapter with breadboard

The socket is assigned as follows:

GND	52	1	32	21	Vcc
A0	51	2	31	22	D0
A1	50	3	30	23	D1
A2	49	4	29	24	D2
A3	48	5	28	25	D3
A4	47	6	27	26	D4
A5	46	7	26	27	D5
A6	45	8	25	28	D6
A7	44	9	24	29	D7
A8	43	10	23	30	/WR
A9	42	11	22	31	/RAS
A10	41	12	21	32	/CAS
A11	40	13	20	33	/CASP
	39	14	19	34	DP (in)
	38	15	18	35	QP (out)
	37	16	17	36	

Figure 6.13: Socket pinout for SIMM/SIPP test

A 30-pin SIMM or SIPP module is assigned as follows:

Pin	Name	Signal
1	VCC	+5 VDC
2	/CAS	Column Address Strobe
3	DQ0	Data 0
4	A0	Address 0
5	A1	Address 1
6	DQ1	Data 1
7	A2	Address 2
8	A3	Address 3
9	VSS	Ground
10	DQ2	Data 2
11	A4	Address 4
12	A5	Address 5
13	DQ3	Data 3
14	A6	Address 6
15	A7	Address 7
16	DQ4	Data 4
17	A8	Address 8
18	A9	Address 9
19	A10	Address 10
20	DQ5	Data 5
21	/WE	Write Enable
22	VSS	Ground
23	DQ6	Data 6
24	A11	Address 11
25	DQ7	Data 7
26	QP*	Data parity out
27	/RAS	Row Address Strobe
28	/CASP*	Parity Column Address Strobe
29	DP*	Data parity in
30	VCC	+5 VDC

Figure 6.14: SIMM/SIPP pinout

For 256 kByte modules, the address lines A0 to A8, the data lines D0-D7 and /CAS, /RAS and /WE (or /WR) must be connected to one another.

For 1 Mbyte modules, the address lines A0 to A9, the data lines D0-D7 and /CAS, /RAS and /WE (or /WR) must be connected to one another.

Furthermore, Vcc or 5V and GND or Vss must be connected to one another.

Only 5V modules can be tested.

6.5.2 Testing ZIP memory

It is possible to test 256 KBit (64k x 4), 1 MBit (256k x 4 or 1024k x 1) and 4 MBit (1024k x 4) ZIP RAM.

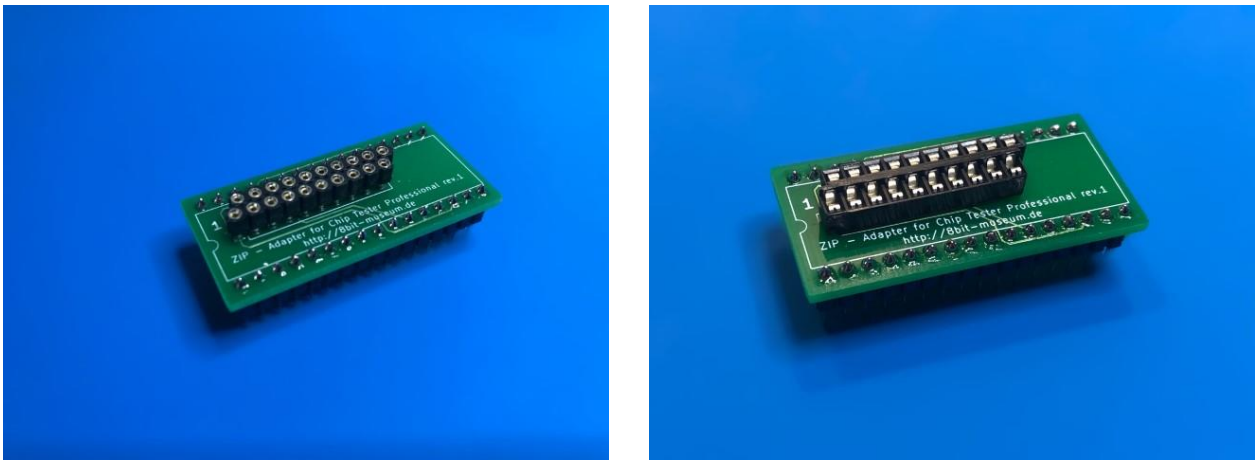


Figure 6.15: Adapter for ZIP

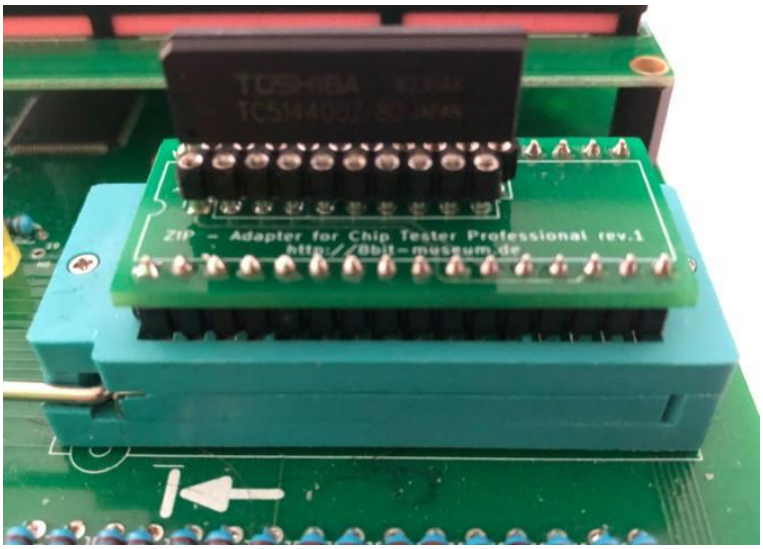


Figure 6.16: Adapter in socket

The chip is inserted into the tester in its adapter. The test ("ZIP20 XXX") is started by selecting the correct memory size.

The ZIP socket consists either of two rows of 20-position machine pin sockets or two rows of 20-position dual-wipe socket strips. The socket with dual-wipe contacts may be a better choice because the chip can be inserted and removed more easily. These can be made by cutting a standard DIP socket and the pieces are soldered together to form a row of offset holes into which the ZIP RAM can be inserted.

The socket is assigned as follows:

GND	52	1	32	21	Vcc
A0	51	2	31	22	D0
A1	50	3	30	23	D1
A2	49	4	29	24	D2
A3	48	5	28	25	
A4	47	6	27	26	
A5	46	7	26	27	
A6	45	8	25	28	D3
A7	44	9	24	29	
A8	43	10	23	30	/WR
A9	42	11	22	31	/RAS
	41	12	21	32	/CAS
	40	13	20	33	
/OE	39	14	19	34	
	38	15	18	35	
	37	16	17	36	

Din	52	1	32	21	Vcc
A0	51	2	31	22	WR
A1	50	3	30	23	/RAS
A2	49	4	29	24	Dout
A3	48	5	28	25	
A4	47	6	27	26	
A5	46	7	26	27	
A6	45	8	25	28	Vss
A7	44	9	24	29	
A8	43	10	23	30	TF
	42	11	22	31	nc
	41	12	21	32	/CAS
	40	13	20	33	
A9	39	14	19	34	
	38	15	18	35	
	37	16	17	36	

Figure 6.17: Socket pinout for ZIP20 (256k/1024k x 4) and ZIP20 (1024k x 1)

/OE	52	1	32	21	Vcc
nc	51	2	31	22	D0
A6	50	3	30	23	D1
A5	49	4	29	24	D3
A4	48	5	28	25	
A7	47	6	27	26	
A3	46	7	26	27	
A2	45	8	25	28	Vss
A1	44	9	24	29	
A0	43	10	23	30	/WR
	42	11	22	31	/RAS
	41	12	21	32	/CAS
	40	13	20	33	
D2	39	14	19	34	
	38	15	18	35	
	37	16	17	36	

GND	52	1	32	21	Vcc
A0	51	2	31	22	Din
A1	50	3	30	23	Dout
A2	49	4	29	24	
A3	48	5	28	25	
A4	47	6	27	26	
A5	46	7	26	27	
A6	45	8	25	28	
A7	44	9	24	29	
A8	43	10	23	30	/WR
	42	11	22	31	/RAS
	41	12	21	32	/CAS
	40	13	20	33	
	39	14	19	34	
	38	15	18	35	
	37	16	17	36	

Figure 6.18: Socket pinout for ZIP20 (64k x 4) and ZIP16 (256k x 1)

As a further option for testing ZIP storage, there are three ZIP-to-DIP adapters available.

Figure 6.19: ZIP-to-DIP

The ZIP-2-DIP adapter is available in three versions:

- for 256k x 1 and 64k x 1 ZIP modules,
- for 64k x 4 and 1024k x 1 ZIP modules,
- for 256k x 4 and 1024k x 4 ZIP modules.

It also allows the testing of ZIP modules, such as with the ZIP20 or ZIP16 adapter. The difference is that the standard test functions for DRAMs are used.

Due to the pin assignment of the memory in the DIP housing, there are three adapters for this.

This means that e.g. a 256k x 1 ZIP module can be tested with the ZIP-to-DIP adapter "256k x 1 / 64k x 1" with the 41256 test function.

The adapter is not suitable for inserting a ZIP module into a DIP16/18/20 socket, as the grid dimension is 15.24mm. However, this adapter should also work with other DRAM testers.

6.5.3 Testing 7481 or 7484 bipolar SRAM

It is possible to test a 7481 or 7484 bipolar SRAM (16 bit x 1). However, the Tester cannot do it alone; a special adapter is required for this.

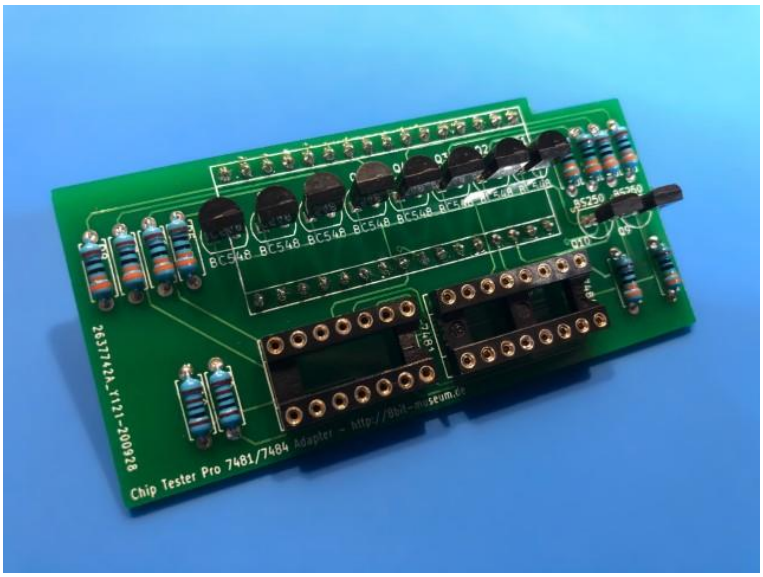


Figure 6.20: Adapter for 7481 and 7484

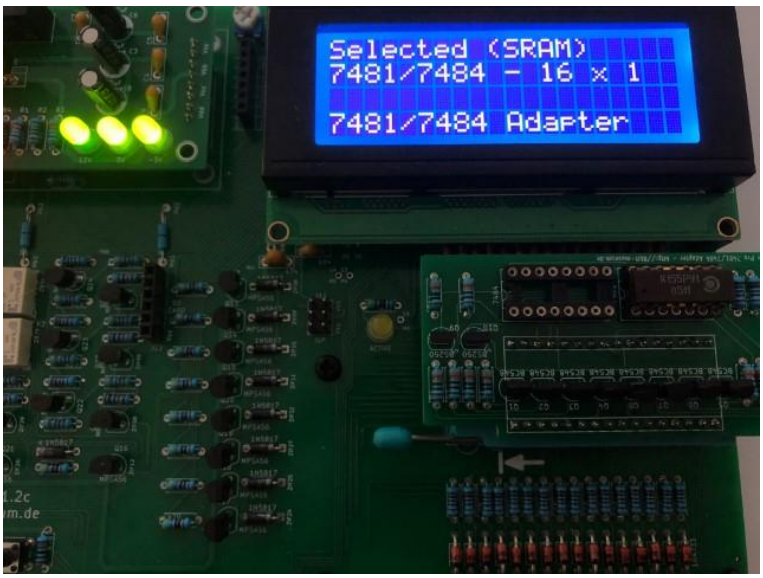


Figure 6.21: Adapter in socket

6.5.4 Testing MK4008 and MK4006 (DRAM, 1024 x 1)

The two memory chips MK4008 and MK4006 (DRAM, 1024 x 1 bit) from MOSTEK can be tested using a simple adapter. The adapter ensures that the voltage supply of -12V and +5V is injected at pin 9 and pin 10. The 0V potential is generated internally by a voltage divider.

The outputs are not TTL compatible and usually require a driver. However, the IC can be tested as described below, although success cannot always be guaranteed.

The special adapter for the MK4008 or the Breakout adapter can be used as an adapter.

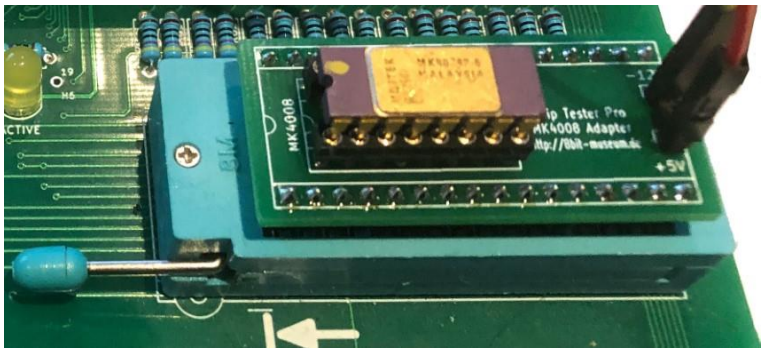


Figure 6.22: MK4008 adapter

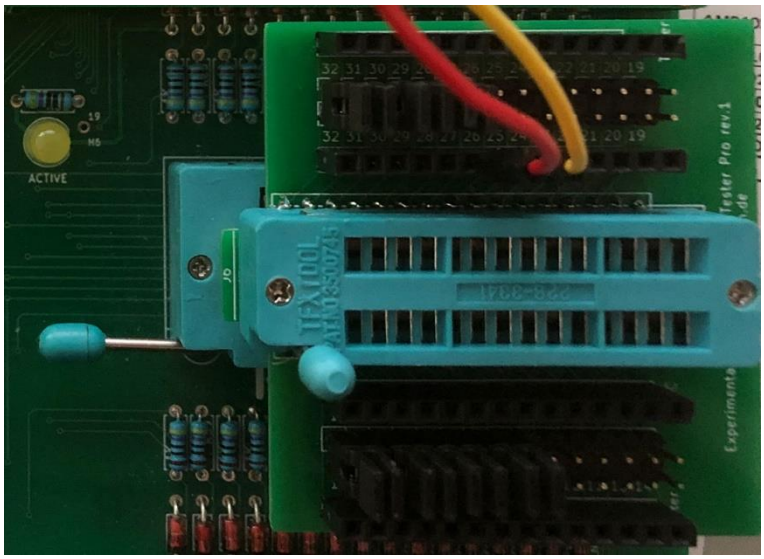


Figure 6.23: MK4008 in the Breakout-Adapter

With Adapter:

The power supply is connected directly at the 7-pin socket header. First the +5V, then the -12V.

With Breakout-Adapter:

Pins 1-8 and 11-16 (related to the 16-pin IC) are jumpered. Pin 10 (red cable) is connected to +5V („5V“ or “[5V]“ at the 7-pin socket header) and pin 11 (yellow cable) is connected to -12V (available on the DC/DC-PCB). An additional 10k ohm resistor must be attached on the right(!) side between pin 5 and pin 8 (not in the picture, Dout with 10k ohm to GND).

Due to the simple structure, however, a simple socket can in principle also be converted.

In addition, a 10k Ohm resistor is used as a pulldown. This is connected to pin 12 (Dout) of the IC in the ZIF socket and connected to GND/Vss at the 7-pin socket header.

Starting with firmware v.23, an additional 10k ohm resistor must be soldered: On the right(!) side, a 10k ohm resistor is soldered between the 5th pin and 8th pin (counting from the top, see picture).

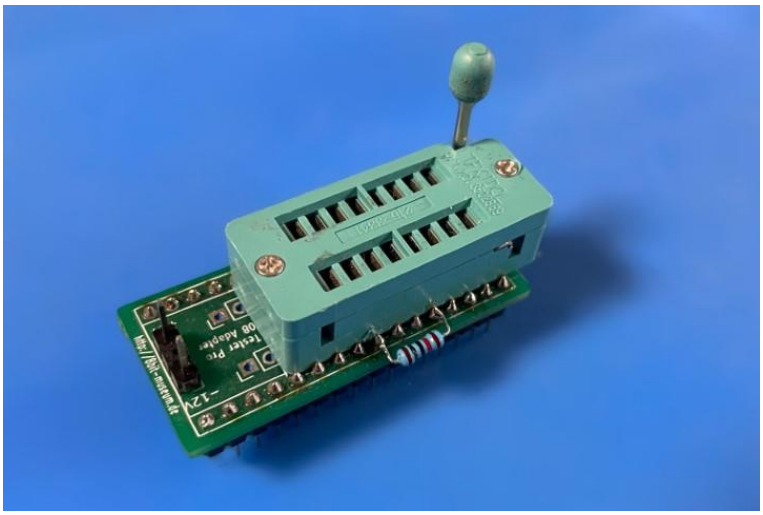


Figure 6.24: MK4008 Adapter (modifiziert)

6.5.5 Testing Intel 2107 and compatible DRAMs

The “2107 adapter” is required to test the Intel 2107 and compatible DRAMs. This can be used from firmware v.20 beta 4. The older test up to firmware v.19 did not require an adapter, but experience shows that it only worked reliably with the Intel 2107C.

The adapter may only be used with the corresponding menu item "2107 - 2107 Adapter". The adapter can be destroyed with the "old" entry from firmware v.19. Be sure to update your firmware to the latest version before using this adapter.

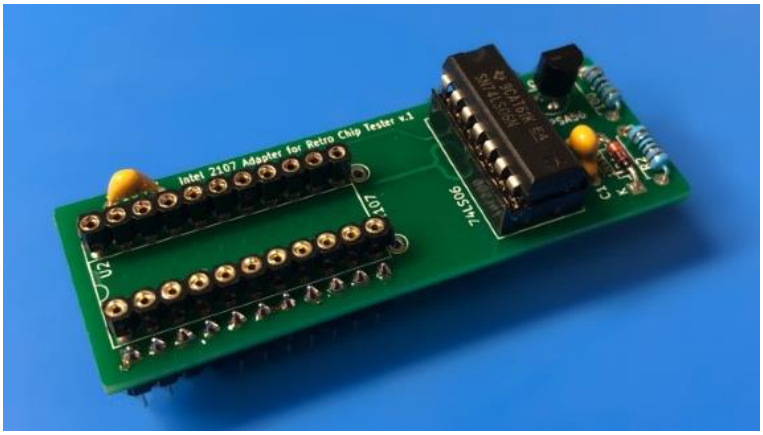


Figure 6.25: 2107 adapter

There is not much to consider when testing with the adapter. Just insert the IC to be tested into the socket on the adapter board (taking note of the position of pin#1) and place the adapter into the ZIF socket on the RCT.

The adapter converts the 5V CE signal from the RCT into a 12V CE signal for the memory chip. The adapter MUST NOT be used with the old test from firmware v.19 because 12V would be applied to the inputs of the 7406 and destroy it. However note that if a mistake is made the RCT or memory chip cannot be damaged as a result.

Technical background:

Ideally, the 2107 is controlled with the help of several ICs. The CE is for example generated with an Intel 8210 “TTL to MOS Level Shifter and High Voltage Clock Driver”, which also adjusts the signal levels at the same time. The inputs and outputs are each managed using an Intel 8212.

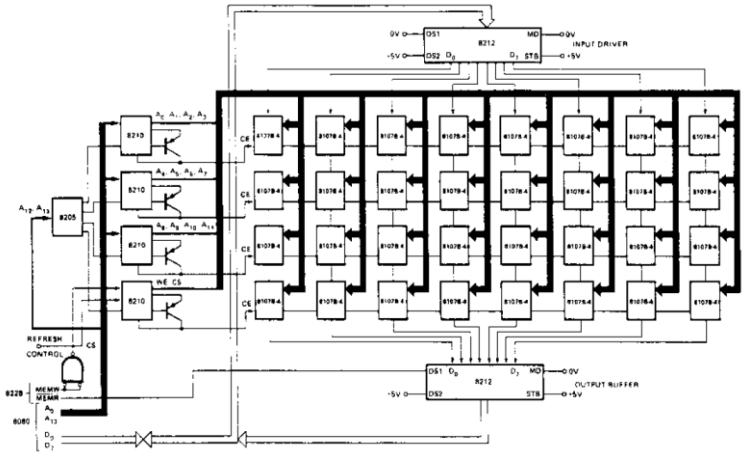


Figure 6.26: Driving an Intel 2107

6.5.6 Testing Intel D8155 and D8156 SRAM

It is possible to test the memory of the Intel D8155 and D8156 (2048-bit static RAM with I/O ports and timer) using an adapter.

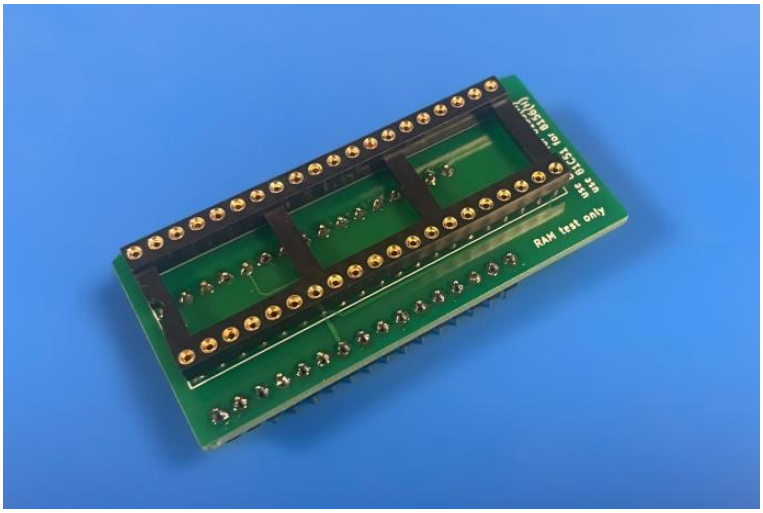


Figure 6.27: Adapter for Intel D8155/D8156

The test is done using the

- 81C50 setting for the Intel D8155, and
- 81C51 setting for the Intel D8156.

6.5.7 Self-defined SRAM, DRAM and ROM memory modules

It is possible to define twelve SRAM, three DRAM, and three ROM modules and use them for tests.

To do this, the SRAM module must be defined in the “customsramX.txt” (X=1, …, 12 / Slot 1 to 12) file and the DRAM module in the “customdramX.txt” (X=1, …, 3 / Slot 1 bis 3) file. ROM modules are defined in “customromX.txt” (X=1, …, 3 / Slot 1 to 3) file. The files must be stored in the root directory of the SD memory card.

After switching on the tester, the files are read in and permanently stored in the EEPROM ("Custom SRAM", "Custom DRAM" and/or “Custom ROM” appears briefly in the display. If the file was read correctly, the extension is changed from "txt" to "old", so that it is not read in again.

The self-defined memory chips can be called via the menu. These can be found at the very end of the menu, directly after the "Self-test" entry.

The self-defined memory chips can be deleted via the configuration menu.

The configuration file for SRAMs (customsramX.txt) can contain the following entries:

Command	Example	Remark
NAME:	2114	Name of the chip (max. 20 characters)
DIP:	18	Number of pins (14, 16, 18, 20, 22, 24, 28, 32)
SIZE:	1024	Number of memory words
FLAGS:	0	Special flags (1 = DOUT inverse, 2 = OE inverse, 4=WR inverse)
A:	5, 6, 7, 4, 3, 2, 1, 17, 16, 15	Address bus (max. 24 values)
DIN:	14, 13, 12, 11	Data bus input (max. 16 values)
DOUT:	14, 13, 12, 11	Data bus output (max. 16 values)
CS:	8	Chip Select (max. 4 values)
PCS:		Positive Chip Select (max. 4 values)
WE:	10	Write Enable (1 value)
OE:		Output Enable (1 value)
GND:	9	0V (max. 4 values)
VCC:	18	+5V (max. 4 values)
VDD:		+12V (max. 4 values)
VBB:		-5V (max. 4 values)
CREAD:	AWcPo1DOCp	Read cycle (max. 20 characters)
CWRITE:	ADOwcP1CpW	Write cycle (max. 20 characters)

The configuration file for DRAMs (customdramX.txt) can contain the following entries:

Command	Example	Remark
NAME:	4116	Name of the chip (max. 20 characters)
DIP:	16	Number of pins (14, 16, 18, 20, 22, 24, 28, 32)
SIZE:	16384	Number of memory words
PAGE:	128	Refresh page size
FLAGS:	0	Special flags (1=DOUT inverse, 2=OE inverse, 4=WR inverse)
ARAS:	5, 7, 6, 12, 11, 10, 13	Address bus ROWs (max. 12 values)
ACAS:	5, 7, 6, 12, 11, 10, 13	Address bus COLUMNs (max. 12 values)
DIN:	2	Data bus input (max. 16 values)
DOUT:	14	Data bus output (max. 16 values)
WE:	3	Write Enable (max. 4 values)
OE:		Output Enable (max. 4 values)
GND:	16	0V (max. 4 values)
VCC:	9	+5V (max. 4 values)
VDD:	8	+12V (max. 4 values)
VBB:	1	-5V (max. 4 values)

The configuration file for ROMs (customromX.txt) can contain the following entries:

Command	Example	Remark
NAME:	27128	Name of the chip (max. 20 characters)
DIP:	28	Number of pins (14, 16, 18, 20, 22, 24, 28, 32)
SIZE:	16384	Number of memory words
FLAGS:	0	Special flags (1 = DOUT inverse, 2 = OE inverse, 4=WR inverse, 8='1'-padding)
A:	10, 9, 8, 7, 6, 5, 4, 3, 25, 24, 21, 23, 2, 26	Address bus (max. 20 values)
DIN:	11, 12, 13, 15, 16, 17, 18, 19	Data bus input (max. 16 values), not required
DOUT:	11, 12, 13, 15, 16, 17, 18, 19	Data bus output (max. 16 values)
CS:	20	Chip Select (max. 4 values)
PCS:		Positive Chip Select (max. 4 values)
OE:	22	Output Enable (max. 4 values)
GND:	14	0V (max. 4 values)
VCC:	1, 27, 28	+5V (max. 4 values)
VDD:		+12V (max. 4 values)
VBB:		-5V (max. 4 values)
CREAD:	AWcPo1DOCp	Read cycle (max. 20 characters)

The memory chip is addressed generically, i.e. an SRAM should behave like a standard SRAM (e.g. a 6116), and a DRAM should behave like a standard DRAM (e.g. a 4164). Since no optimizations are used, such a test takes about 3-4 times as long as a comparable internal test. No timing specifications can be made either. It is therefore not guaranteed that a chip defined in this way can actually be tested.

If an error-free test result is displayed, the chip is probably OK (at least if it is addressed extremely slowly).

If the test fails and you do not have a memory chip that has already successfully tested good, the chip may be defective, but it does not have to be and the failed test result could be due to the chip being incompatible with the generic test.

Example configuration of a 2114 SRAM (customsram1.txt):

```
name:  2114 1k x 4 example
dip:   18
size:  1024
flags: 0

a:     5, 6, 7, 4, 3, 2, 1, 17, 16, 15
din:   14, 13, 12, 11
dout:  14, 13, 12, 11
cs:    8
pcs:
we:    10
oe:    0

gnd:   9
vcc:   18
vdd:
vbb:
```

Example configuration of a 4116 DRAM (customdram1.txt):

```
name:  4116 16k x 1 example
dip:   16
size:  16384
page:  128
flags: 0

aras:  5, 7, 6, 12, 11, 10, 13
acas:  5, 7, 6, 12, 11, 10, 13
din:   2
dout:  14
ras:   4
cas:   15
we:    3
oe:

gnd:   16
vcc:   9
vdd:   8
vbb:   1
```

When the generic read/write cycles are not compatible with an IC, e.g. if the address and data have to be stored clocked, these can be defined using the CREAD and CWRITE commands. This definition does not allow precise timing and is very slow compared to "normal" tests. For this reason, this definition is only possible for SRAMs and ROMs. Of the flags, only "DOUT inverse" and "'1'-padding" are taken into account.

Character	Meaning (CREAD and CWRITE)
A	Setting the lower address (address width as defined in “a:”)
a	Setting the upper address (address width as defined in “a:”)[3]
D	Writing data (CWRITE), read data (CREAD)
C	all CS set HIGH
c	all CS set LOW
P	all PCS set HIGH
p	all PCS set LOW
W	WE set HIGH
w	WE set LOW
O	OE set HIGH
o	OE set LOW
1..4	Delay of 60/120/180/240ns

[3] e.g. used for RCA CDP1831 and CDP1833

6.6 Problematic memory modules

In this chapter, some “problematic” memory modules are described and ways in which they can be checked are presented. However, there is no guarantee that these “workarounds” will always work. Unfortunately, due to the flexibility of the tester, you have to accept these (minor) restrictions.

6.6.1 Testing older DRAMs

DRAMs can usually be tested without any problems. Unfortunately, there are a few DRAMs, mainly of the types 2104er (4k x 1), 4116 (16k x 1), 3732 (32k x 1), 4164 (64k x 1), 4416 (16k x 4) and sometimes 4464 (64k x 4), where a test aborts with an error at the very beginning (at address "0"). There are several possibilities why a test cannot be done, but there are also workarounds for known issues:

1. Some types have an increased power consumption which has to be intercepted with a decoupling capacitor (mostly > 500mW). To do this, clamp a 100 nF capacitor between Vcc and Vss in the socket. For a 4116, these are pin 9 (Vcc) and pin 16 (Vss). For a 4164, these are pin 8 (Vcc) and pin 16 (Vss).

   You can also use the Decoupling Adapter (see sec. 6.6.4) for this (set the corresponding jumper) or the “on-board” Decoupling Capacitor.

2. Due to an increased current consumption, a higher voltage drop across the transistors may occur that results in the supply voltage dropping to a level that is out-of-specification for the specific device being tested when the device is powered on. A simple blocking capacitor between Vcc and Vss is no longer sufficient here. In this case connect the Vcc pin of the IC to Vcc (5V) on the upper socket at location J4 between the power module and the LCD. Check the J4 connector carefully because most pins there have voltages on them and selecting the wrong position can potentially damage the IC you are trying to test.
3. Some older DRAMs require pull-ups on the data lines so that the signal levels are sufficient. To do this, try switching on the pullups that are switched off by default in the configuration (Config → "DRAM Pullup: 1".)

When DRAMs of the above types are tested, the following procedure can be used:

First test the DRAMs without a decoupling capacitor. If these are tested as OK, then they are to be classified as "very good" with regard to their behavior. The ICs that were tested as defective without a decoupling capacitor are then tested again with a decoupling capacitor. The ICs that have been tested as OK are also OK within the scope of the specification, but have higher inrush currents and generate interference on the Vcc line due to current peaks. All ICs that were tested as defective even with a decoupling capacitor should be discarded.

6.6.2 Testing ICs having a higher current consumption

The Tester is able to test ICs with a peak power consumption of 1 Watt (and more). However, in the case of a few ICs (especially ICs with signs of aging), an increased power requirement occurs especially at the beginning when initially powered on, which can lead to malfunctions.

If, with an older IC, the test ends with an error right at the beginning (address 0) or the IC in the socket is not recognized at all, the following measures may help:

- Add a 100nF capacitor (block capacitor) between GND and Vcc (clamp together with the IC in the socket). This bridges the suddenly falling voltage.
- Connect Vcc and/or GND directly to the voltage supply of the Tester (socket between display and DC/DC module). This prevents an increased voltage drop at the transistors due to an increased power requirement.

Usually, the additional capacitor should be sufficient. In old circuits, these block capacitors can be found on every IC. Unfortunately, due to the different pin assignments of the ICs to be tested, such a capacitor cannot be installed permanently on the RCT ZIF socket, so when required you must make the connection manually.

If the test does not pass even with these measures, the IC should no longer be used.

6.6.3 Testing the output drivers of ICs

With older ICs it can happen that the output drivers are no longer able to deliver the current specified in the data sheet (often between 5 - 20mA). In a real system, errors can occur because the output can no longer supply the required current. Such component can still be tested in the tester as error-free, since the outputs are virtually not loaded by the (MOS) inputs of the MPU (only a few uA → voltage-controlled).

When you want to test the outputs under load, you have to apply an external resistor. This can easily be done with the breakout adapter or by clamping an appropriate resistor in the ZIF socket.

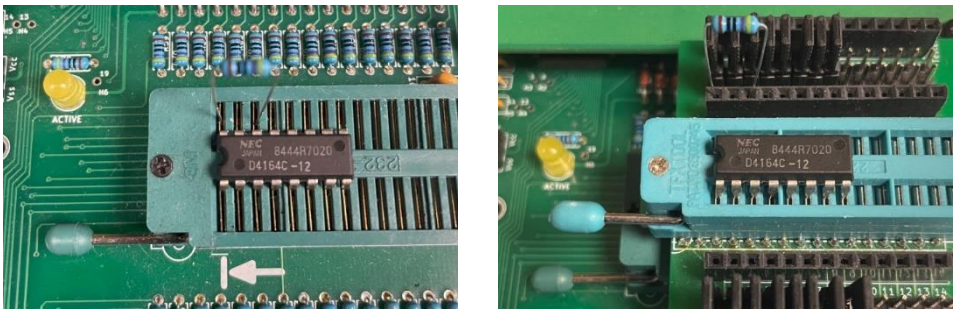


Figure 6.28: Load Resistor

The NEC uPD4164 shown in the picture should be able to deliver 5mA. A 1k Ohm resistor loads the output with about 5mA. The test should continue to run without errors if the output drivers are working.

If the output is to be loaded with 10mA, a 470 Ohm resistor can be used. Please consult the data sheet for the maximum output current that can be used to avoid damaging the component.

6.6.4 Using the Decoupling Adapter

Some DRAMs have an increased power consumption that has to be absorbed with a decoupling capacitor. For this purpose, a 100 nF capacitor can be clamped into the socket between Vcc and Vss. Instead of clamping a 100 nF capacitor between Vcc and Vss in the socket, you can use the Decoupling Adapter.

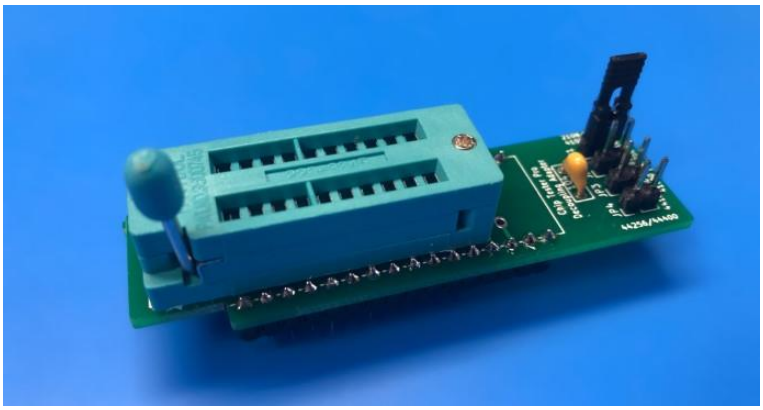


Figure 6.29: Decoupling Adapter

Usually, this adapter is not required since Vcc can be provided quite reliably. When a DRAM has an increased power consumption (mostly > 500mW) and the test fails immediately you can use this adapter.

This adapter can be used with following ICs:

Package	Vcc Pin[4]	Jumper	Supported ICs (and others with same pinout)
DIP16	9	JP1	DRAMs, e.g. 4116, 4108, 2108, 2104
DIP16	8	JP2	DRAMs, e.g. 4816, 4164, 3732, 4128, 41256 SRAMs, e.g. 3101, 2112 SRAMs, e.g. 6561 (DIP18(!)) and most logic ICs
DIP18	9	JP3	DRAMs, e.g. 4408, 4416, 4464, 41000 SRAMs, e.g. 74c910, 6518, 2114 and most logic ICs
DIP20	10	JP4	DRAMs, e.g. 44256, 44400 SRAMs, e.g. 4047 and most logic ICs

Figure 6.30: Supported ICs

The decoupling capacitor can of course also be used with many SRAMs and logic ICs in DIP16, DIP18 and DIP20 housings, even if it is not explicitly necessary.

The board rev.1.2k has the same jumpers next to the ZIF socket, so these can be used instead of the decoupling adapter.

[4] Vss for SRAMs and Logic-ICs

The decoupling capacitor can be set using four jumpers:

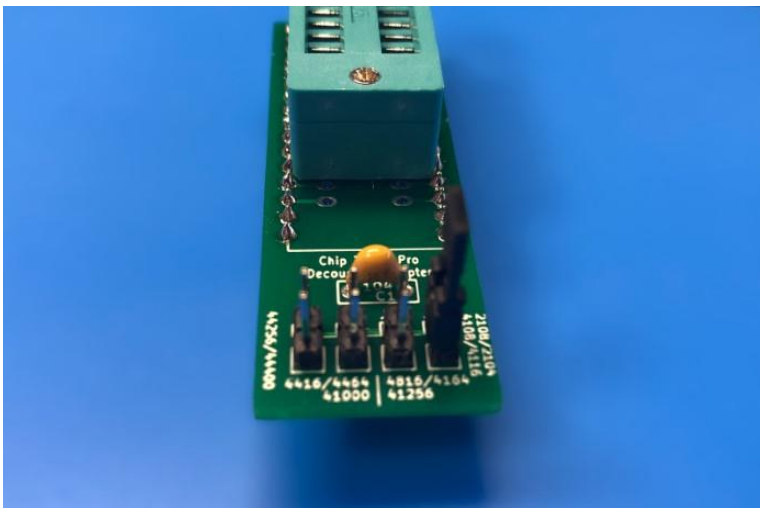


Figure 6.31: Jumpers

The 1x03 pin header is used to power the IC directly. When an IC requires lots of current you can connect it directly with the J4 power header socket located between the DC/DC module and LCD display.

This does not seem to be necessary with the ICs that have already been tested, but the board is prepared for it. You can bypass the transistors with wires and connect the IC power pins directly to the J4 power header socket located between the DC/DC module and the LCD display.

6.6.5 Using the “on-board” Decoupling Capacitor (from rev.1.2k)

Directly next to the ZIF socket there are three pin headers with which a 100nF decoupling capacitor can be switched on during DRAM tests.

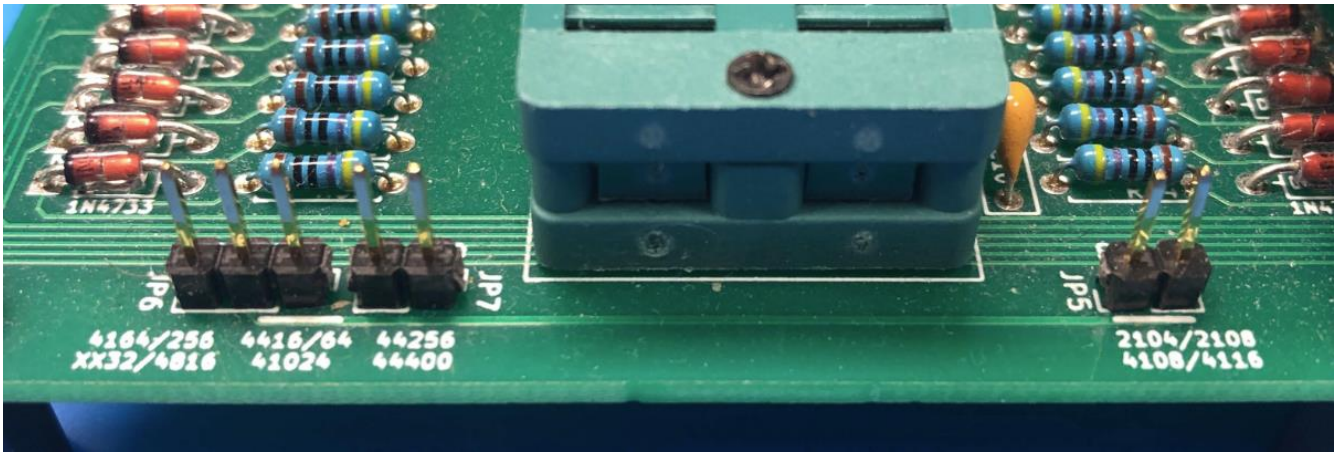


Figure 6.32: “on-board” Decoupling Capacitor

The supported ICs are the same as with the Decoupling Adapter:

Package	Vcc Pin[5]	Jumper	Supported ICs (and others with same pinout)
DIP16	9	JP5	DRAMs, e.g. 4116, 4108, 2108, 2104
DIP16	8	JP6 links	DRAMs, e.g. 4816, 4164, 3732, 4128, 41256 SRAMs, e.g. 74c910, 3101, 2112 SRAMs, e.g. 6561 (DIP18(!)) and most logic ICs
DIP18	9	JP6 rechts	DRAMs, e.g. 4408, 4416, 4464, 41000 SRAMs, e.g. 6518, 2114 and most logic ICs
DIP20	10	JP7	DRAMs, e.g. 44256, 44400 SRAMs, e.g. 4047, 2114 and most logic ICs

Figure 6.33: Supported ICs

The decoupling capacitor can of course also be used with many SRAMs and logic ICs in DIP16, DIP18 and DIP20 housings, even if it is not explicitly necessary.

[5] Vss for SRAMs and Logic-ICs

6.6.6 Testing ITT4027 (and if needed compatible SRAMs)

The ITT4027 memory IC (and presumably also all compatible SRAMs with comparable power consumption) might have problems with the switched voltage Vss.

To test these components, pin 16 (Vss) must be connected directly to Vss/GND. For this connect pin 16 with Vss on the 7-pin socket header.

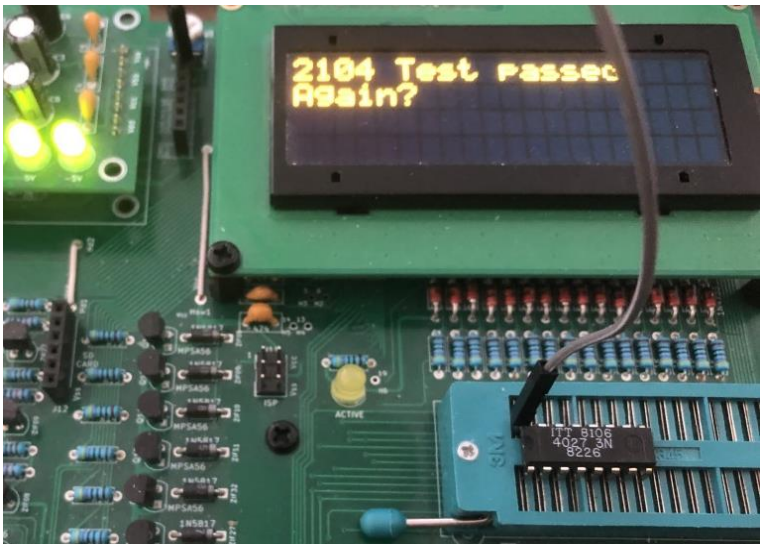


Figure 6.34: Testing the ITT4027

Alternatively, a (film) capacitor of 100nF can be clamped between Vss (pin 16) and Vcc (pin 9) (decoupling capacitor) due to the high-power consumption.

6.6.7 Testing SIMM/SIPP modules

The Tester can also test SIMM/SIPP modules. However, the maximum power consumption of the modules must be observed here. The Tester delivers a maximum of approx. 300-400mA. If the maximum power consumption of the modules remains roughly within these limits, it should be possible to test them.

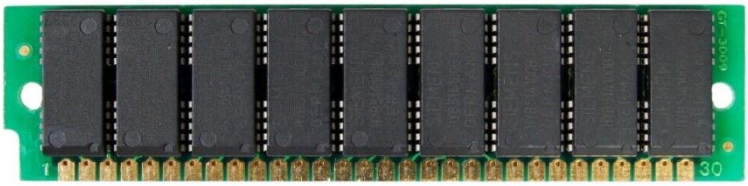


Figure 6.35: SIMM 30 (9 ICs)

The following data sheets show roughly where the limits are:

Operating Current* ($\overline{RAS}$, $\overline{CAS}$, Address Cycling @ t_{RC}=min.)	KMM594000A- 7 KMM594000A- 8 KMM594000A-10	I_{CC1}	— — —	945 855 765	mA mA mA

The module (with nine ICs) cannot be reliably tested (the values are maximum values; the actual current consumption is lower). The following applies: the faster the module and the more ICs, the higher the current consumption.

The following module (with three ICs) can be tested:

Operating Current * ($\overline{RAS}$, $\overline{CAS}$, Address cycling @tRC=min.)	KMM594100AN - 5 KMM594100AN - 6 KMM594100AN - 7 KMM594100AN - 8	ICC1	- - - -	305 275 245 215	mA mA mA mA

6.6.8 DRAMs from Russian or Eastern European production

When memory tests of DRAMs from Russian or Eastern European production fail frequently while being tested, the voltage level could be too low for a logical high due to the age of the components. To solve this, the pull-up resistors for DRAMs in the configuration menu can be switched on temporarily. This increases the high level slightly so that the test may pass. When the tests are completed be sure to put the pull-up resistor setting in the configuration menu to off as it can affect other tests negatively.

However, it should be noted that if these chips are used in a device (such as an old computer) and still have issues, the level there is probably too low or at least "close to the limit". This can happen as ICs age. You should consider replacing the IC with one that operates reliably otherwise it can affect the function of the device.

6.7 Testing of ROM, PROM, EPROM or EEPROM

The Retro Chip Tester can test or read a variety of ROMs, PROMs, EPROMs and EEPROMs. The testing is based on the fact that a CRC32 is calculated over the content, which can be compared with an already known checksum. If this is available in the internal database, the content is displayed directly. When a SD card is connected, the content can also be read out and saved.

When selecting a memory chip, some additional information is shown in the display:

Figure 6.36: EPROM 2708

In the fourth line, information about the power supply is usually displayed. For EPROM 2708 these are Vss, Vcc, Vbb and Vdd. Since the size of the display is limited, only Vdd (12V on pin 19) and Vbb (-5V on pin 21) are shown here.

With some EPROMs, e.g. the 2764, special settings are also displayed **during the test**, e.g. Vcc on pin 1 (Vpp) and Vcc on pin 27 (/PGM).

Empty "blocks" (1 block = 1/16 of the total memory size of a chip) are displayed with an empty box in the progress bar. This way you can quickly see whether a memory is e.g. half empty.

A memory cell is considered "empty" if the content is 0xFF (or 0xF). This also applies to bipolar PROMs, which contain 0x00 by default.

6.7.1 Identifying content

If a ROM, PROM, EPROM or EEPROM is inserted, a CRC32 can be calculated. The checksum can be used to determine if the content can still be read correctly. Almost 700 ROMs are recognized and displayed accordingly.

Figure 6.37: ROM recognized as “901447-10” for the Commodore PET

If the CRC32 is known for a ROM, its identifier is displayed, otherwise the calculated CRC32 is displayed. If the CRC32 changes several times, the chip is most likely defective or the supply voltage too low. In this case, use the barrel jack to supply the tester with voltage (problem description in chapter 3). In this case, use the barrel jack to supply the tester with voltage.

The calculation of the CRC32 over large memory chips, e.g. 271000/231000 and above (from 128k x 8) is extremely slow. The function was built in to be able to determine a checksum in rare cases, not to do series tests with it.

6.7.2 Reading of ROM, PROM, EPROM or EEPROMs

With a suitable micro SD card adapter, memory chips can also be dumped.

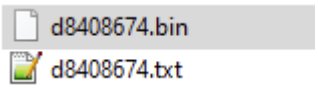
To do this, support for SD cards must be switched on in the configuration ("**SD Card: 1**") and the micro SD card adapter must be connected to the chip tester. The connection is described in sec. 8.

With support switched on, when you…

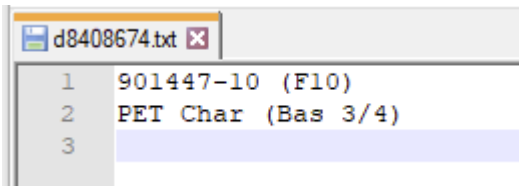
… long press the OK button, the content of the memory chip is written to the file `<crc32>.bin`. When the ROM is known, another file `<crc32>.txt` is created, which contains a description of the memory chip.

Example: PET Character ROM for Basic 3 and 4

Two files are created:

- 

The content of the text file is:

- 

… short press the OK button, only the CRC32 is calculated and if possible, the recognized memory chip is displayed.

If a memory has less than 8-bit data bus width, it is padded with '0' as standard. The padding can be switched to '1' in the configuration. The padding is included in the CRC32 calculation.

The memory card must be formatted FAT16 or FAT32. If it is not recognized, format it once with the SD Association's SD Memory Card Formatter.
Download from: https://www.sdcard.org/downloads/formatter/

Since the tester has no real time clock, no file date is set by default. There is the option of specifying a date and time, when a file `datetime.txt` is created on the SD card. If the Tester is started (or after a reset), the file date and time will be used when creating additional files.

6.7.3 Enlarging the internal ROM database

The chip tester has an internal database with some ROMs. This list can be expanded using an external database file on the SD memory card.

The file must have the name "customcrc32.txt" and be structured as follows:

```
; This is a comment
<crc32>; <rom name>; <rom description>
```

As an example:

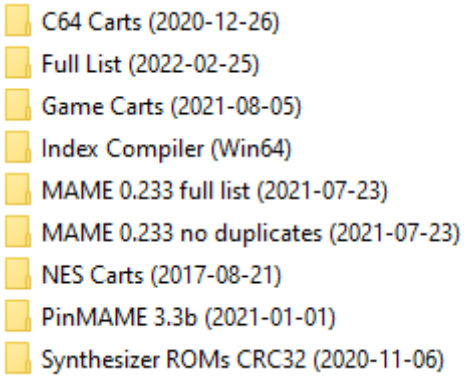
```
0xd8408674;901447-10 (F10);PET Char (Bas 3/4)
```

The CRC32 must have the prefix "`0x`". Only the first 40 characters are taken into account, of which only the first 20 characters can be shown on the display.

To search the database file quickly, an index file (customidx32.bin) must be created with the included compiler (customcrc32.exe). The index file has to be stored on the SD memory card together with the database.

6.7.4 Using prepared databases

There are some prepared databases for download:



The most comprehensive database is the "Full List" with around 400,000 entries, which includes all other databases. Basically, there is nothing wrong with using this database in principle, but you should not forget one disadvantage of the CRC32 procedure: Collisions can occur, i.e. ROMs with different content can have the same CRC32 checksum. The probability is about 1 in 10,000. If you only work with synthesizer ROMs, you should rather use the specialized database, which reduces with only 2000 entries the probability of a collision to about 1:2,000,000.

Once you have decided on a database, the two files

`customcrc32.txt` and `customidx32.bin`

are simply copied to the main directory of the SD card. The next time you read a ROM (short press on OK) or save a ROM (long press on OK), the content will be identified using this database

6.7.5 Special features when reading ROMs and PROMs

When reading ROMs, a requirement must be observed: The signals “Chip Select”, “Chip Enable” and “Output Enable” are set during manufacture to be either active-low or active-high. Most chips are active-low, but many early chips from the late 70's have their own scheme and of course this has an impact if the chip to be read does not use the expected signal level. The Chip Tester supports these most common signal level combinations:

ROM	Type	Pin - Signal	active	for example used in
2308 - 1k x 8	A	18 – /CS2* 20 – /CS1	low low	
	B	18 – /CS2* 20 – CS1	low high	
	C	Autodetect	high/low	
2316 - 2k x 8	A	18 – /CS2* 20 – /CS1* 21 – CS3*	low low high	Commodore
	B	18 – CS2* 20 – /CS1* 21 – /CS3*	high low low	Apple II
	C	Autodetect	high/low	
2332 - 4k x 8	A	20 – /CS1* 21 – CS2*	low high	Acorn Atom Commodore CBM, C64
	B	20 – /CS1* 21 – /CS2*	low low	TI 99/4A Sinclair ZX80
	C	20 – CS1* 21 – CS2*	high high	Gottlieb Pinball Machines
	D	Autodetect	high/low	
2364(24) - 8k x 8	A	20 – /CS*	low	Commodore 1541, VC20 Atari 400/800, DAI EACA Video Genie Sinclair ZX81, Vtech VZ 200
	B	20 – CS*	high	
	C	Autodetect	high/low	
2364(28) - 8k x 8	A	20 – /CE 22 – /OE 26 – CS2 27 – CS1	low low high high	VEB Robotron KC87/KC85 Apple Lisa, CoCo 2 Vtech VZ 300 (Memotech MTX)
	B	Autodetect	high/low	
23128 - 16k x 8	A	20 – /CS2* 22 – /CS1* 27 – CS3*	low low high	Commodore 16/116, Plus/4 Commodore C64/C128 BBC B, Matra Alice 32, Oric I
	B	20 – /CS2* 22 – /CS1* 27 – /CS3*	low low low	Sinclair ZX Spectrum BBC B, Dragon 32/64
	C	20 – CS2* 22 – OE* 27 – /CS3*	high high low	Sinclair QL
	D	Autodetect	high/low	

ROM	Type	Pin - Signal	active	for example used in
23256 - 32k x 8	A	20 – /CE* 22 – /OE*	low low	Amstrad CPC, Atari ST Commodore C128 Enterprise 64, CoCo 3
	B	20 – /CE* 22 – OE*	low high	Sinclair QL
	C	Autodetect	high/low	
23512 - 64k x 8		20 – /CE 22 – /OE	low low	
231000 - 128k x 8		20 – /OE*	low	
232000 - 256k x 8		22 – /CE 24 – /OE	low low	
234000 - 512k x 8		22 – /CE 24 – /OE	low low	
6540 - 2k x 8		2 – /CS5 3 – /CS4 4 – /CS3 17 – CS1 27 – CS2	low low low high high	Commodore PET

*) Signal (high- or low-active) programmable

If bipolar (P)ROMs are read out, it should be noted that the Chip Tester does not process the data and store it as it was read out. This is important if the data is used to program a new PROM, as the data for some PROMs must be inverted for this. Usually, the programmer's software should fix this.

You should use the known CRC32 database in the root dir of the SD card to ensure your read is good and identified as a known good ROM. See section 6.7.3 for more info.

Since some (P)ROMs have Open Collector outputs, ensure that the pull-ups are activated (standard) for these types. When unsure check the datasheet for your IC.

With 4-bit memory, the four bits are stored in the lower four bits of a byte. If a PROM with 256x4 bits is read out, a file with 256 bytes is stored on the SD memory card, which contains the data in the lower 4 bits of each byte.

When a ROM is selected for which several profiles are stored, the active state of some pins needs to be selected. The display shows the pins that are active-low after “L=” and the pins that are active-high after “H=”. Brief IC pin orientation info is given in the 4th line.

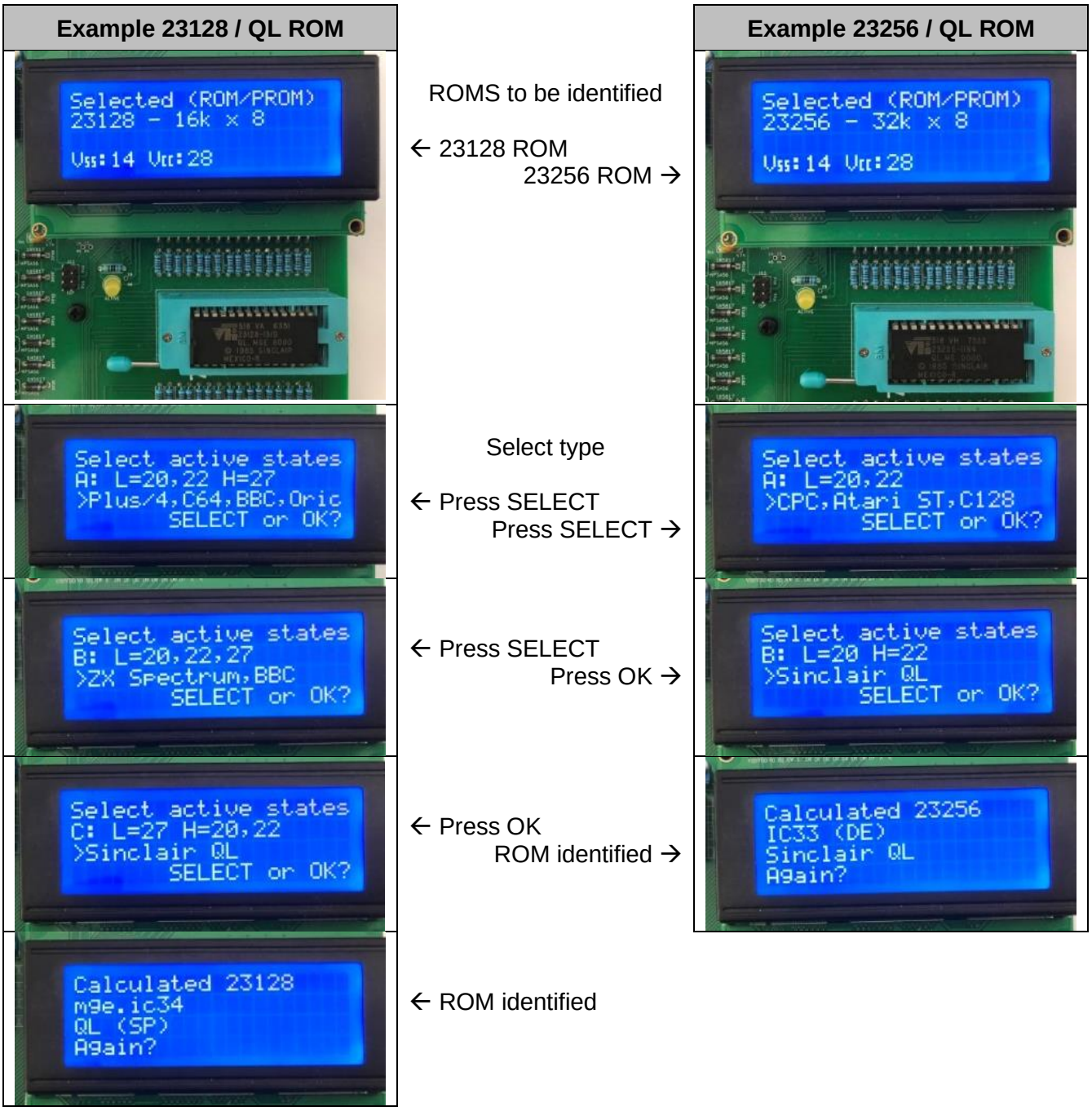
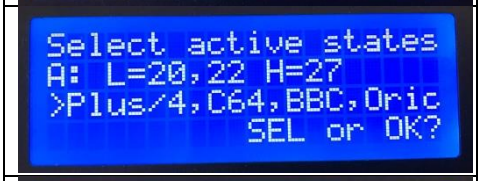
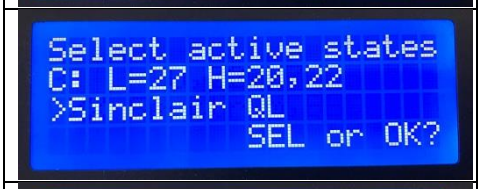
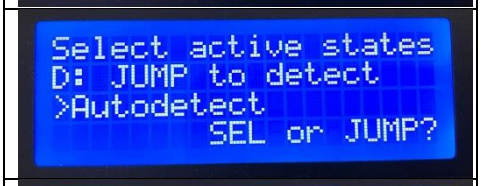
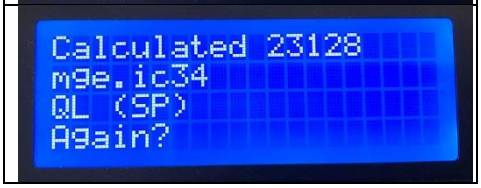


Figure 6.38: Selection of the „active states“ of a 23128 and 23256

The automatic detection of the “active states” of the Chip Select and Chip Enable signals is much more convenient with some (P)ROMs. If "Autodetect" or "JUMP to detect" is shown in the display, press JUMP once. The Tester now tries to recognize how the data can be read out. The following shows the procedure using the example of a 23128:

Example 23128 / QL ROM	
	ROM to be identified ← 23128 ROM
	Select type (A) ← Press SELECT
	Select type (B) ← Press SELECT
	Select type (C) ← Press SELECT
	Select type (D) - Autodetect ← Press JUMP
	Pin 27 recognized as low-active, pins 20 and 22 as high-active ← Press OK
	← ROM identified

6.7.6 Dumping of 2716 (2k x 8), 2532 (4k x 8), 2508 (1k x 8), 2758 (1k x 8)

When dumping the 2716 (2k x 8), 2532 (4k x 8), 2508 (1k x 8) and 2758 (1k x 8) EPROMs, it should be noted that these EPROMs also have a voltage supply of 5V at pin 21 (Vpp) for reading the EPROM. Since the RCT does not have a Vcc driver on this pin, it is set to a logical high. This is usually sufficient to read out the mentioned ICs.

However, there are some EPROMs that have an increased current consumption at Vpp, so that the voltage level at Vpp drops too much. Different CRC32 values are then read each time they are dumped, i.e. the dump is not reliable.

The problem also occurs with other testers and can easily be eliminated with an intermediate socket.

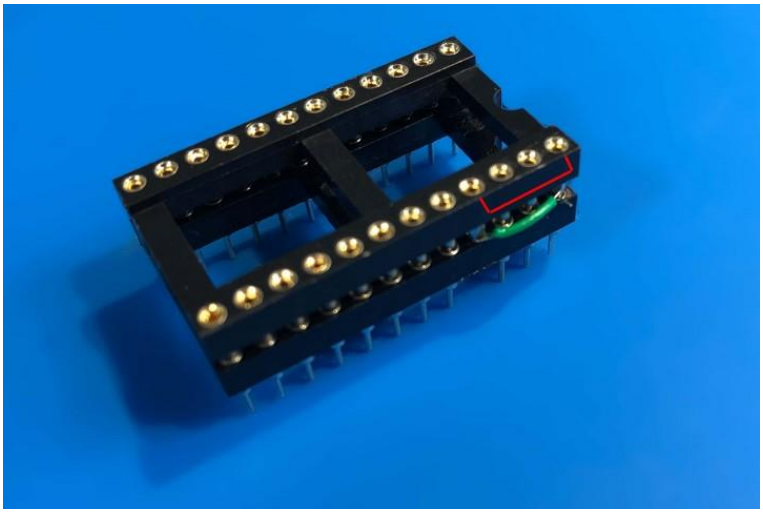


Figure 6.39: Intermediate socket for Vpp at 5V

Pin 21 on the lower socket has been removed so that there is no longer a connection to the ZIF socket. Pins 21 (Vpp) and 24 (Vcc) of the upper socket have been connected with a small wire bridge (see red marking).

Alternatively, the definition "2716 Rev (2k x 8 - EPROM).txt" can be used for the 2716. The IC must then be inserted rotated 180 degrees. This means that Vcc can also be injected on pin 21.

Note:

A "TMS2516" is **compatible** with the usual "2716". A "TMS2716" is **not compatible** with the "2716"; several supply voltages are required here.

6.7.7 Dumping an Intel 1702A / 1602A

It is possible to dump the Intel 1702A / 1602A with the help of an adapter.

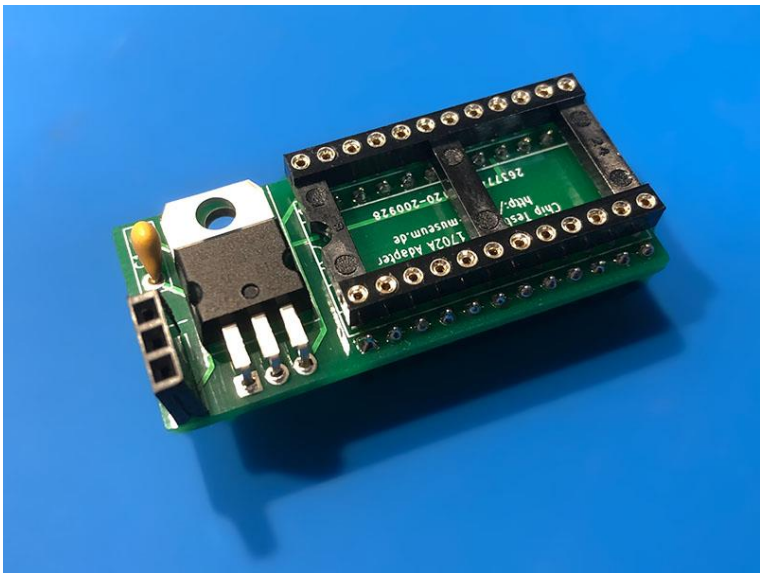


Figure 6.40: Adapter for the Intel 1702A / 1602A

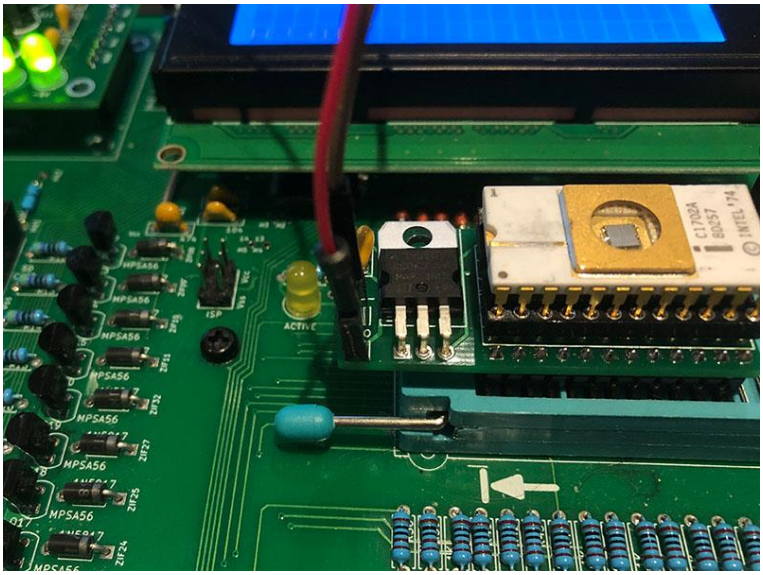


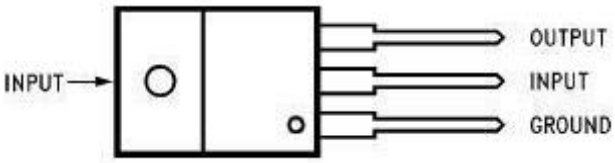
Figure 6.41: Adapter in socket

The connections

- “Vss/GND” with “Vss/GND” on the Tester (socket header to the right of the DC/DC module), and
- “-12V” with “-12V” on the DC/DC-Module

must be connected so that the adapter can provide the supply voltage for the Intel 1702A / 1602A.

Older DC/DC module boards do not yet have a connection for -12V, here the -12V can be tapped on the cooling plate of the LM7905 (see picture, marked with “Input”).



The adapter has been successfully tested with several Intel 1702A and Intel 1602A chips. However, due to the nature (and especially the age) of the chip, there is no guarantee that reading will always work reliably.

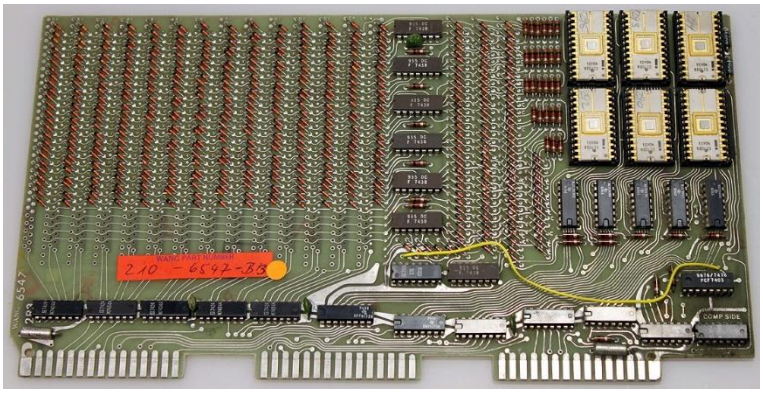


Figure 6.42: Tested 1702A

When reading out, please observe the following precautionary measures so that a short circuit cannot occur accidentally (e.g. when the adapter is dropped):

1. First insert the chip into the adapter.
2. Insert the adapter with the chip into the Tester.
3. Lastly, connect the voltage supply with the board (Vss and -12V).

6.7.8 Dumping characters ROMs RO-3-2513 or Motorola MCM6670 series

The Retro Chip Tester can also dump character ROMs. However, a few special features of these ROMs must be taken into account.

Character ROM RO-3-2513 (64 x 8 x 5):

The 2513 ROM is available in two versions: One with 5V power supply (with the label RO-3-2513) and one with +5V/-5V/-12V power supply. Without an adapter, only the 5V variant can be dumped directly. The data is read via the address lines as specified (pin 14 = A0 to pin 22 = A8). The missing bits 6 and 7 are padded with "0". The stored file has a length of 512 bytes.

Character ROM MCM6670 (128 x 7 x 5):

The MCM6670 series from Motorola (including the SCM37530 from the TRS-80 Model 1) has 7 address services for 128 characters with 7(!) rows (RS1 - RS3) and 5 data lines.

A total of 1024 memory words with 5 bits each are read. The missing bits 6 and 7 are padded with "0". For the bit combination RS1 = RS2 = RS3 = 0 (= row 0) the ROM always returns "0x00". The stored file has a length of 1024 bytes.

Figure 6.43: The first 21 bytes (24 bytes with row 0) of an SCM37530

In the figure, a “0” has been replaced by a space and a “1” has been replaced by an asterisk.

6.7.9 Dumping character ROM 2513 (+5V/-5V/-12V version)

The Character ROM 2513 requires three voltages: +5V, -5V and -12V. The missing voltage of -12V is made available with the help of an adapter. The ROM is then read out via setting 2704.

The data is read via the address lines as specified (pin 14 = A0 to pin 22 = A8). The missing bits 6 and 7 are padded with "0" (the adapter has pulldown resistors for this). The stored file has a length of 512 bytes.

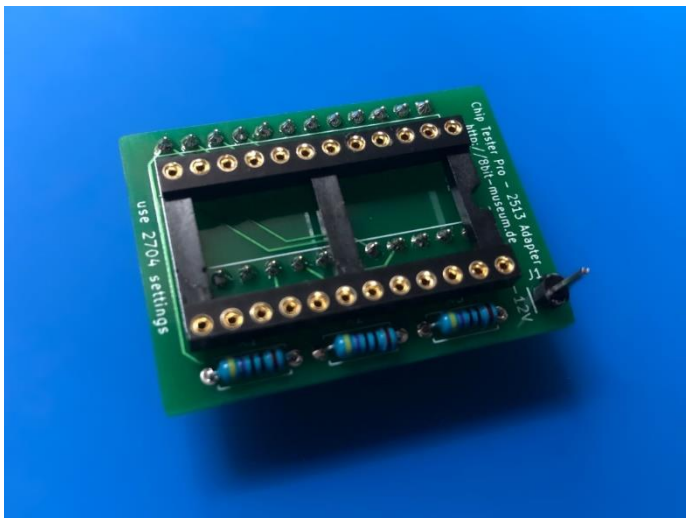


Figure 6.44: 2513 Adapter (+5V/-5V/12V version)

6.7.10 Dumping the ROMs MM4204 and MM5204

The ROMs MM4204 and MM5204 require two voltages: +5V and -12V. The missing voltage of -12V is made available with the help of an adapter. The ROM is then read out via setting 2704.

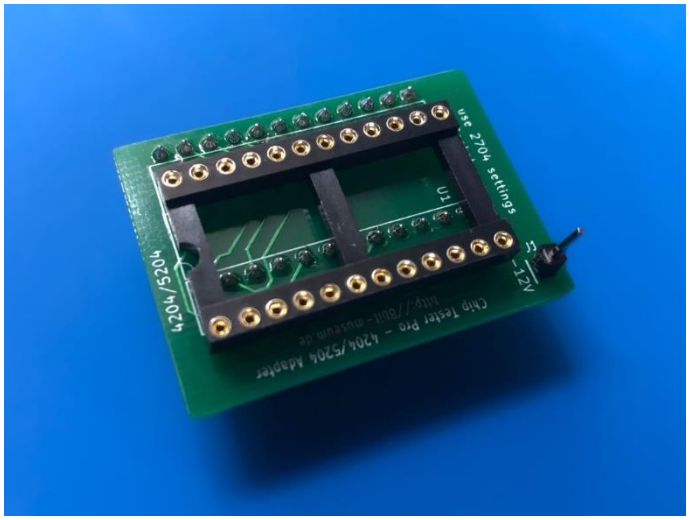


Figure 6.45: 4204/5204 Adapter

Attention: The IC is rotated 180 degrees in relation to the Tester socket.

6.7.11 Dumping the ROM Signetics 2530

The ROM 2530 from Signetics was rarely used, e.g. in the arcade machine "Qwak!". The ROM can be dumped using the custom definition “2530 (512 x 8 - ROM, ADAPTER).txt” (from Firmware v.22) and an adapter.

Since the adapter is quite simple it can be build using two 24 pin sockets (round holes):

1. The first socket is put into the ZIF socket, the second one of mounted on top of the first one.
2. Lower socket: We remove pin 4 and pin 12, so there is no connection. Pin 12 is used to inject 12v, pin 4 is used for Vss.
3. Upper socket. We add a wire to pin 12 (-12v). This pin is not connected to the lower socket, it is connected to the -12v connector on the DC/DC board.
4. Since ZIF pin 4 cannot provide Vss, we use Vss from pin 17 (ZIF pin 25). We connect pin 17 from the lower socket to pin 4 of the upper socket.
5. Connect the lower and upper socket, so all other pins are 1:1 mounted.

A ZIF socket can of course also be used for the upper socket.

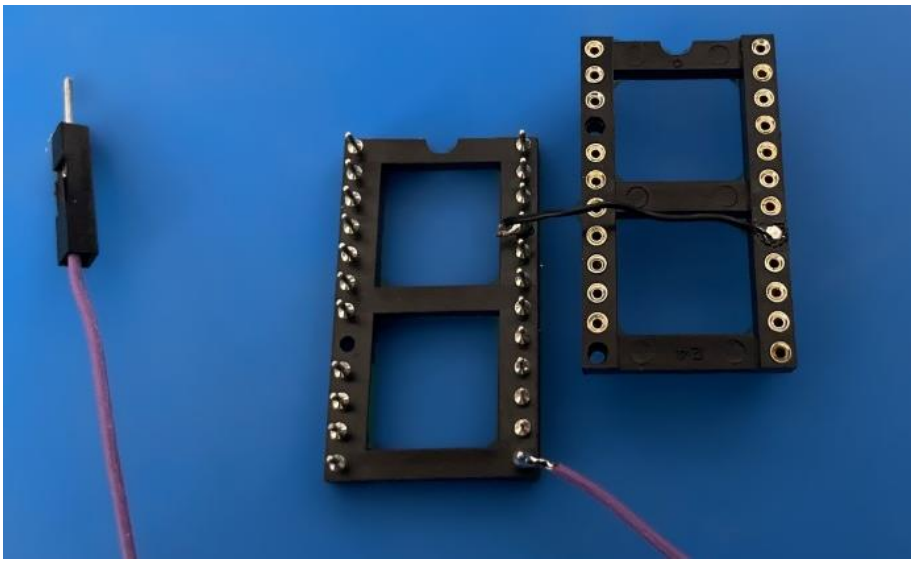


Figure 6.46: 2530 Adapter (self-build)

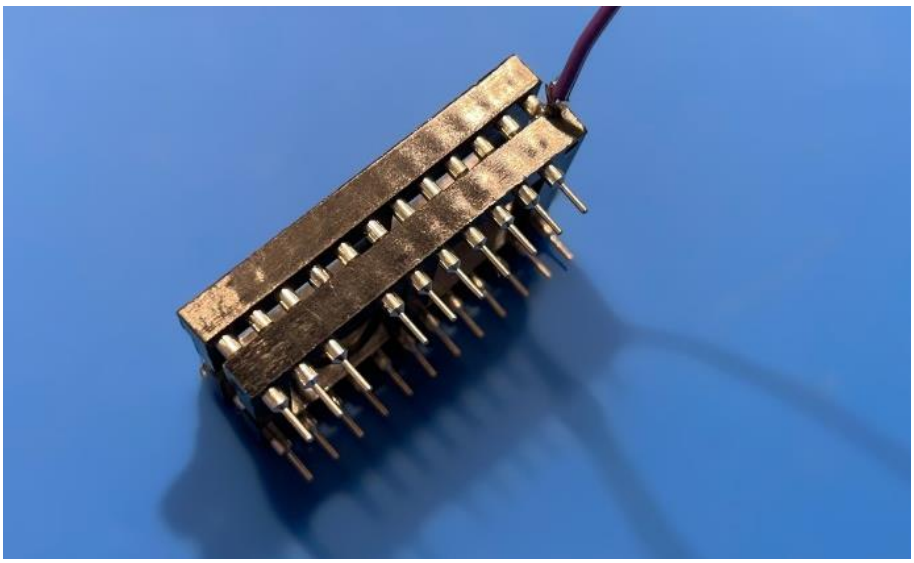


Figure 6.47: 2530 Adapter (self-build, side view)

Figure 6.48: 2530 Adapter

Figure 6.49: Dumping a 2530 ROM

6.7.12 Dumping the ROM Mostek MK28000 and TMS4800

The ROM MK28000 from Mostek was used very rarely, e.g. in the arcade machine "Tank". The ROM can be dumped using the custom definition "MK28000 - TMS4800 (2k x 8 - ROM, ADAPTER).txt" (from firmware v.22) and an adapter.

The MK28000 requires pulldowns on the data lines applied by the adapter (10k Ohms). By default, the RCT works with pullups enabled for ROMs (since many older devices have open-collector outputs). For this reason, the pullups should be switched off in the configuration for reading out an MK28000.

As a rule, reading out also works with activated pullups (these have approx. 10k to 20k Ohms), but then only approx. 1.5v to 2v is achieved for a logical LOW on the data lines due to the voltage divider. This should be recognized as a logical LOW, but reading out with deactivated pullups is more reliable. Alternatively, the pulldowns on the adapter could be reduced to 4.7k Ohms or even 3.3k Ohms.

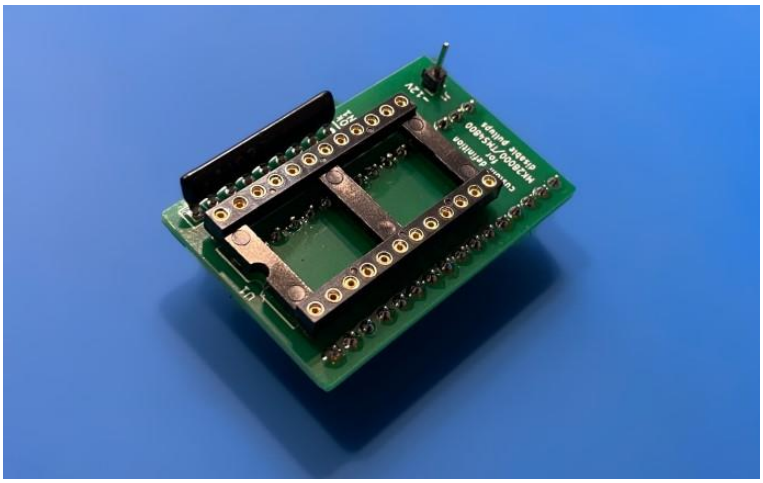


Figure 6.50: MK28000 Adapter

A self-built version of the adapter can be found in the BOM document.

6.7.13 Dumping the EEPROM uPD454 and ROM uPD464

The NEC uPD454 and uPD464 can be read using a simple adapter. It is dumped via the custom definition „uPD454 - uPD464 (256 x 8, EEPROM, ROM).txt“.

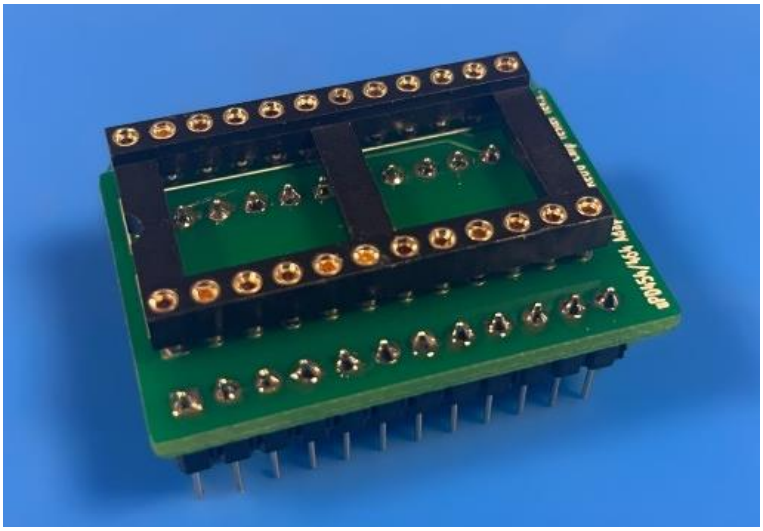


Figure 6.51: uPD454 / uPD464 Adapter

6.7.14 Dumping the EPROMs CDP18U42

The EPROM CDP18U42 from RCA was rarely used. The EPROM is dumped via the setting "74S271/470" (74S271, 74S371, 74S470, 74S471).

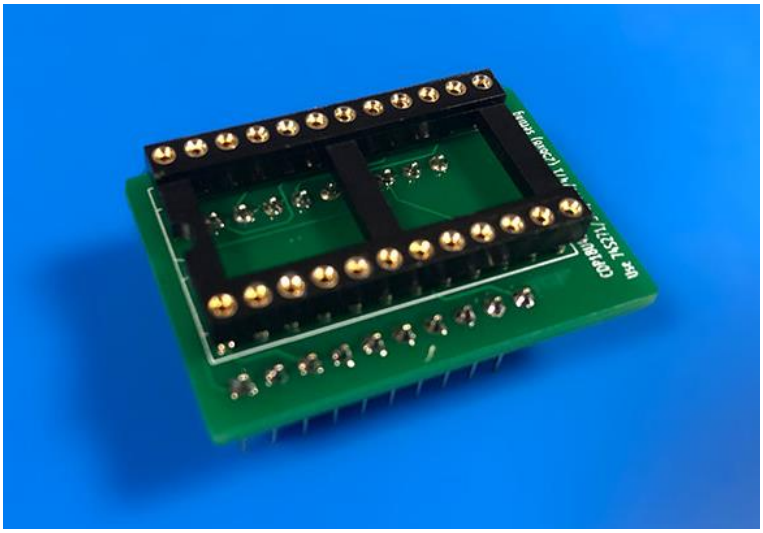


Figure 6.52: CDP18U42 Adapter

6.7.15 Dumping Commodore C64 Cartridges

C64 cartridges with a storage capacity of 16 kByte (2x 8 kByte) can be dumped. Switching between ROMH and ROML is realized by a simple switch. The setting 2764 (8k x 8 EPROM) is used for dumping the cartridge.

The label on the cartridge points to the front when dumping.

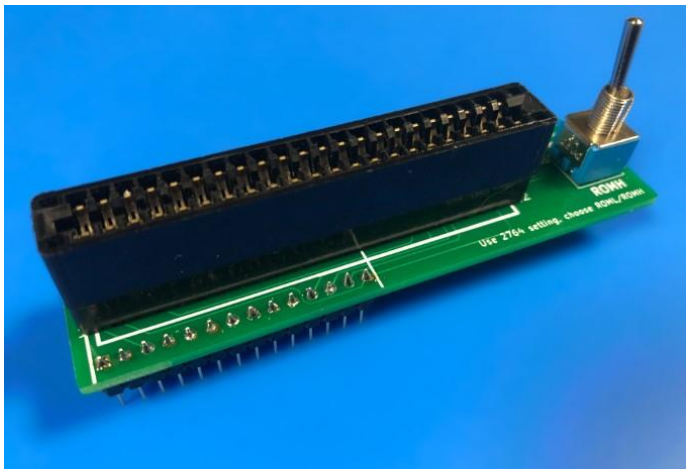


Figure 6.53: C64 Cartridge adapter

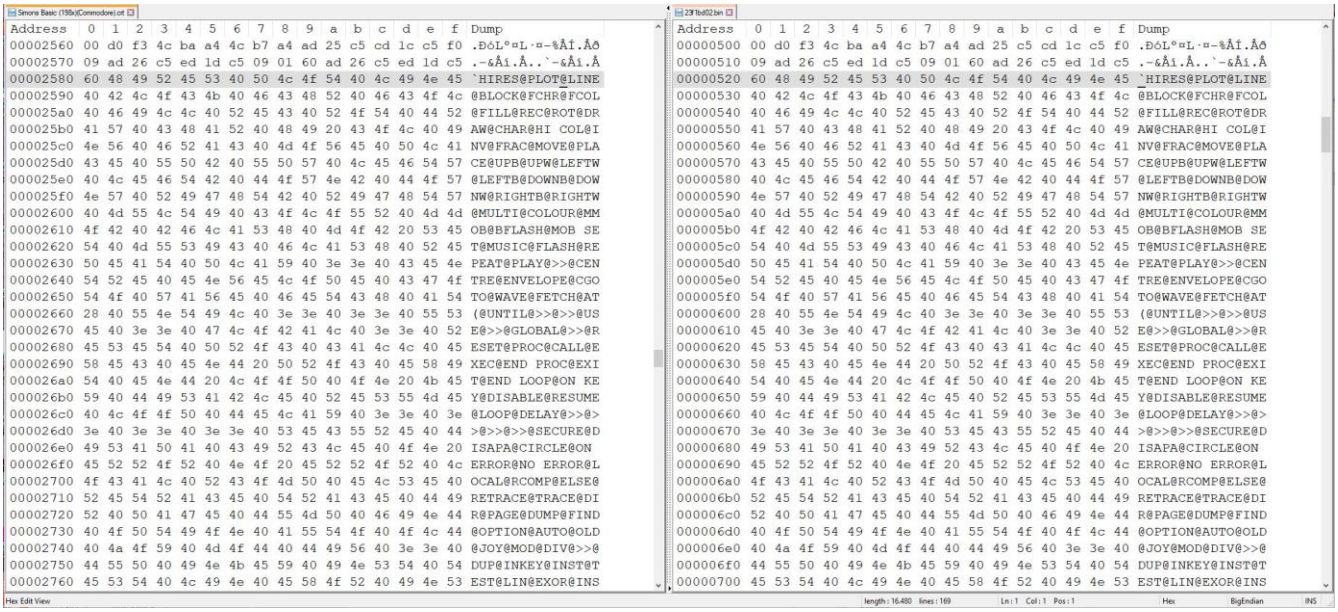


Figure 6.54: Comparison of a dump with a Cart dump (CRT)

Note that a CRT contains an “header” of 0x60 (96) bytes (not shown in the picture). The information to be compared is found at start address 0x0060 or 0x2560.

6.7.16 Dumping Commodore VC20 Cartridges

VC20 cartridges with a capacity of 8 to 32 kByte can be dumped. The cartridges can be located in four different memory locations. These are selected with a jumper (BLK5, BLK3, BLK2, BLK1). Almost all games, with the exception of the Scott Adams text adventures, are in block 5 (BLK5). 16K cartridges, use usually block 3 (BLK3) (a few games use block 1 (BLK1)). The setting 2764 (8k x 8 EPROM) is used for dumping the cartridge. Each block must be dumped individually (set one(!) jumper only).

The label on the cartridge points to the front when dumping.

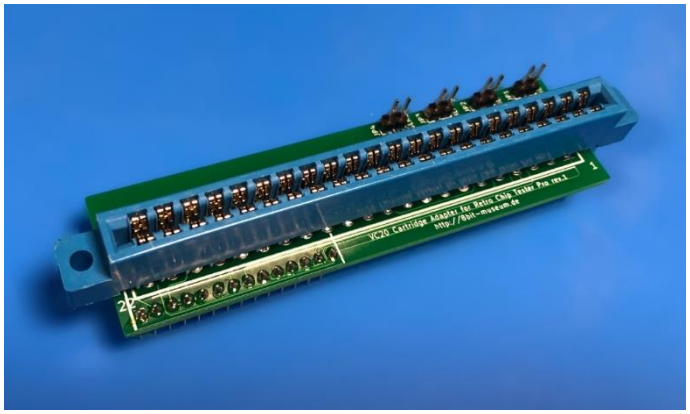


Figure 6.55: VC20 Cartridge Adapter

Figure 6.56: VC20 Cartridge Adapter

6.7.17 Dumping Atari VCS/2600 Cartridges

VCS/2600 cartridges with a capacity of 2 kByte, 4 kByte and 8 kByte can be dumped. Dumping 16 kByte cartridges is not possible with every cartridge, as those with two ROMs may have a high current consumption that the tester cannot provide. In this case, each dump shows a different CRC32 (i.e. there were errors when reading out). You can try again using the DC barrel jack and a power supply that can provide 7.5-9V at 1A. If it reads different each time that cart cannot be dumped with the RCT. In that case you can open the cart, remove the ROM and read it as a common mask ROM using the 23xx setting.

All cartridges (2, 4, 8, 16 kByte):

All cartridges are dumped via the corresponding menu item "VCS/2600 Cart". The jumper **must** be closed with a 4, 8 and 16 kByte cartridge (it **may** also be closed with a 2k cartridge). After selection, the cartridge size must be selected. Standard bank switching (0xfff8/0xfff9 or 0xfff6/0xfff7/0xfff8/0xfff9) is supported.

2 kByte and 4 kByte cartridges:

The adapter can also be used to dump 2 and 4 kByte cartridges with any other EPROM reader. Here you can switch between 2k and 4k using the jumper. The setting 2716 (2k x 8) or 2732 (4k x 8) is then used for dumping.

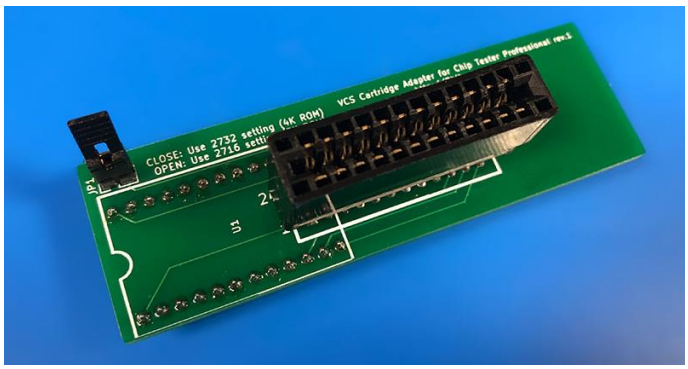


Figure 6.57: VCS/2600 Cartridge Adapter

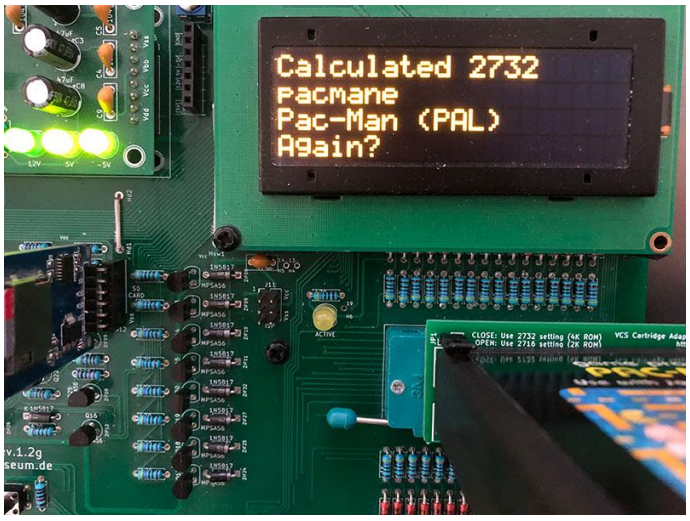


Figure 6.58: Alignment of the cartridges

The label on the cartridge points in the direction of the display when it is dumped. If the adapter has a second jumper (JP2), this must always be set to "STD".

6.8 Testing PLA and FPLA

PLAs and FPLAs can also be tested via a small detour. The procedure is identical to reading out ROMs (see sec. 6.5.6 and sec.6.7).

A ROM(!)-Definition file has been created for the PLA part number Signetics 82S100, which is used in some Commodore computers. The definition file looks like this:

```
; Definition for Signetics 82S100 (64k x 8, ROM)
; Status: working, confirmed
; used to check Commodore PLAs, e.g. 906114-01, 251641-02, etc
; Rename to "customrom1.txt"
;
name:  82S100 (64k x 8)
dip:   28
size:  65536
flags: 0

a:     9,8,7,6,5,4,3,2,27,26,25,24,23,22,21,20
dout:  18,17,16,15,13,12,11,10
cs:    19
pcs:
we:
oe:

gnd:   14
vcc:   28
```

When the CRC32 is known, the recognized PLA is displayed directly, otherwise only the CRC32. This can then be included in the external CRC32 database.

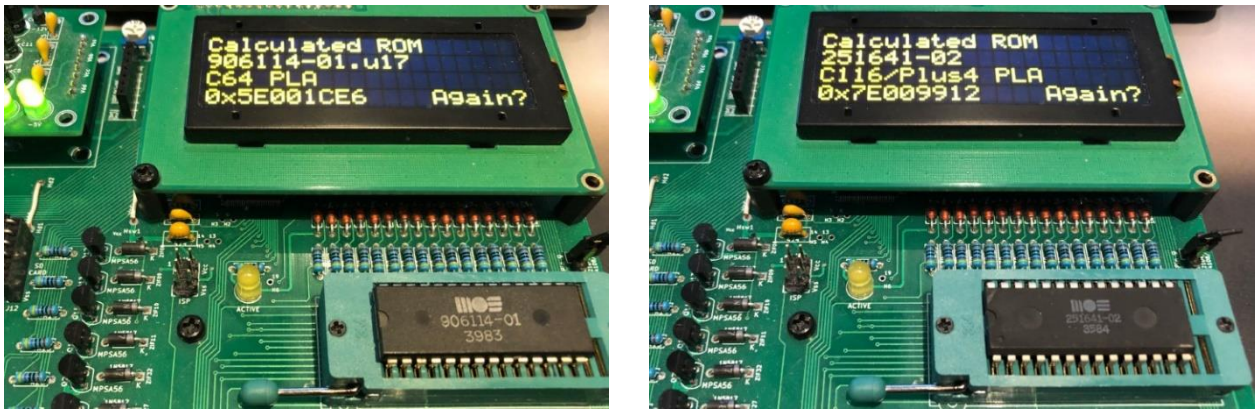


Figure 6.59: Identification of PLAs (left C64 PLA, right C116/Plus4 PLA)

A corresponding binary file can also be created that you can use to replace the PLA with an EPROM. A suitable adapter must be used for the 27512 (Input In to Address line An, Output Fn to Data line Dn).

However, this is not the recommended method and it relies heavily on the brand and speed of EPROM or it will not work. Search and you can find more info on the internet about replacing a Commodore PLA with an EPROM. There are better methods such as using 2 GALs or a CPLD or buying one of the many available Commodore PLA reproductions, but this is left up to you to research. The PLA function in the RCT is purely so you can test that your PLA is good, not to reproduce it in another way.

6.8.1 Notes on Commodore PLAs

There is already a PLA2EPROM adapter that is used on the C64 (or CBM computers). Here the assignment is not 1:1, as mentioned above.

The N82S100 data sheet shows its pin assignment, e.g. Pin 2 = I7, Pin 3 = I6, Pin 9 = I0, Pin 10 = F7.

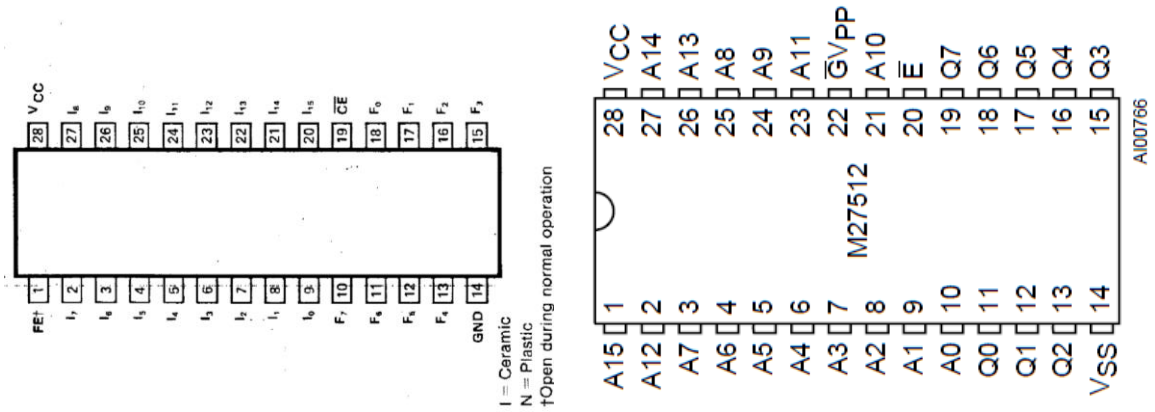


Figure 6.60: Pinout N82S100 PLA and 27512 EPROM

From a logical point of view it make sense to map these accordingly to the address and data lines:

$I_n \rightarrow A_n$ and $F_n \rightarrow D_n$

This is what the custom rom definition for the N82S100 does.

Unfortunately, this is inconvenient when replacing the PLA with a 27512. With wiring technology, the wiring would be too complex and a circuit board would be required. So, whenever an input (I) matched an address line (A), these were connected and the rest was wired. That way it was possible to create an adapter with only two sockets.

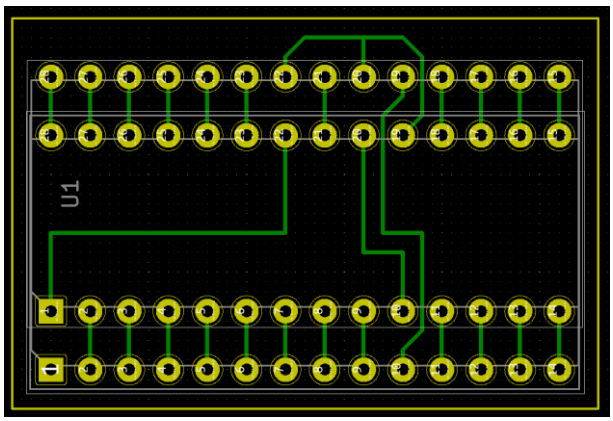


Figure 6.61: PLA2EPROM adapter socket

The problem with this is that the pinout is different for both chips. This adapter connects e.g. I1 (pin 8) with A2 (pin 8), I7 (pin 2) with A12 (pin 2). This simplifies the wiring, but creates the problem that a special dump for the EPROM must be created using this adapter.

The PLA2EPROM Custom ROM Definition creates a dump exactly in this way, so that it can be used directly for the 27512 EPROM. Please note: This is just a quick'n'dirty solution. Such a replacement *can* work, but does not have to be.

The following ROM file creates a dump specifically for this adapter that can be used in a 27512:

```
; Definition for PLA2EPROM dumps (64k x 8, ROM)
; Status: working, confirmed
; these dumps can be used in a 27512 as PLA replacement
; PLA-to-EPROM adapter is required
; Rename to "customrom1.txt"
;
name:  PLA2EPROM (64k x 8)
dip:   28
size:  65536
flags: 0

a:     20,9,8,7,6,5,4,3,25,24,21,23,2,26,27,22

din:   11,12,13,15,16,17,18,10
dout:  11,12,13,15,16,17,18,10
cs:    19
pcs:
we:
oe:

gnd:   14
vcc:   28
```

6.9 Reading the "electronic signature" of an EPROM

The "electronic signature" of an EPROM (2764 to 27512) can be read.

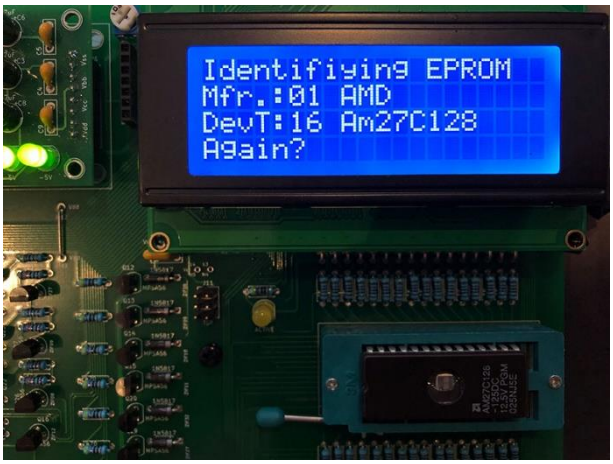


Figure 6.62: Identification of an Am27C128 from AMD (manufacturer 0x01, device 0x16)

A simple adapter must be used for EPROMs with 32 pins, since the required voltage at the corresponding pin cannot be provided directly by the RCT:

A 28-pin and 32-pin socket are plugged together as shown in the following pictures. Pins 1, 2 and 3 as well as 29, 30 and 31 are connected to each other on the 32-pin socket. Pin 28 of the 28-pin socket is connected to pin 32 of the 32-pin socket, which must not have a connection to the pin above it.

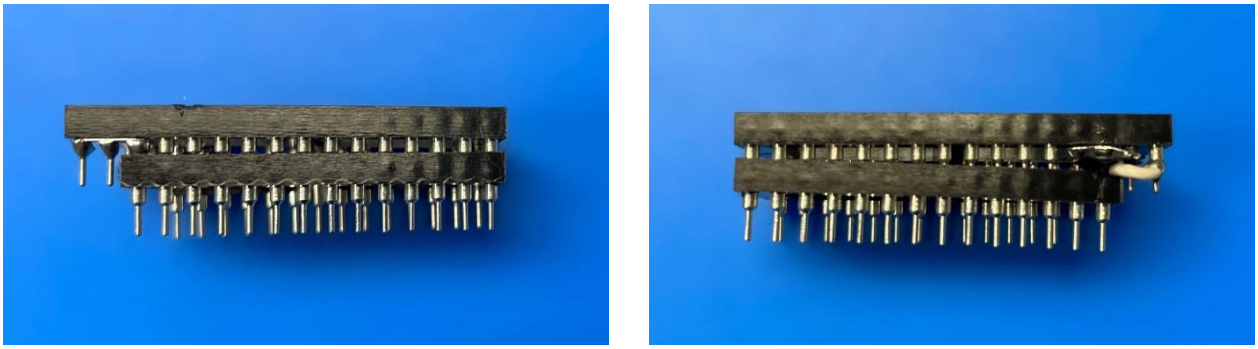


Figure 6.63: Adapter for reading the signature of 32-pin EPROMs

The signature is then read with the setting "ES 27256-27512".

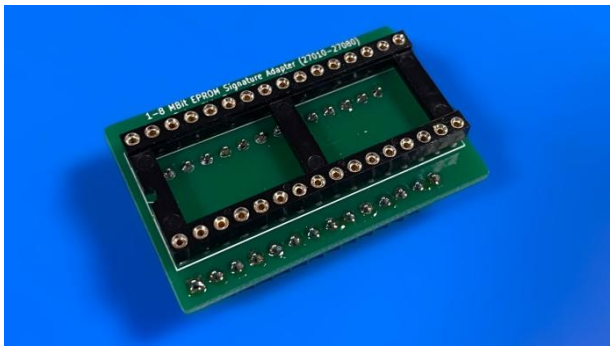


Figure 6.64: Adapter for reading the signature of 32-pin EPROMs

Examples:

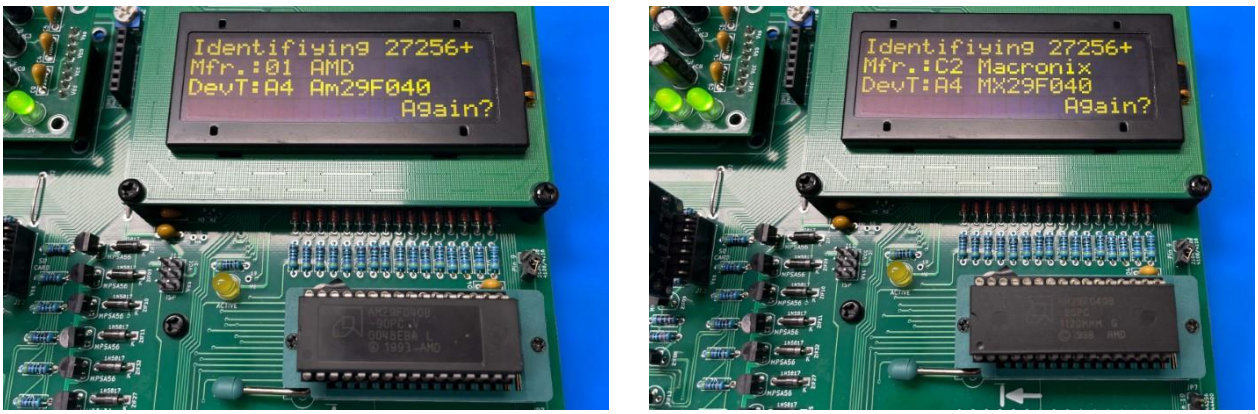


Figure 6.65: Original Am29F040 (left) and "Fake" Am29F040 (right)

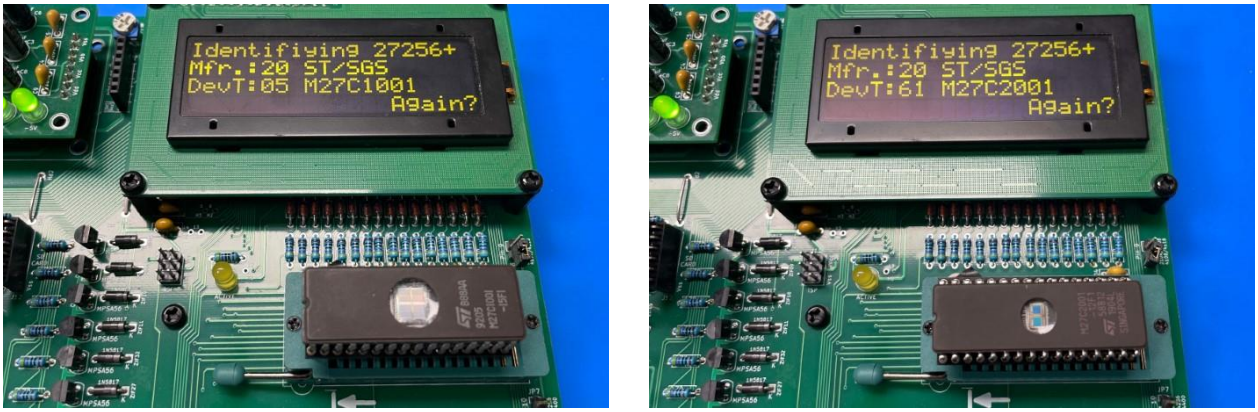


Figure 6.66: Original M27C1001 (left) and Original M27C2001 (right)

The manufacturer and type signatures are not necessarily unique. In some cases, the same signature was used by several manufacturers and the same signature can also be used by several types. In the case of identical signatures, the more frequently used one is displayed.

The signature of EEPROMs can also be read. It should be noted that the content can possibly be erased.

6.10 Testing logic ICs (74xx, 40xx, and others)

Many logic chips from the 74xx and 40xx series can be tested. The logic family plays (almost) no role because the tester tests the logic, not the technical specifications, and the power supply is 5V. Thus, the following variants from the 74xxx series can be tested: Standard TTL (no letter), L, H, S, LS, AS, ALS, F, HC, HCT, AC, ACT.[6]

The Logic Testers are called via the menu:

Figure 6.67: Calling the logic tester

Then the chip to be tested is selected.

Figure 6.68: Selecting a 7400

If the chip is OK, "`Chip OK`" is displayed, otherwise "`Chip FAILED`".

The errors found are displayed for 2 seconds during the test. The display is as follows:

0 or 1 = 0V or 5V signal applied to an input.
L or H = expected signal 0V or 5V at an output.
G or V = power supply (GND or Vcc)

A "H with an arrow down" means that a H was read, but an L was expected.
A "L with an arrow up" means that an L was read, but a H was expected.

Figure 6.69: Defective "4-bit binary counter (74169)"

In the above example, a H was read at pin 11, 13, but an L was expected in each case, and an L was read at pin 12, but a H was expected.

Please be sure to follow the instructions in sec. 6.10.4!

A test is repeated multiple times when started with long press on OK (see configuration). Not all tests support this. FUNC (or SELECT/JUMP) aborts prematurely.

[6] There is one restriction: Due to the current limitation to approx. 2-3mA, components with a particularly high current consumption at one input cannot necessarily be reliably tested (usually chips with parallel-connected flip-flops, e.g. for counters such as 7490 (standard TTL)).

6.10.1 Chips with special power supply

Some chips deviate from the standard scheme with regard to the power supply.

If a "Pos x" is displayed after the type designation, the chip must be inserted at "Position x" in the socket. The chips are normally used aligned upwards (corresponds to "Pos 1"). Since some chips do not expect the power supply to be on the usual pins (e.g. 7475, 7476, 7490, 7492, 7493), this chip must then be used offset ("Pos 4" means that three rows must remain free).

Figure 6.70: Chip inserted at position 4 (7490)

If “Rev x” is displayed after the type designation, the chip must be rotated 180 degrees and inserted into the socket at “Position x”. However, only very few ICs are affected by this.

Figure 6.71: Chip inserted rotated 180 degrees at position 1 (DL2416)

6.10.2 Pin assignment of logic chips

6.10.2.1 Pin assignment of 74xx and 54xx logic chips

The pin assignment is usually identical for the different specifications:

- 54xx - Military/Airspace Grade Device
- 64xx - Industrial Device
- 74xx - Commercial Grade
- 75xx - Interface Device

This means that a 5400 with the 7400 setting can usually be tested without any problems, provided the package type is identical. Caution is advised if e.g. a 54xx chip in a different package, e.g. "W" (ceramic flat pack) is tested (see example "7454" below).

6.10.2.2 There are different versions of some logic chips

The 7454 and 74LS54 have a different function and thus also pin assignment:

- The 7454 contains four AND gates with two inputs each, which are linked with a NOR.
- The 74LS54 contains two AND gates with two inputs and two AND gates with three inputs that are linked with a NOR.

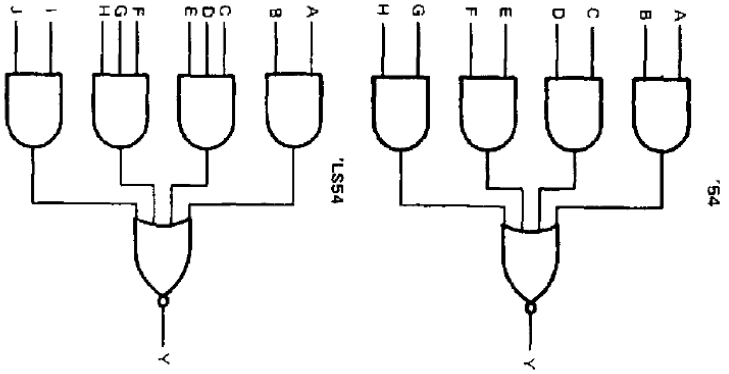


Figure 6.72: 74LS54 and 7454

The chip tester allows e.g. the selection of “7454” or “7454 (LS)”.

Since the display can only show a limited number of characters, the data sheet should be consulted for older ICs. The following ICs (and a few more) are shown separately:

Selection	ICs to be tested
7451	7451, H 51, S 51 only
7451(LS)	L 51, LS 51 and all others (HC, HCT etc.), if exist
7453	7453 only
7453(H)	H 53 only
7454	7454 only
7454(H)	H 54 only
7454(LS)	L 54, LS 54 and all others (HC, HCT etc.), if exist
7455(H)	H 55 only
7455(LS)	L 55, LS 55 and all others (HC, HCT etc.), if exist
7471(H)	H 71 only
7471(L)	L 71 only

Selection	ICs to be tested
7476	76 only
7476(LS)	LS 76 only
7478(H)	H 78 only
7478(L)	L 78 only
7478(LS)	LS 78 and all others (HC, HCT etc.), if exist
7485	7485 and all others (HC, HCT etc.), if exist
7485(C)	C 85 only
7490	7490 and all others (HC, HCT etc.), if exist
7490(C)	C 90 only
7493	7493 and all others (HC, HCT etc.), if exist
7493(C)	C 93 only
7493(L)	L 93 only
7495	7495, LS 95 and all others (HC, HCT etc.), if exist
7495(L)	L 95 only

6.10.2.3 Different pin assignments for different package types:

Chips with the package code "D" (DIP), "J" (CDIP = ceramic DIP) and "N" (plastic DIP) usually have the same pin assignment

Chips with the package code "W" (ceramic flat pack) may have different pin assignments and should only be checked after looking at the data sheet.

Example 7454N/7454J and 7454W:

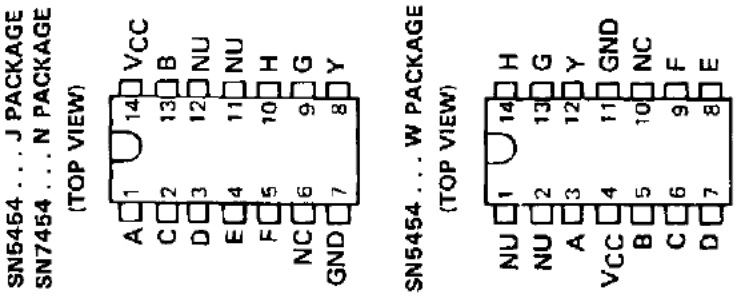


Figure 6.73: 7454 J/N and 7454 W

In contrast, the 74LS54 has the same pin assignment in all versions:

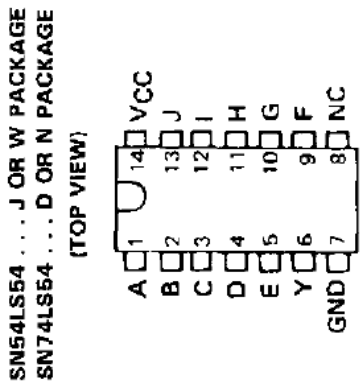


Figure 6.74: 74LS54 D/J/N/W

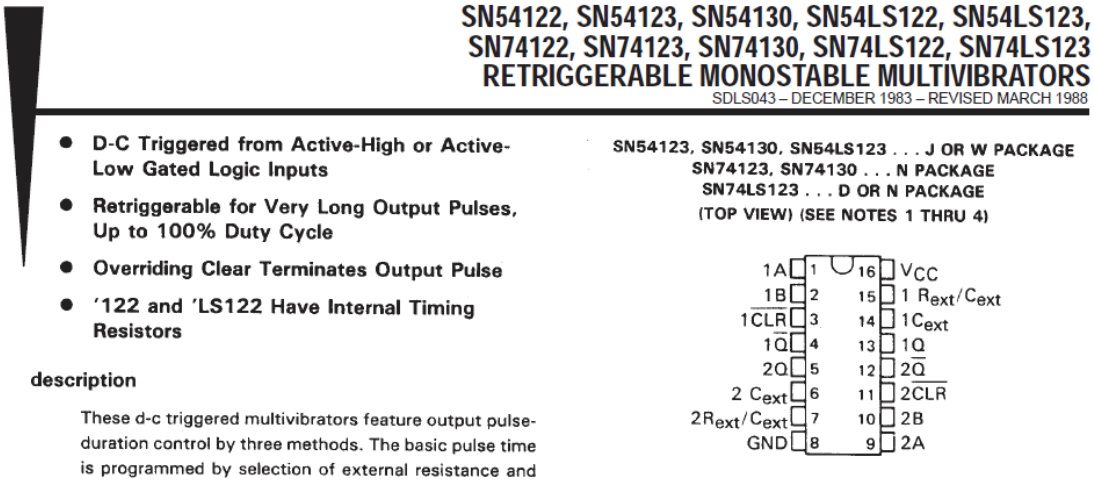
The Chip Tester supports the common pin assignments of the DIP package types.

6.10.2.4 Different functionality but with same type number

Some logic ICs seems to be a certain type but aren't.

For example:

The **SN74130** is a "Retriggerable Monostable Multivibrator".



SN54122, SN54123, SN54130, SN54LS122, SN54LS123,
SN74122, SN74123, SN74130, SN74LS122, SN74LS123
RETRIGGERABLE MONOSTABLE MULTIVIBRATORS
SDLS043 – DECEMBER 1983 – REVISED MARCH 1988

- D-C Triggered from Active-High or Active-Low Gated Logic Inputs
- Retriggerable for Very Long Output Pulses, Up to 100% Duty Cycle
- Overriding Clear Terminates Output Pulse
- '122 and 'LS122 Have Internal Timing Resistors

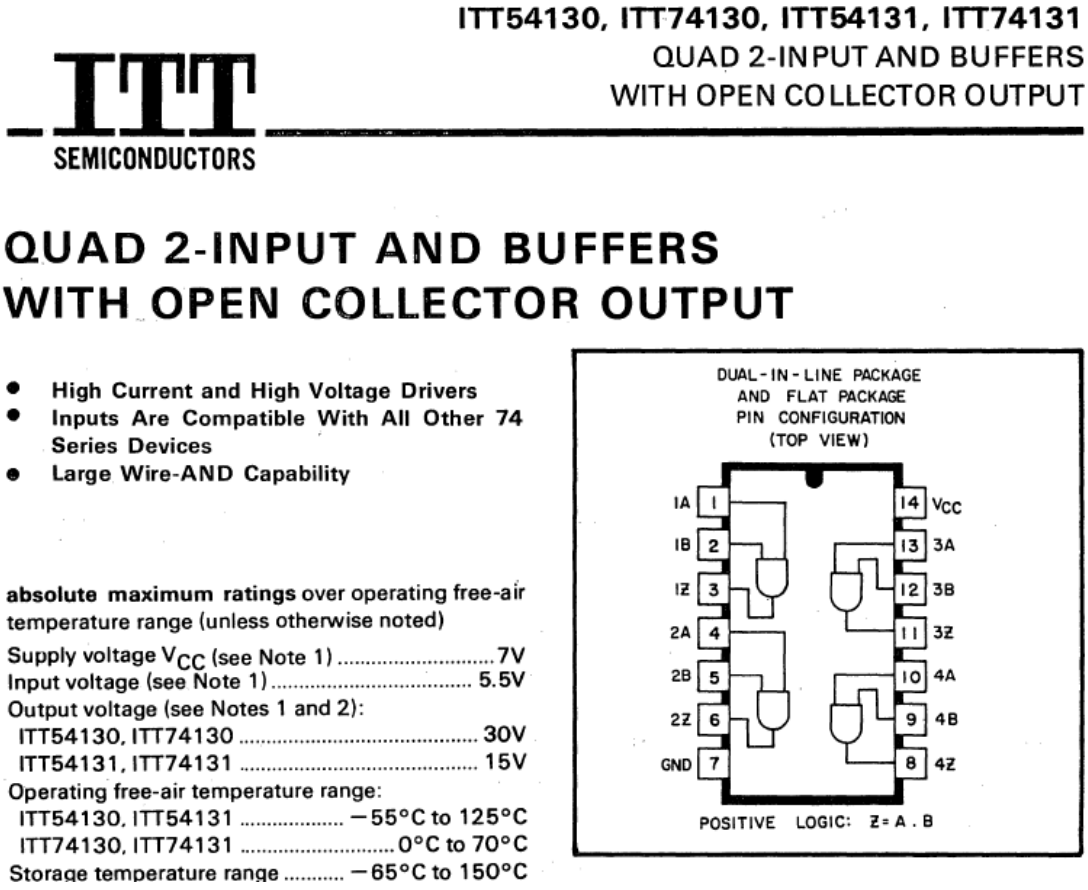
SN54123, SN54130, SN54LS123 . . . J OR W PACKAGE
SN74123, SN74130 . . . N PACKAGE
SN74LS123 . . . D OR N PACKAGE
(TOP VIEW) (SEE NOTES 1 THRU 4)

description

These d-c triggered multivibrators feature output pulse-duration control by three methods. The basic pulse time is programmed by selection of external resistance and

Figure 6.75: SN74130

The **ITT74130** is a "Quad 2-Input AND Buffers with Open Collector Output".



ITT54130, ITT74130, ITT54131, ITT74131
QUAD 2-INPUT AND BUFFERS
WITH OPEN COLLECTOR OUTPUT

ITT SEMICONDUCTORS

QUAD 2-INPUT AND BUFFERS WITH OPEN COLLECTOR OUTPUT

- High Current and High Voltage Drivers
- Inputs Are Compatible With All Other 74 Series Devices
- Large Wire-AND Capability

DUAL-IN-LINE PACKAGE
AND FLAT PACKAGE
PIN CONFIGURATION
(TOP VIEW)

POSITIVE LOGIC: Z = A . B

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage V_{CC} (see Note 1) 7V
Input voltage (see Note 1) 5.5V
Output voltage (see Notes 1 and 2):
ITT54130, ITT74130 ..... 30V
ITT54131, ITT74131 ..... 15V
Operating free-air temperature range:
ITT54130, ITT54131 ..... −55°C to 125°C
ITT74130, ITT74131 ..... 0°C to 70°C
Storage temperature range ..... −65°C to 150°C

Figure 6.76: ITT74130

So when a test fails, please check if the IC has the expected function.

There are also functionally different types of the **74LS47**. With the SN74LS47 from Texas Instruments, the digit "6" is generated without a "tail":

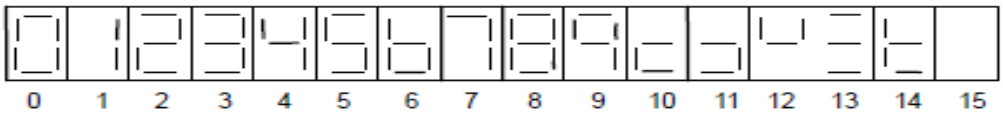


Figure 6.77: SN74LS47

On the other hand, with an SN74LS247, the number “6” is generated with a “tail”:

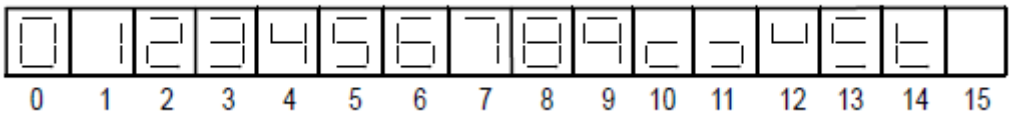


Figure 6.78: SN74LS247

However, there are some manufacturers where the 74LS47 also outputs a "tail" at the digit "6". Of course, the test "7447" fails in this case. These ICs must be tested with the 74247 setting.

6.10.3 Testing Russian Chips

Russian chips are displayed using transliteration according to ISO 9: 1995 and in Cyrillic script.

Characters on the chip

А	Б	к	Ц	Д	Е	Ф	г	г	И	К	Л	М	Н	л	О	П	Р	С	Т	у	В	у	Х	И	Ж
A	B	C	C	D	E	F	G	H	I	K	L	M	N	N	O	P	R	S	T	U	V	W	X	Y	Z

Characters on the display

The TTL families are marked according to the following scheme:

	Soviet Union		**Czechoslovakia**			**Poland**			**East Germany**		
	5400	**7400**	**5400**	**7400**	**8400**	**5400**	**6400**	**7400**	**6400**	**7400**	**8400**
74	133	К155	МН54	МН74	МН84	UCA54	UCA64	UCY74		D1	E1
74L	134 136	КР134 К158									
74H	130	К131					UCA64H	UCY74H		D2	E2
74S	530	КР531	МН54S	МН74S	МН84S			UCY74S		DS	
74LS	533	К555						UCY74LS		DL...D	DL...DG
74AS	1530	КР1530									
74AL	1533	КР1533	МН54ALS	МН74ALS							
74F	1531	КР1531									
74HC	1564	КР1564									
74HCT	5564								U74HCT...DK		
74AC	1554	КР1554									
74ACT	1594	КР1594									
74LVC	5574										
74VHC	5584										

Source: Wikipedia

6.10.4 Testing old bipolar logic chips and S-TTL/H-TTL

As of board version 1.2i, the protective resistors are labeled with 470 ohms as standard. The mentioned 7490 can be tested without any problems.

When testing very old logic chips of the 74xx series it can happen that a chip is tested as defective, although it actually works correctly. The same applies to ICs of the S-TTL and H-TTL families. This occurs very rarely, but it is possible, so this problem is briefly described here.

The chip tester is designed in such a way that the I/O ports of the ATmega2560 are protected by 1k Ohm resistors as current limiters (at 5V, almost 5mA are possible, rather a little less).

Example 7490 (old TTL IC):

From the datasheet of a 7490A it can be seen that this requires an input current of -4.8 mA for a LOW signal at input /CKB (aka /CP1) (simply speaking: in MOS technology, inputs are voltage-controlled, in bipolar technology, current-controlled, i.e. a not inconsiderable current is required for switching). As a comparison to the more modern LS series: The 74LS90 has a value of -1.6mA.

PARAMETER ¶		TEST CONDITIONS†	'90A			'92A			'93A			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	MIN	TYP‡	MAX	
I_{IH}	High-level input current	Any reset			40			40			40	μA
		CKA			80			80			80	
		CKB			120			120			80	
I_{IL}	Low-level input current	Any reset			−1.6			−1.6			−1.6	mA
		CKA			−3.2			−3.2			−3.2	
		CKB			−4.8			−4.8			−3.2	

Figure 6.79: Datasheet 7490A

The required current is just about the range that the Chip Tester can supply as input current. With the military variants 5490 or 8490, the required input current can be a little higher.

The following behavior occurred with an 8490: The 3-bit counter counted correctly 0→1→2→3 with every 1→0 transition from /CP1, with the subsequent transition from 1→0 no further counting (i.e. the change from Q1/Q2→L and Q3→H). After /CP1 was connected to Vss with a 470 Ohm resistor, the chip counted without any problems. No error occurred on /CP0.

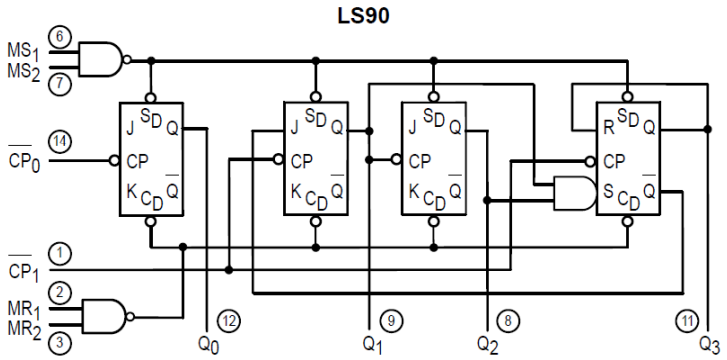


Figure 6.80: Logic diagram 7490A

The logic diagram of the 7490 shows that /CP1 controls two gates and therefore an increased current is necessary.

Figure 6.81: Resistor at /CP1

Usually, maximum values are given in the datasheets, i.e. the typical value should be lower. However, due to the age of the chip (the component mentioned above was from 1974), an increased power consumption may be necessary. If you were to compare the power consumption with the specification, the chip would probably have to be sorted out as "defective".

When very old TTL chips from the 1970s (no L or LS) are to be tested, the datasheet should be reviewed beforehand. If the current consumption of an input (mostly with counters, such as the 7490) is >2.5mA, it may be necessary to use a pulldown e.g. 470 - 1k Ohm. For this, the resistor is inserted between Vss and /CP1 as shown in the figure above. Alternatively, the resistor can also be attached to /CP1 and connected with a wire to Vss on the connector J4 located between the DC/DC module and the LCD.

The “pulldown” is at most a “workaround”. The better choice is to reduce the value of the protective resistors.

Example 74118 / 74119 (old TTL IC):

With the 74118 and 74119 (Hex Set-Reset Latch) the common reset input requires an increased input current of approx. 10mA.

L-Eingangsstrom:			
U_S = 5,25 V, U_{IL} = 0,4 V			
an $\overline{S}_1$ bis $\overline{S}_6$ und $\overline{R}_1$ bis $\overline{R}_6$	$-I_{IL}$	1,6	mA
an $\overline{R}$	$-I_{IL}$	9,6	mA

Figure 6.82: Datasheet 74118 / 74119

A pulldown of 470 Ohms must also be used between R (pin 9) and Vss (pin 8) or between R (pin 13) and Vss (pin 12). The 9.6mA are so high that a resistor may be necessary even with the change described below (470 Ohm protection).

Alternatively, a resistor can be temporarily connected in parallel to the protection resistors:

Figure 6.83: 1k ohm resistor in parallel with the 470 Ohm protection resistor

The “pulldown” is at most a “workaround”. The better choice is to reduce the value of the protective resistors.

Example 74H108 (old TTL IC):

With the 741H108, the common clock input requires a high input current of almost 10mA.

Here, for example, a 470 Ohm resistor can be set between ground (pin 7) and CLK (pin 9).

Alternatively, a resistor can be temporarily connected in parallel to the protective resistors:

Well-known ICs with high power consumption:

Type	Description	Comment
7490	CLK2 = 4.8mA	testable with 470 Ohm
7492	CLK2 = 4.8mA	testable with 470 Ohm
7493	CLK2 = 4.8mA	testable with 470 Ohm
7494	Preset 1 / 2 = 6.4mA	testable with 470 Ohm (*)
7496	PE = 8mA	testable with 470 Ohm
74H101	CLK = 4.8 mA	testable with 470 Ohm
74H102	CLK = 4.8 mA	testable with 470 Ohm
74H103	CLK = 4.8 mA	testable with 470 Ohm
74H104	CLK = 4.8 mA	testable with 470 Ohm
74H108	CLK = 9.6 mA	testable with < 470 Ohm
74118	R = 9.6 mA	testable with < 470 Ohm
74119	R = 9.6 mA	testable with < 470 Ohm
74S181	S = 8mA, C = 10mA	testable with < 470 Ohm
74S182	G1 = 16mA, G0, G2 = 14.4mA	not testable with resistance
74196	CLK2 = 6.4mA	testable with 470 Ohm (*)
DM8530	CLK2 = 4.8mA	testable with 470 Ohm
DM8532	CLK2 = 4.8mA	testable with 470 Ohm
DM8533	CLK2 = 4.8mA	testable with 470 Ohm

(*) this value is borderline, due to aging effects it may no longer be possible to test with the mentioned resistance

Other logic testers, e.g. the well-known TL866 (see https://www.eevblog.com/forum/testgear/tl866-testing-ttl-7493/), have the same problem but the Retro Chip Tester can at least be modified for these (few) problematic components.

For those who like to experiment / risk takers:

The I/O ports of the ATmega are so highly secured that the tester is guaranteed to survive short circuits in any case. With 470 Ohm at all I/O ports, the protection is usually still sufficiently good.

If you want to test the remaining 74xx ICs that require more than approx. 8mA input current at one input, you have the following options:

1st step: Further reduction of protective resistances:

The I/O ports of the ATmega can withstand a maximum of 20mA, i.e. at 5V the protective resistor may have 250 Ohms (220 Ohms are also ok). The Zener diodes can withstand a maximum of 500mW, i.e. approx. 40mA at 12V. The protective resistance for the Zener diodes should be at least (12-5) V / 40mA = 175 Ohm. The load current is negligible here. So, the recommended minimum is 220 Ohms. That should be enough for all old bipolar ICs.

When you are careful, you can alternatively swap all protective resistors - except those for pins 8, 9, 27 and 28 – for 220 Ohms. Pins 8, 9, 27 and 28 can have 12V and should then be left at 470 Ohms.

2nd step: Removal of protective resistors (not recommended):

If the protective resistors interfere with your own experiments, these can be removed (except for the protective resistors on pins 8, 9, 27 and 28). This is not recommended and is very risky. If an IC that requires a 12V power supply should have an internal short circuit, the I/O ports would be exposed to it without protection.

6.10.5 Use of the Breakout Adapter

With the 74182 (Look-Ahead Carry Generations), the shared enable inputs require a high input current of 14-16mA. As a rule, this IC cannot be tested directly. The method presented here is also an example of the use of the Breakout Adapter.

I_{IL}	Low-level input current	C_n input	V_{CC} = MAX, V_I = 0.4 V	−3.2	−3.2	mA
		$\overline{P}3$ input		−4.8	−4.8	
		$\overline{P}2$ input		−6.4	−6.4	
		$\overline{P}0$, $\overline{P}1$, or $\overline{G}3$ input		−8	−8	
		$\overline{G}0$ or $\overline{G}2$ input		−14.4	−14.4	
		$\overline{G}1$ input		−16	−16	

Figure 6.84: Datasheet 74182

The 74LS182, on the other hand, can be tested without any problems:

I_{IL}	C_n $\overline{G}0$, $\overline{G}2$ $\overline{G}3$, $\overline{P}0$, $\overline{P}1$ $\overline{P}2$ $\overline{P}3$ $\overline{G}1$		−0.4 −2.8 −1.6 −1.2 −0.8 −3.2	mA	V_{IN} = 0.4 V V_{CC} = MAX

Figure 6.85: Datasheet 74LS182

This IC can also be tested the help of the Breakout Adapter. A 74LS07 is used for this, a 6x buffer with open collector outputs, which is able to drive up to 40mA. The inputs of the 74182 that require more than 8mA are connected to the output of a buffer. Since the 74LS07 has open collector outputs, pull-up resistors with 4.7k ohms each are used. The input of a buffer is connected to the ZIF socket of the RCT.

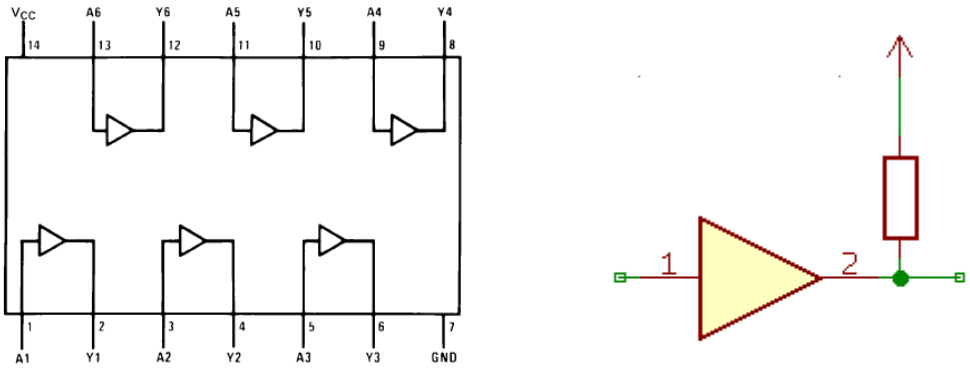


Figure 6.86: 74LS07 (left), circuit (right)

The 74LS07 is placed on a breadboard. The supply voltage is provided by the RCT. The outputs of the 74LS07 are each provided with a pull-up resistor:

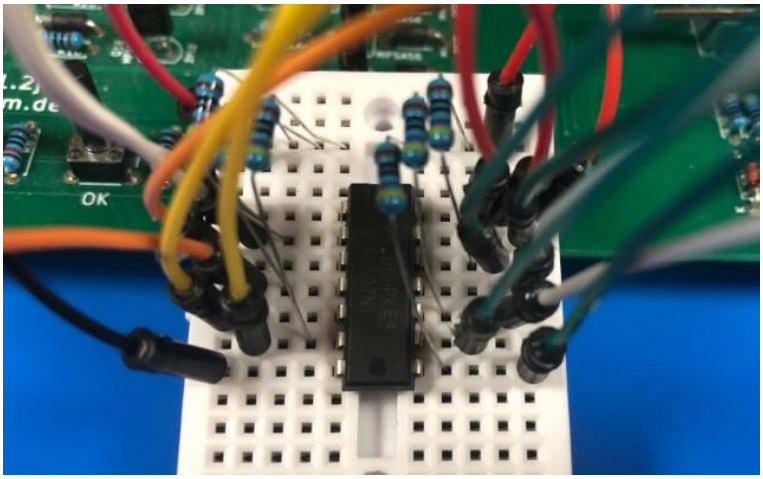


Figure 6.87: 74LS07 on the breadboard with pull-up resistors

On the Breakout Adapter, all pins except for pins 1, 2, 3, 4, 5, 14 are bridged with a jumper.

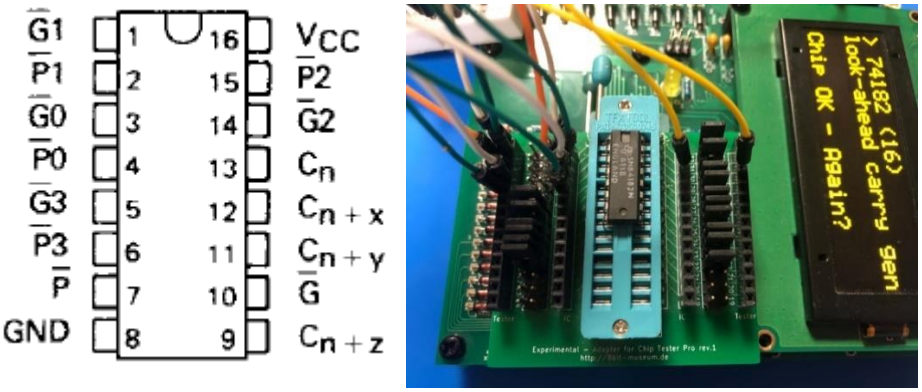


Figure 6.88: 74182 „Look-Ahead Carry Generations”

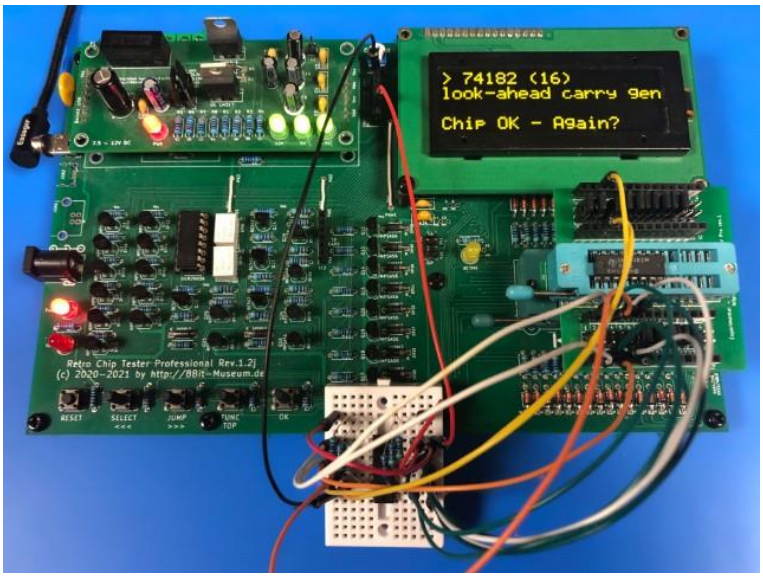


Figure 6.89: Complete construction

The inputs of the 74LS07 are then connected to the “socket” connectors on the breakout adapter, the outputs of the 74LS07 to the “IC” connectors on the breakout adapter.

6.10.6 Testing of Multivibrators (74121, 74122, 74123 et. al.)

The multivibrators 74121, 74122, 74123, 74130, 74221, 74422, 74423 and 8T22 require some external components (at least one resistor and capacitor) to function correctly, which are provided by an adapter.

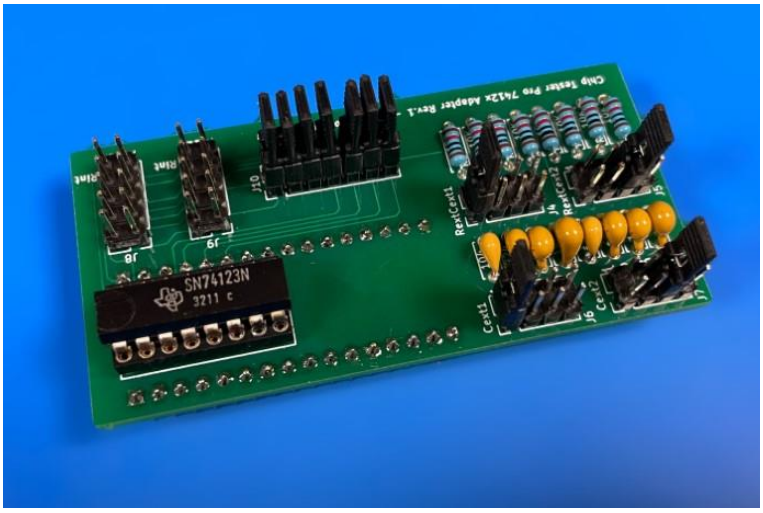


Figure 6.90: Adapter in 74123/74130/74221/74423 configuration

The adapter has three pin headers:

- 2x5 Pins for 74121,
- 2x5 Pins for 74122, 74422 and 8T22,
- 2x7 Pins for 74123, 74130, 74221 and 74423.

Appropriate Jumpers must be set to select the IC to be tested. One jumper - each for 74121 and 74122/74422) - is marked with "Rint". This Jumper can be set (if required) when the internal resistance is to be used.

Four pin headers are labeled "RextCext" and "Cext". Positions 1 through 4 are used to select one of four resistor/capacitor combinations. "RextCext2" and "Cext2" is only used on the 74123, 74130, 74221 and 74423 ICs.

Figure 6.91: 74123 with Pos 1 and Pos 3 set

The figure shows a 74123 with the first multivibrator tested in position 1 (10k Ohms and 100nF) and the second multivibrator in position 3 (10k Ohms and 470nF).

At the end of the test, no "PASS" or "FAIL" is displayed but the measured time in microseconds. The following table shows which values are to be expected for the ICs.

Typ	Config		LS	HC	HCT
74121	A	1: 750 2: 1480 3: 3520 4: 6660			
74121 with Rint	A	1: 120 2: 240 3: 580 4: 1080			
74122	B	1: 300 2: 600 3: 1160 4: 2270	1: 350 2: 690 3: 1460 4: 2850		
74122 with Rint	B	1: 150 2: 300 3: 590 4: 1140			
74123	C	1: 270 2: 540 3: 1000 4: 1970	1: 370 2: 740 3: 1530 4: 2870		1: 450 2: 910 3: 1570 4: 3480
74130	C	?			
74221	C		1: 790 2: 1570 3: 3520 4: 6920	1: 1160 2: 2320 3: 4890 4: 10090	
74422	B		?	?	
74423	C		1: 360 2: 720 3: 1460 4: 2760	1: 440 2: 880 3: 1660 4: 3370	
8T22 (no Rint)	B	?			

All data in the table are in microseconds. The values are approximate (measured) values and depend on the tolerances of the resistors and capacitors used. A, B, C indicate the configuration.

The following values are used by default for the individual positions:

- Position 1: 10k Ohm / 100nF
- Position 2: 10k Ohm / 220nF
- Position 3: 10k Ohm / 470nF
- Position 4: 10k Ohm / 1000nF

Other values on the adapter can also be used if required. The corresponding data sheet must then be consulted for the expected result. The resolution of the timer is 10us.

If no adapter is used, the components can also be plugged directly into the ZIF socket.

6.10.7 Testing of Multivibrators (CD4098, CD4528, CD4538)

The multivibrators CD4098, CD4528 and CD4538 require some external components (at least one resistor and capacitor) to function correctly, which are provided by an adapter.

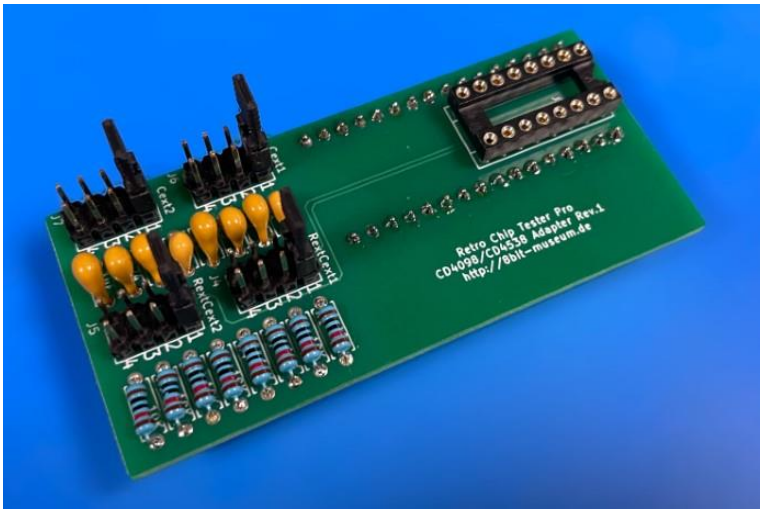


Figure 6.92: Adapter for CD4098, CD4528, CD4538 with Pos 1 set

Four pin headers are labeled "RextCext" and "Cext". Positions 1 through 4 are used to select one of four resistor/capacitor combinations.

At the end of the test, no "PASS" or "FAIL" is displayed but the measured time in microseconds. The following table shows which values are to be expected for the ICs.

Typ	1: 100nF	2: 220nF	3: 470nF	4: 1000nF
CD4098	400	800	1150	2700
CD4528	450	950	1500	3700
MC14528	220	430	1070	2180
SCL4528	300	580	1350	2840
TC4528	310	590	1450	2910
CD4538	700	1500	2400	5800

All data in the table are in microseconds. The values are approximate (measured) values and depend on the tolerances of the resistors and capacitors used.

The following values are used by default for the individual positions:

- Position 1: 10k Ohm / 100nF
- Position 2: 10k Ohm / 220nF
- Position 3: 10k Ohm / 470nF
- Position 4: 10k Ohm / 1000nF

Other values on the adapter can also be used if required. The corresponding data sheet must then be consulted for the expected result. The resolution of the timer is 10us.

If no adapter is used, the components can also be plugged directly into the ZIF socket.

6.10.8 Testing of expandable logic ICs (“expander”)

Expandable ICs can be tested, but since the special expander inputs do not carry a TTL level, tests are only possible to a limited extent. This means that

- the behavior of the expander outputs is not checked,
- the expander inputs may need to be isolated for a test to pass.

The expander outputs are checked like open-collector outputs. So there shouldn't be any restrictions regarding the tests.

The following logic ICs have expander inputs and outputs:

Type	Description	Comment
7423	Inputs	if needed isolate pins 1 and 15
7450	Inputs	if needed isolate pins 11 and 12
7452	Inputs	if needed isolate pin 9
7453	Inputs	if needed isolate pins 11 and 12
74H55	Inputs	if needed isolate pins 5 and 9
7460	Outputs	Expander for 7423, 7450, 7453, 74H55
7461	Outputs	Expander for 7452
7462	Outputs	Expander for 7450, 7453, 74H55

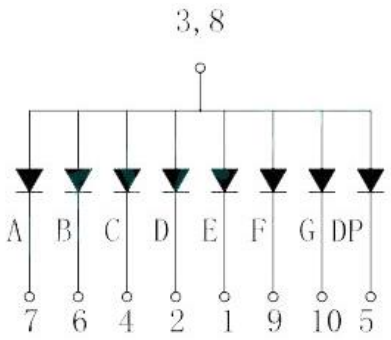
6.11 Testing of other devices

Some other devices can be tested, such as 7-segment displays. These can be found in the “Misc Tests” menu item.

Figure 6.93: Misc Tests

The devices are used in the tester in a similar way to the TTL devices.

Example: One segment with a dot point (DP), common anode (CA) on pins 3 and 8 as circuit diagram and with the text displayed in the menu:




```
LED 1x7 Seg., DP, CA: 3,8
```

If it is unclear which component it is, measure the resistance between pins 3/8, 1/6 or 7/9 (correspondingly for displays consisting of two segments). The corresponding module is identified at 0 Ohm.

6.12 Reading out PALs (Programmable Array Logic)

The 20-pin ICs of the PAL and GAL families have a security bit which is supposed to prevent the chip from being copied. The underlying logic equations can be reconstructed with very little effort if all possible input combinations are iterated and the outputs are logged and then analyzed. In this way a copy can be made that is logically identical to the original.

However, this approach does not work with the following ICs:

- all “registered” PALs (PAL16R4, PAL16R6, PAL16R8 etc.)
- all GALs, that have been configured as "registered" (GAL16V8 etc.)

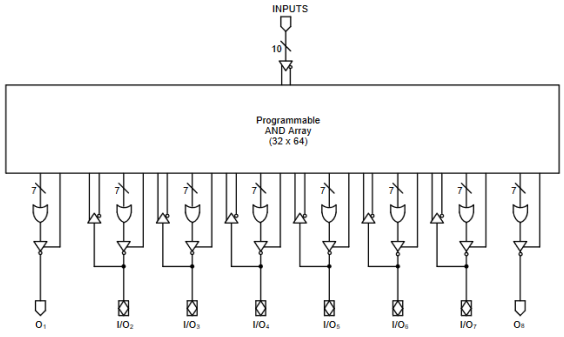
In principle, only pure combinatorial logic ICs can be analyzed in this way, also no pure combinatorial logic ICs that implement latches using combinatorial logic cannot be analyzed either.

A GAL16V8 can be configured as "simple" (combinatorial), "registered" or "complex". Only purely combinatorial configurations can be analyzed. When pin 11 (/OE) is connected to GND and pin 1 (CLK) to a clock source or a write strobe, the GAL is *probably* operated as "registered".

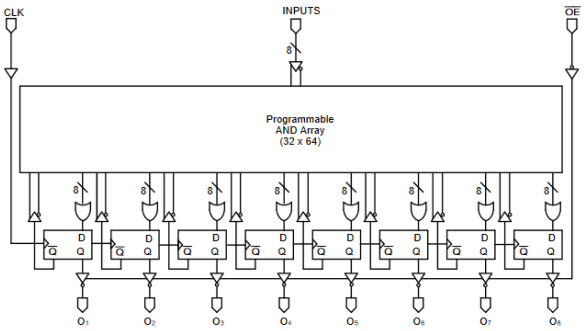
PAL Architectures Emulated by GAL16V8	GAL16V8 Global OLMC Mode
16R8	Registered
16R6	Registered
16R4	Registered
16RP8	Registered
16RP6	Registered
16RP4	Registered
16L8	Complex
16H8	Complex
16P8	Complex
10L8	Simple
12L6	Simple
14L4	Simple
16L2	Simple
10H8	Simple
12H6	Simple
14H4	Simple
16H2	Simple
10P8	Simple
12P6	Simple
14P4	Simple
16P2	Simple

Another restriction is that tri-state outputs cannot be recognized. Additional hardware would be required for this.

The PALs listed as “simple” in the table on the right or the corresponding chips emulated by a GAL16V8 can be read out in a combinatorial manner. The chips marked with "complex" can, under certain circumstances, be read out.



PAL16L8 (“combinatorial”)



PAL16R8 (“registered“)

6.12.1 Preparation

This function can only be used meaningfully together with an SD card that saves the read values as a truth table.

For this purpose, support for SD cards must be activated in the configuration and the micro SD card adapter must be connected to the chip tester. The connection is described in in sec. 8 ("Connecting the SD card adapter").

Reading out a PAL is made possible via the following menu item (“PAL 20pin” respectively “PAL 24pin“):

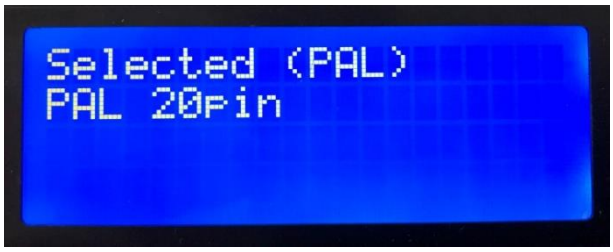


Figure 6.94: Menu “PAL”

Using this feature is similar to reading out a ROM or EPROM, i.e. with activated support:

... **if the OK button is pressed for a long time**, the content of the PAL is written to a file.

... if the OK button is pressed briefly, only a CRC32 is calculated from the determined output values.

The CRC32 value is currently worthless, it is currently used to give the output file on the SD card a unique name. If necessary, it can be used to compare two PALs.

In the next step, the PAL must be defined, it must be determined which pins are used for input and which pins are used for output.

Setting ("OK" pressed briefly)

Setting ("OK" pressed long)

A PAL10H8 or PAL10L8 is preset, which has inputs on pins 1 to 9 and 11 and outputs on pins 12 to 19. GND is on pin 10 and Vss on pin 20.

I0	1		20	Vcc
I1	2		19	O7
I2	3		18	O6
I3	4		17	O5
I4	5		16	O4
I5	6		15	O3
I6	7		14	O2
I7	8		13	O1
I8	9		12	O0
GND	10		11	I9

Figure 6.95: Pre-assigned PAL10H8 or PAL10L8

The pin with the lower number is assigned to the lower-order bit.

The assignment can be changed using the keypad. Pressing SELECT and JUMP at the same time switches between the two groups [IO-] and [GV]. Pressing JUMP switches through the selected group, i.e. [I] = input → [O] = output → [-] = ignored or [G] = GND → [V] = Vss. The next pin is selected with SELECT. OK starts the readout process.

Attention:

A maximum of 8 output pins are supported. If more are selected, only the lower 8 bits (i.e. the first eight pins defined as output) are saved!

If more than eight GND or Vcc are defined, only the first eight definitions are taken into account.

The last set assignment is retained until the next RESET.

6.12.2 Starting the readout process

When the readout process is started, the configuration is initially displayed for 5 seconds:

Figure 6.96: PAL10H8 / PAL10L8: 10 inputs, 8 outputs, 1024 input combinations
Vcc = pin 20 and GND = pin 10

The readout process is then started and at the end a CRC32 is displayed, which was calculated using the output values determined. This value is used as the file name for a binary file which saves the output values (maximum 8 output pins).

Figure 6.97: PAL with CRC32

A CSV file is also saved that contains all input values with the associated output values. The first line contains the pin numbers related to a 20-pin or 24-pin IC.

Depending on the number of inputs, the saving process can take a few minutes.

More information can be found in chapter 15.

6.12.3 Further possibilities for use with example

The PAL readout function can in principle be used for all chips that work purely combinatorial. E.g. the behavior of a normal 7400 can also be recorded with it. To do this, the inputs and outputs of the chip must be specified. Since the 7400 only has 14 pins, pins 8 to 13 in the "virtual 20-pin socket" are not used and must be ignored. The power supply is at pin 7 (GND) and pin 14 (Vss) of the IC (or pin 20 of the socket).

Figure 6.98: Definition of a 7400 (4-way NAND)

In the next step we get the configuration displayed:

Figure 6.99: 7400 (4-way NAND): 8 inputs and 4 outputs, 256 input combinations
Vcc = pin 20 and GND = pin 7 (attention: counting with a 20-pin socket)

After reading out, there are two files on the SD card:

1) A binary file that only contains the output values (since there are four outputs, these are stored in the lower 4 bits):

```
00000000 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000010 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000020 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000030 0b 0b 0b 0a 0b 0b 0b 0a 0b 0b 0b 0a 09 09 09 08
00000040 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000050 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000060 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000070 0b 0b 0b 0a 0b 0b 0b 0a 0b 0b 0b 0a 09 09 09 08
00000080 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000090 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
000000a0 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
000000b0 0b 0b 0b 0a 0b 0b 0b 0a 0b 0b 0b 0a 09 09 09 08
000000c0 07 07 07 06 07 07 07 06 07 07 07 06 05 05 05 04
000000d0 07 07 07 06 07 07 07 06 07 07 07 06 05 05 05 04
000000e0 07 07 07 06 07 07 07 06 07 07 07 06 05 05 05 04
000000f0 03 03 03 02 03 03 03 02 03 03 03 02 01 01 01 00
```

Figure 6.100: File: d9bb1c56.bin

2) A CSV file that contains both the input and output values in readable form (here only the first 20 values out of a total of 256 values):

13	12	10	9	5	4	2	1	-	11	8	6	3
19	18	16	15	5	4	2	1	-	17	14	6	3
0	0	0	0	0	0	0	0	-	1	1	1	1
0	0	0	0	0	0	0	1	-	1	1	1	1
0	0	0	0	0	0	1	0	-	1	1	1	1
0	0	0	0	0	0	1	1	-	1	1	1	0
0	0	0	0	0	1	0	0	-	1	1	1	1
0	0	0	0	0	1	0	1	-	1	1	1	1
0	0	0	0	0	1	1	0	-	1	1	1	1
0	0	0	0	0	1	1	1	-	1	1	1	0
0	0	0	0	1	0	0	0	-	1	1	1	1
0	0	0	0	1	0	0	1	-	1	1	1	1
0	0	0	0	1	0	1	0	-	1	1	1	1
0	0	0	0	1	0	1	1	-	1	1	1	0
0	0	0	0	1	1	0	0	-	1	1	0	1
0	0	0	0	1	1	0	1	-	1	1	0	1
0	0	0	0	1	1	1	0	-	1	1	0	1
0	0	0	0	1	1	1	1	-	1	1	0	0
0	0	0	1	0	0	0	0	-	1	1	1	1
0	0	0	1	0	0	0	1	-	1	1	1	1
0	0	0	1	0	0	1	0	-	1	1	1	1
0	0	0	1	0	0	1	1	-	1	1	1	0

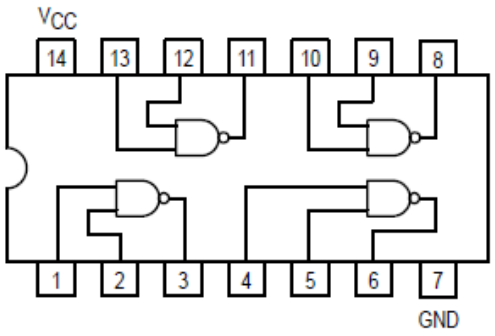


Figure 6.101: Truth table

The first line (in gray) contains the pin numbers related to a 20-pin socket. The yellow line has been added for clarity and contains the corrected pin numbers based on a 14-pin socket.

The behavior of the outputs at pin 3 and pin 6 can be clearly seen, which are always LOW when both inputs, i.e. pin 1/2 and pin 4/5, are HIGH (NAND gate).

The binary file contains the values in the right column:

- 1st byte: value 0x0f (1111)
- 2nd byte: value 0x0f (1111)
- 3rd byte: value 0x0f (1111)
- 4th byte: value 0x0e (1110)
- etc. up to the 256th byte

After the truth table has been generated, the possibly tricky part starts: The circuit synthesis, in which the functional equation is created as a disjunctive or conjunctive normal form from the truth table. An essential component is the optimization and finding of minimized function equations with De Morgan's laws and switching algebra. KV diagrams can also help with optimization, at least if there are a few (up to four) input variables.

Further references:

- https://en.wikipedia.org/wiki/Boolean_function
- https://en.wikipedia.org/wiki/Disjunctive_normal_form
- https://en.wikipedia.org/wiki/Karnaugh_map

6.13 Testing SOIC (SOP) components

With the help of an adapter, ICs in the SOIC or SOP packaging can also be tested. These adapters are available for a few Euros on common marketplaces. When buying socket adapters, make sure to select adapters that have 1:1 pinout. Also check datasheets to ensure through-hole and surface-mounted parts have the same pinout. In some cases, an adapter can be used for an IC that has a different pinout and the adapter translates the connections using 1:1 by signal instead of using a 1:1 by pin number scheme. Check the IC datasheet and socket adapter datasheet when in doubt.

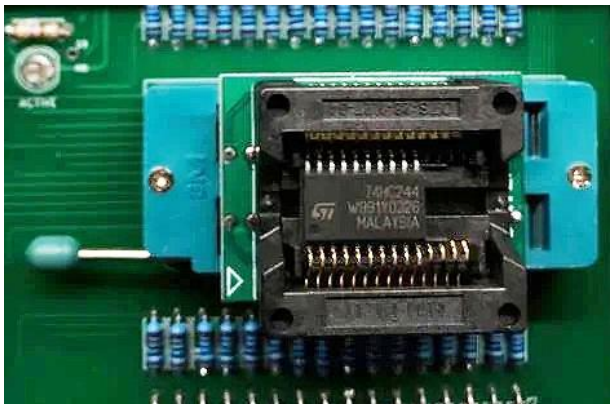


Figure 6.102: SOIC-28 adapter

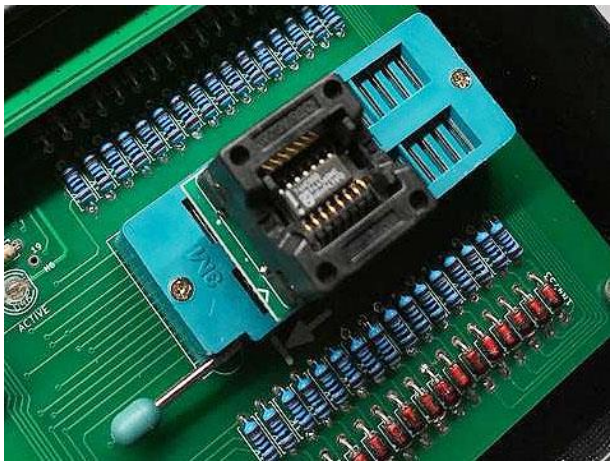


Figure 6.103: SOIC-16 adapter

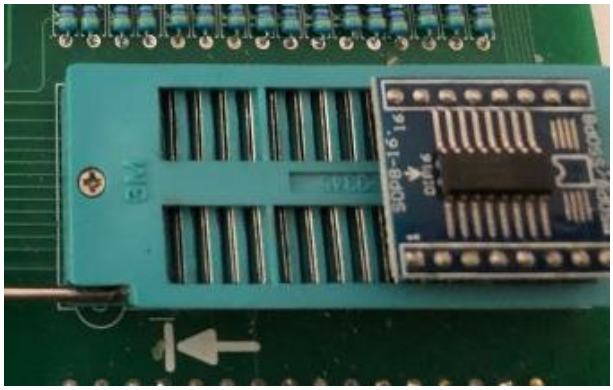


Figure 6.104: SOIC-16 adapter

7 Programming EPROMs

From firmware version v.20 the RCT can also program EPROMs. Programming an EPROM requires a simple adapter that provides the necessary programming voltage. The SD card adapter and an SD card containing the binary file to be programmed are also required.

Please note:

The RCT is primarily a tester, not a programmer. It cannot be ruled out that further ICs will be supported in the future. But then it will support ICs that are no longer known to common programmers. It is also not guaranteed that an EPROM from every manufacturer can be programmed. The RCT only supports standard algorithms.

Before programming, a blank check of the EPROM is done. This test can be skipped by long pressing the OK button.

Compatibility matrix:

In order to program an EPROM, an appropriate adapter is required that provides the programming voltage. The following table shows which EPROMs can be programmed:

Programmable EPROMs			
EPROM	**Menu**	**Required adapter**	**Voltage (Vpp)**
2708	2708	2708 Adapter	26 V
2516	2716	2716/2532 Adapter	25 V
2758 [1]			25 V
2716			25 V
2716 B			12,5 V
TMS2716	TMS2716	TMS2716 Adapter	25 - 27 V
2532	2532	2716/2532 Adapter	25 V
2732	2732	2732 Adapter	25 V
2732 A			21 V
2732 B			12,5 V
2564	2564	2564 Adapter	25 V

Please be sure to consult the data sheet regarding the permitted voltage.

Notes:

[1] The 2758 is effectively a 2716 with either the lower or upper 1K bytes defective and Ar (pin 19) needs to be set accordingly. It can be programmed as 2716 with a trick: the 1 kByte must be in the file twice in a row. The chip is then programmed once with Ar=0 (A10=0) and once with Ar=1 (A10=1).
The TMS2508 should also be able to be programmed in this way. Here, however, the memory matrix is then programmed twice (pin 19 is nc). However, this has not yet been tested.

7.1 Setting the programming voltage (Vpp)

When programming EPROMs, it is essential to ensure that the programming voltage (Vpp) is correct. This can be found in the data sheet. The step-up regulator of the programming adapter must be set as precisely as possible to this voltage. If this is too low, programming fails; if it is too high, the memory can be damaged (the maximum programming voltage should not be exceeded by more than 0.5 V).

To set Vpp, first connect the step-up controller to 5V. There are two options for this:

1. Connection to an external 5V voltage source (the adapter must not be in the tester), or
2. Connection to the socket (between the DC/DC module and the display) to Vcc (5V) and GND.

The 5V are then connected in the front left, GND/Vss in the front right. When the adapter has a Vcc/GND connection, this should be used for this. The programming voltage should be measured at the rear connections with a voltmeter. To do this, turn the (blue) trimmer counter-clockwise until the appropriate voltage is displayed (the voltage will not change at the beginning, but increase suddenly at the end).

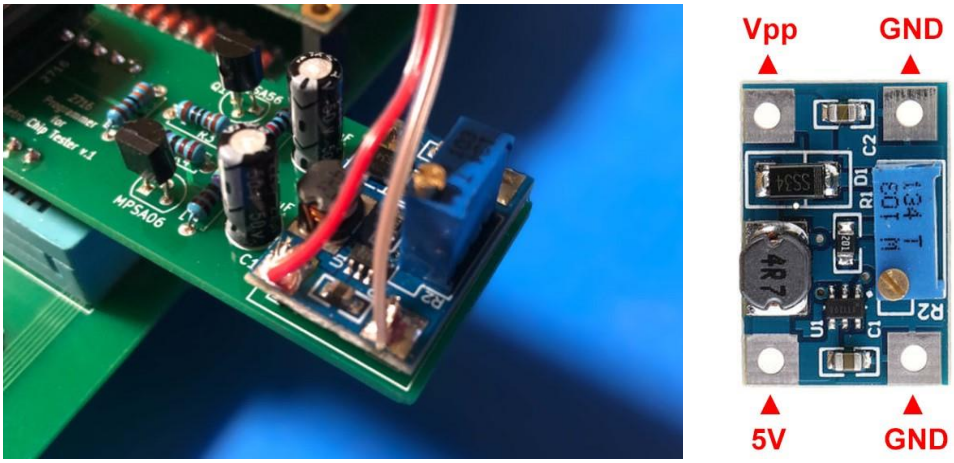


Figure 7.1: Connection of the input voltage (5V)

7.2 Selection of the ROM file to be programmed (from FW v.23)

After selecting the menu entry, the ROM file to be programmed can be chosen. Since there might be several files on the SD card, only files matching the following criteria are displayed:

Files starting with

- "2708." (when 2708 has been selected)
- “2716.” (when 2716 or TMS2716 has been selected)
- “2532.” (when 2532 has been selected)
- “2732.” (when 2732 has been selected)
- “2564.” (when 2564 has been selected)

and all files ending with “.rom” or “.bin”.

The maximum length of a filename including the extension is 20 characters. Files with longer filenames are ignored. You can exit the file selection pressing “FUNC/TOP” or at the end of the list by pressing “OK” (or “FUNC/TOP”).

7.3 2708 EPROM

Programming the 2708 EPROM requires a simple adapter that provides the necessary programming voltage. The SD card adapter and an SD card containing the binary file to be programmed are also required.

Preparation:

The programming voltage of the adapter must be set to 25V – 27V for most 2708 EPROMs. It is recommended to try programming with 25V first. Please be sure to consult the data sheet regarding the permitted voltage.

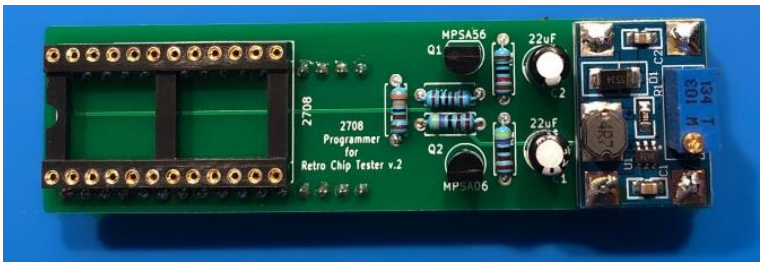


Figure 7.2: Programming adapter

The adapter was tested without additional modification with EPROMs, which have the following maximum current consumption during programming: Icc < 10 mA, Idd < 65 mA, Ibb < 45 mA, Iprgm < 20 mA.

If the current consumption of Icc or Iprgm is higher and the programming fails as a result, connect Vcc and GND (located on the step-up module at the bottom left (Vcc) and right (GND)) with "[Vcc]" and “Vss/GND” available at the socket between display and DC/DC module. The easiest way to do this is to solder the wire directly to the step-up module.

Programming:

The binary file must be stored on the SD card under the name "`rom2708.bin`" and contain exactly 1024 bytes. From FW v.23 this file can be selected.

Then **2708** is selected in the **PROGRAMMING** submenu. The programming process starts immediately. If everything runs without errors, "Finished" appears after a while.

Figure 7.3: 2708 EPROM with adapter

The following errors can occur:

- "rom2708.bin missing": The binary file could not be found.
- "End of file error!": The binary file contains less than 1024 bytes.
- "Verification failed": The content of the memory module differs from the content of the binary file.
- "Blank check failed": The EPROM to be programmed is not empty.

If the check fails, you can try to program the block again.

The programming parameters used: N = 256, t_{PW} = 390usec, t_{SD} > 10usec, t_{DH} > 1usec

7.4 2716/2516 EPROM

Preparation:

The programming voltage of the 2716/2532 adapter must be set to 25V for most 2716/2516 EPROMs. Please be sure to consult the data sheet regarding the permitted voltage.

It is essential to ensure that the adapter must also be connected to "[Vcc]" and "Vss/GND" on the socket between the display and the DC/DC module, as the EPROMs have too high a current consumption.

Programming:

The binary file must be stored on the SD card under the name "`rom2716.bin`" and contain exactly 2048 bytes. From FW v.23 this file can be selected.

Then **2716/2516** is selected in the **PROGRAMMING** submenu. The programming process starts immediately. If everything runs without errors, "Finished" appears after a while.

Figure 7.4: 2716 EPROM with adapter

The following errors can occur:

- "rom2716.bin missing": The binary file could not be found.
- "End of file error!": The binary file contains less than 2048 bytes.
- "Verification failed": The content of the memory module differs from the content of the binary file.
- "Blank check failed": The EPROM to be programmed is not empty.

If the check fails, you can try to program the block again.

The programming parameters used: N = 1, t_{PW} = 50msec, t_{SD} > 10usec, t_{DH} > 1usec

Note:

A "TMS2516" is **compatible** with the usual "2716". A "TMS2716" is **not compatible** with the "2716"; several supply voltages are required here.

7.5 TMS2716 EPROM (from FW v.22)

Preparation:

The programming voltage of the TMS2716 adapter must be set to 25-27V for most TMS2716 EPROMs. Please be sure to consult the data sheet regarding the permitted voltage.

It is essential to ensure that the adapter must also be connected to "[Vcc]" and "Vss/GND" on the socket between the display and the DC/DC module, as the EPROMs have too high a current consumption.

Programming:

The binary file must be stored on the SD card under the name "`rom2716.bin`" and contain exactly 2048 bytes. From FW v.23 this file can be selected.

Then **TMS2716** is selected in the **PROGRAMMING** submenu. The programming process starts immediately. If everything runs without errors, "Finished" appears after a while.

The following errors can occur:

- "rom2716.bin missing": The binary file could not be found.
- "End of file error!": The binary file contains less than 2048 bytes.
- "Verification failed": The content of the memory module differs from the content of the binary file.
- "Blank check failed": The EPROM to be programmed is not empty.

If the check fails, you can try to program the block again.

The programming parameters used: N = 256, t_{PW} = 390usec, t_{SD} > 10usec, t_{DH} > 1usec

Note:

A "TMS2516" is **compatible** with the usual "2716". A "TMS2716" is **not compatible** with the "2716"; several supply voltages are required here.

7.6 2532 EPROM

Preparation:

The programming voltage of the 2716/2532 adapter must be set to 25V for most 2532 EPROMs. Please be sure to consult the data sheet regarding the permitted voltage.

It is essential to ensure that the adapter must also be connected to "[Vcc]" and "Vss/GND" on the socket between the display and the DC/DC module, as the EPROMs have too high a current consumption (see Figure 7.4).

Programming:

The binary file must be stored on the SD card under the name "`rom2532.bin`" and contain exactly 4096 bytes. From FW v.23 this file can be selected.

Then **2532** is selected in the **PROGRAMMING** submenu. The programming process starts immediately. If everything runs without errors, "Finished" appears after a while.

The following errors can occur:

- "rom2532.bin missing": The binary file could not be found.
- "End of file error!": The binary file contains less than 4096 bytes.
- "Verification failed": The content of the memory module differs from the content of the binary file.
- "Blank check failed": The EPROM to be programmed is not empty.

If the check fails, you can try to program the block again.

The programming parameters used: N = 1, t_{PW} = 50msec, t_{SD} > 10usec, t_{DH} > 1usec

7.7 2732 EPROM

Preparation:

The programming voltage of the 2732 adapter must be set to 25V for most 2732 EPROMs and 21V for most 2732A. Please be sure to consult the data sheet regarding the permitted voltage.

It is essential to ensure that the adapter must also be connected to "[Vcc]" and "Vss/GND" on the socket between the display and the DC/DC module, as the EPROMs have a too high current consumption (see Figure 7.4).

Programming:

The binary file must be stored on the SD card under the name "`rom2732.bin`" and contain exactly 4096 bytes. From FW v.23 this file can be selected.

Then **2732** is selected in the **PROGRAMMING** submenu. The programming process starts immediately. If everything runs without errors, " Finished" appears after a while.

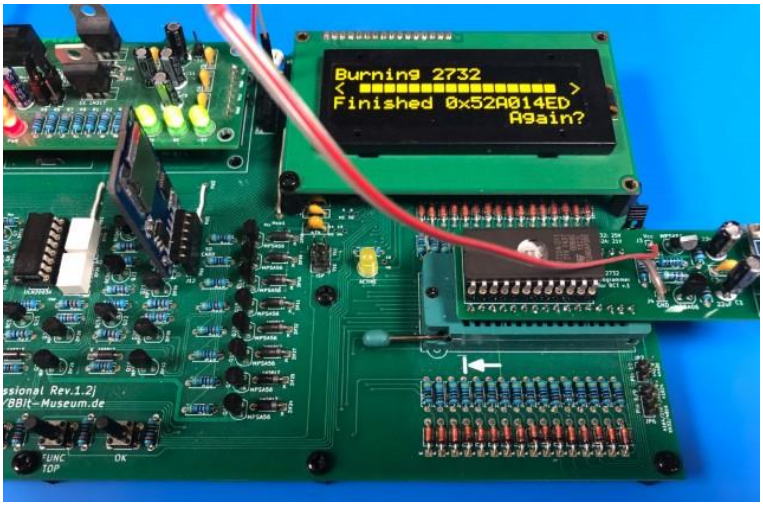


Figure 7.5: 2732 EPROM with adapter

The following errors can occur:

- "rom2732.bin missing": The binary file could not be found.
- "End of file error!": The binary file contains less than 4096 bytes.
- "Verification failed": The content of the memory module differs from the content of the binary file.
- "Blank check failed": The EPROM to be programmed is not empty.

If the check fails, you can try to program the block again.

The programming parameters used: N = 1, t_{PW} = 50msec, t_{SD} > 10usec, t_{DH} > 1usec

7.8 2564 EPROM (from FW v.22)

Preparation:

The programming voltage of the 2564 adapter must be set to 25V for most 2564 EPROMs. Please be sure to consult the data sheet regarding the permitted voltage.

It is essential to ensure that the adapter must also be connected to "[Vcc]" and "Vss/GND" on the socket between the display and the DC/DC module, as the EPROMs have too high a current consumption (see Figure 7.4).

Programming:

The binary file must be stored on the SD card under the name "`rom2564.bin`" and contain exactly 8192 bytes. From FW v.23 this file can be selected.

Then **2564** is selected in the **PROGRAMMING** submenu. The programming process starts immediately. If everything runs without errors, "Finished" appears after a while.

Figure 7.6: 2564 EPROM mit Adapter

The following errors can occur:

- "rom2564.bin missing": The binary file could not be found.
- "End of file error!": The binary file contains less than 8192 bytes.
- "Verification failed": The content of the memory module differs from the content of the binary file.
- "Blank check failed": The EPROM to be programmed is not empty.

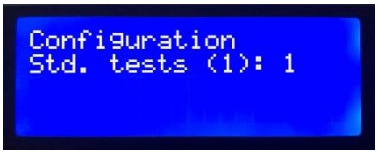
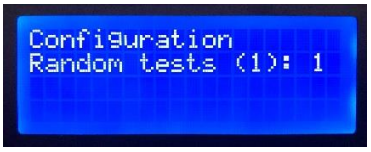
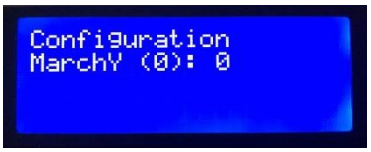
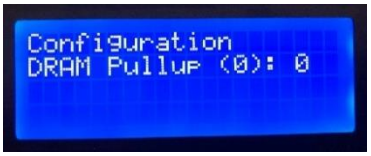
If the check fails, you can try to program the block again.

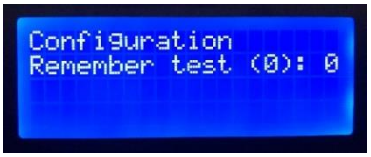
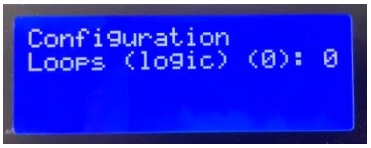
The programming parameters used: N = 1, t_{PW} = 50msec, t_{SD} > 10usec, t_{DH} > 1usec

8 Settings menu / configuration menu

Certain settings can be made via the configuration menu, which are permanently saved in the EEPROM of the ATmega2560.

The fuses must be set accordingly so that this setting is still available even after the ATmega2560 has been reprogrammed (see also section 4). However, it can also make sense that the EEPROM is also cleared when the firmware is updated.

Activate SD card support. With support activated for SD cards, ROMs and EPROMs can also be read out using a suitable SD card module. Default setting: OFF (0)	
Performing the standard tests (pattern tests, counting tests). These are the standard tests that are usually performed. Default setting: ON (1)	
Perform random number tests for SRAMs and DRAMs. These tests slow down testing significantly, so they can be turned off. Default setting: ON (1) – Only in combination with “Std. tests”.	
Performing tests according to the “March Y” algorithm. Instead of (or in addition to) the standard tests, these tests can be performed. Default setting: OFF (0)	
Performing tests according to the “March U” algorithm. Instead of (or in addition to) the standard tests, these tests can be performed. Default setting: OFF (0)	
The data lines of SRAMs are connected with pull ups. Default setting: ON (1)	
The data lines of DRAMs are connected with pull ups. Default setting: OFF (0)	
The data lines of (P)ROMs and EPROMs are connected with pull ups. Default setting: ON (1)	
If this function is switched on, the tester tries to recognize in advance whether pull-ups are required for DRAMs or SRAMs (only if pull-ups for DRAMs or SRAMs are deactivated). [*] Default setting: OFF (0)	
(P)ROMs and EPROMs can be padded either by ‚0‘ or by ‚1‘, when a word consists of less than 8 bits. Default setting: ‚0‘-Padding (0)	

When programming EPROMs, cells with 0xFF (empty areas) can be skipped. Default setting: OFF (0)	
If this function is activated, the tester tries to determine the DRAM features (page mode, static column mode). [*] Default setting: OFF (0)	
Remember the last tested chip. Default setting: OFF (0)	
This specifies how often a logic test is repeated when triggered by a long press on OK. The number corresponds to 2^n, where n is the set value. Default: Off (0), 1 = 2 runs, 2 = 4 runs, 3 = 8 runs, … (from 6 the Tester switches into a fast mode), 11 = infinite runs	
Turn on/off advisory tones and button clicks. Default setting: ON (1)	
Set an alternative test mode (see table below). Default setting: OFF (0)	
Before executing a test, the Tester checks whether an SRAM or DRAM has been inserted. [*] Default setting: ON (1)	
Increasing the key repetition speed. Default setting: OFF (0)	
The self-defined SRAM and DRAM memory modules will be deleted.	
Reset the configuration to the default values.	
Exit the settings menu and save the settings.	

[*] These functions are experimental. Should errors occur in relation with these, the default setting should be used.

Meaning of the **alternative test** mode (is executed with a long keypress on OK):

Value	Description
0	No alternative mode specified.
1	DUMP mode: A long keypress on OK runs a forced standard test for SRAMs or DRAMs and saves a dump of the read values to the SD card. This must not be enabled when ROMs should be dumped! Displays „**DMP**“. This mode does not have to be activated if ROMs/EPROMs are to be dumped! It only influences the behavior with regard to SRAMs and DRAMs.
2	Only for 4164 DRAM: Uses RMW mode instead of Early-Write-Mode when writing data. Displays „**RMW**“.

9 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.

9.1 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.

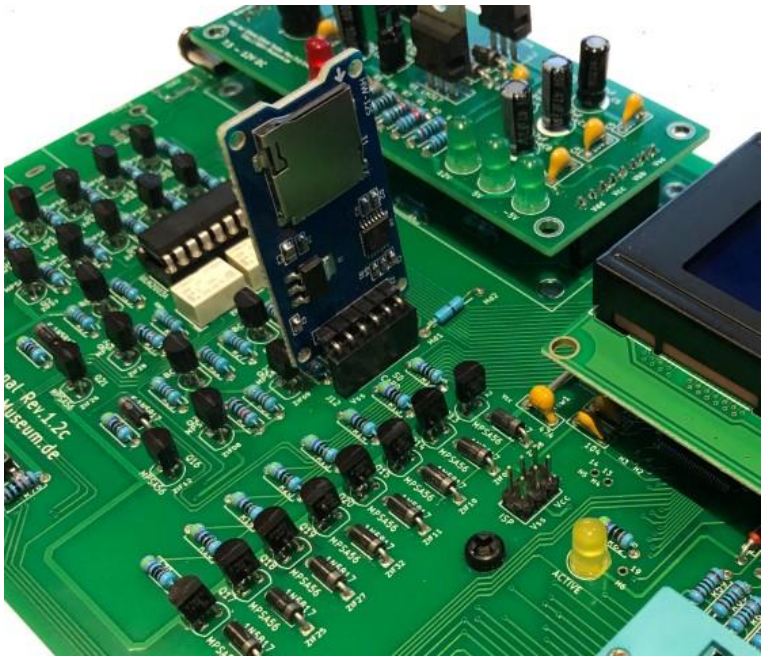


Figure 9.1: SD card adapter in the assembled header.

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).

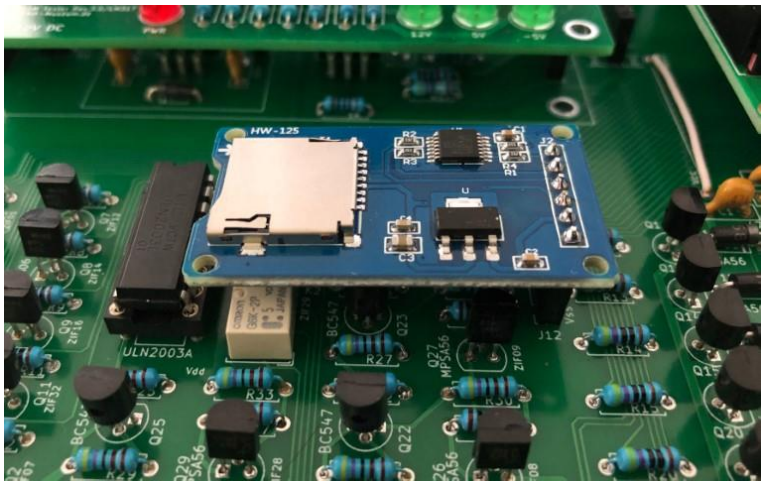


Figure 9.2: SD card adapter installed horizontally

9.2 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.

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.

9.3 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):

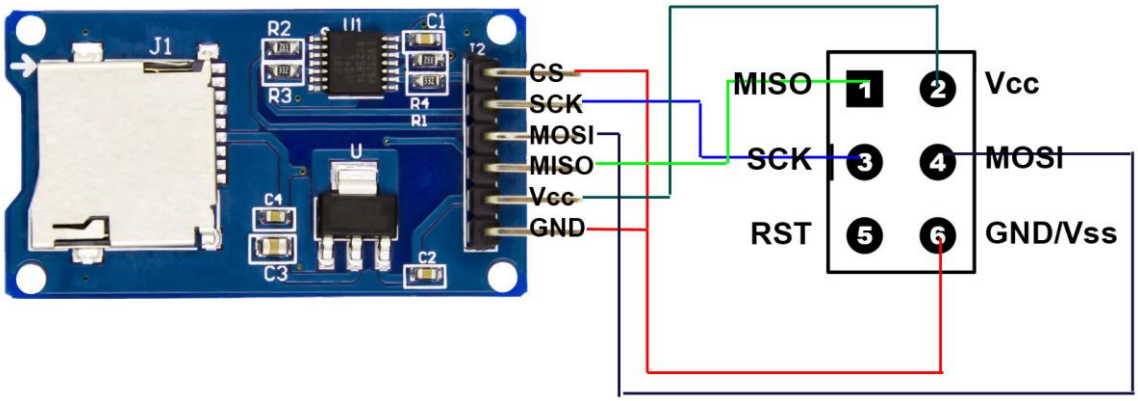


Figure 9.3: Wiring of the SD card adapter

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

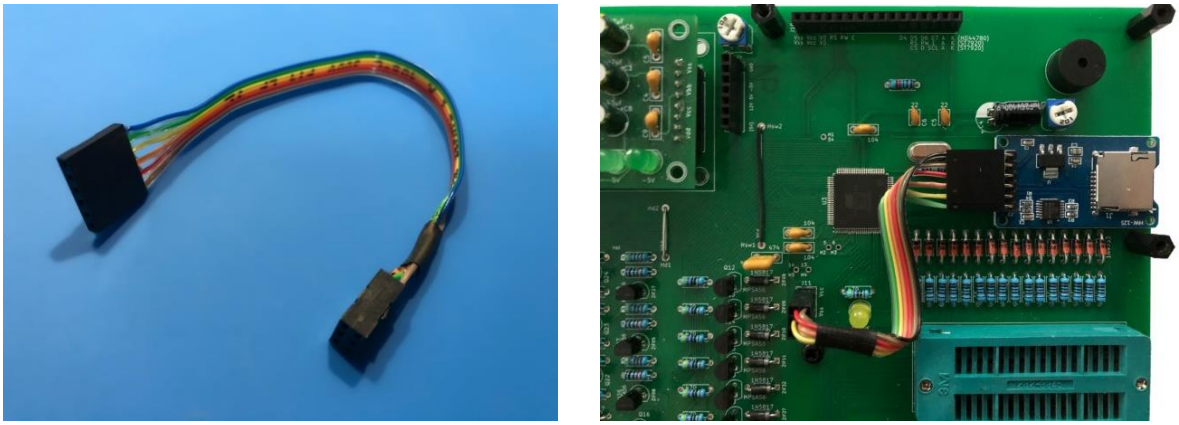


Figure 9.4: SD card adapter with extension

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. 9.1).

10 Warnings

Warnings regarding performing tests:

Always insert chips **aligned left** (in the socket: “upwards”)!

All voltages are only switched on during the test.

The ports of the ATmega2560 are protected against short circuits, i.e. the ATmega2560 should not be damaged if such a defect occurs. Usually nothing happens when a defective chip is tested, but there is always a (very little) risk.

Warnings regarding the power supply of the memory tester:

The tester should be operated with a power supply. The current consumption will rarely be over 500mA, but especially the older chips need 180-200mA at 5V, which together with LCD, ATmega2560 and the LEDs can be over 500mA. Excessive power consumption can damage a USB port. If short circuits occur, they can also damage a USB port.

The polyfuse protects the USB port against a current >1100mA. The polyfuse can be omitted and short-circuited. If the polyfuse is not used, the tester should only be operated with a power supply.

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.

11 Known Issues and Limitations

First of all: The tester cannot recognize all the problems, but quite a few. ;)

11.1 Detection of errors (memory chips)

The Retro Chip Tester was developed in order to test "old" memory chips from the 1970s and 1980s, which are often no longer recognized by today's programming devices. While it is often recommended to test chips in a similar 'tester' device to test their functionality, it is forgotten that the rest of the hardware is also correspondingly old and frequent switching on and off can provoke further errors.

An ideal memory tester should, of course, detect

- Defective memory cells,
- Timing errors due to material fatigue (e.g. with DRAMs), and
- Incorrect signal levels due to material fatigue.

Unfortunately, such a tester cannot be manufactured at an acceptable price, a real tester will always be a compromise between the detection rate of defective chips and the price.

Defective memory cells:

Basically, the following errors should be recognized:

- Stuck-At Fault
- The logical value of a cell is always 0 or 1.
- Transition fault
- A change from 0 → 1 or 1 → 0 fails.
- Coupling fault
- A write operation in one cell changes the contents of a second cell.
- Neighborhood Pattern Sensitive Fault
- The content of one cell is influenced by the content of another cell.
- Address Decoder Fault
- Any error related to the address decoder (no access possible, changing several cells at the same time, accessing a cell from several addresses)

[Lindner13][7] differentiates these faults (based on SRAMs) in more detail in single-cell fault primitives and two-cell fault primitives, i.e. connected cells. His more detailed description is cited in the following two paragraphs.

Single-cell fault primitives are:

[7] [Lindner13] Test Set Optimization for Industrial SRAM Testing, Michael Linder, Technische Universität München, 2013, http://d-nb.info/1045729981

- State Faults: The value of the cell flips without any sensitizing operation and depends on the initial state of the cell.
- Transition Faults: The cell fails to flip when it is written with the opposite value. I.e. transition 0 → 1 or 1 → 0.
- Write Destructive Fault: A non-transition write operation (0 → 0 or 1 → 1) causes a transition.
- Read Destructive Fault: A read operation causes the cell to flip and the incorrect value is returned to the output.
- Deceptive Read Destructive Fault: A read operation causes the cell to change its value, however the correct output is returned.
- Random Read Destructive Faults: A read operation flips the cell and a random logic value is returned to the output.
- Incorrect Read Fault: A read operation returns the incorrect value to the output; however the stored value in the cell remains correct.
- Random Read Fault: A read operation returns a random logic value to the output while the stored value remains correct.
- Undefined State Fault: Without any sensitizing operation, the logic value of a cell flips into an undefined state.
- Undefined Write Fault: An undefined state of the cell is causes by a write operation.
- Undefined Read Fault: The cell is brought into an undefined state by a read operation.
- Stuck-At Fault: The cell remains stuck at a value for any operation.
- No Access Fault: The cell cannot be accessed. A write operation cannot change the value of the cell and a read operation returns a random value. This fault needs not to be caused by the address decoder, but can also be caused by an open word-line.
- Data Retention Fault: The value of a cell changes after a certain time T without accessing the cell.

Two-cell fault primitives are:

- State coupling fault: The v-cell is forced into a given logic state if the a-cell is in a given logic state without performing any operation on the v-cell or a-cell.
- Undefined State coupling fault: The state of the v-cell is undefined while the a-cell is in a given logic state without performing any operation on the v-cell or a-cell.
- Disturb coupling fault: Any operation performed on the a-cell causes the v-cell to flip.
- Undefined Disturb coupling fault: Any operation performed on the a-cell forces the v-cell into an undefined state.
- Idempotent coupling fault: A transition write operation on the a-cell causes the v-cell to flip.
- Inversion coupling fault: A transition write operation on the a-cell inverts the logic value of the v-cell.
- Transition coupling fault: A given logic value in the a-cell causes a failing transition write operation performed on the v-cell.
- Write Destructive coupling fault: A given logic state of the a-cell causes a transition in the v-cell although a non-transition write operation is performed on the v-cell.
- Read Destructive coupling fault: If the a-cell is in a given state, a read operation on the v-cell changes its value and returns the incorrect value to the output.
- Deceptive Read Destructive coupling fault: If the a-cell is in a given state, a read operation on the v-cell changes its value and the correct value is returned to the output.
- Random Read Destructive coupling fault: If the a-cell is in a given state, a read operation on the v-cell changes the value in the v-cell and a random value is returned to the output.
- Incorrect Read coupling fault: If the a-cell is in a given state, a read operation on the v-cell returns the incorrect value to the output.
- Random Read coupling fault: If the a-cell is in a given state, a read operation on the v-cell returns a random value to the output while the value of the v-cell remains correct.
- Undefined Write coupling fault: A write operation on the v-cell forces it into an undefined state, while the a-cell is in a given state.
- Undefined Read coupling fault: A read operation on the v-cell forces it into an undefined state, while the a-cell is in a given state. The value returned to the output can be correct, incorrect or random.

The tester uses traditional tests by default: The chips are written with different patterns, when a memory cell is defective, it will usually be found. Furthermore, the cells are written with an ascending and descending pattern, so a failed address line can be discovered. The slowest (but from these also most reliable) test is the random number test. If you want to test chips quickly, you can switch off this test in the configuration.

These traditional tests recognize the above errors usually reliably. Several algorithms have been developed to optimize the speed of memory tests, e.g. the “March” tests. The chip tester currently supports the “March Y” and “March U” test.

The “March-Y” test is faster than the traditional tests combined. It is a good compromise between recognition accuracy and speed. If you want, you can activate this test algorithm instead of or in addition to the traditional tests in the configuration. The test runs according to the following sequence { ⇳(w0); ⇧(r0,w1,r1); ⇩(r1,w0,r0); ⇩(r0) }. The running time is 8n.

The “March-U” test is slower than the traditional tests combined, but detects errors very reliably. If you want, you can activate this test algorithm instead of or in addition to the traditional tests in the configuration. The test runs according to the following sequence: { ⇳(w0); ⇧(r0,w1,r1,w0); ⇧(r0,w1); ⇩(r1,w0,r0,w1); ⇩(r1,w0); ⇩(r0) }. The running time is 14n.

The test for defective memory cells is very reliable and usually identifies most of the defective RAMs.

The following table gives an overview of which errors can be identified by which test.

	Pattern	Alt/Count	Rnd	March-Y	March-U
Stuck-At Fault	X	(X)	(X)	X	X
Transition Fault	X	(X)	(X)	X	X
Coupling Fault		(X)	X	X	X
Neighborhood Pattern Sensitive Fault		(X)	X	X	X
Address Decoder Fault		(X)	X	X	X

Timing-Errors:

Accurate timing tests cannot be performed with a 16MHz clocked ATmega2560. A resolution of at least approx. 1-5ns would be necessary here, which cannot be achieved with an ATmega2560, that already requires 62,5ns for a command. This means that signal edges and compliance with timings cannot be measured. The refresh for DRAMs is also not explicitly tested, but only implicitly by accessing memory cells in succession (the cycle takes longer than 2 ms). Usually this shows whether a memory cell is still able to store data. Such timing errors are quite rare, so most times the memory cells themselves are defective.

The tester addresses all chips with a rather conservative (= slow) timing. If a chip has problems with these timings, it is very likely that it is defective. The error itself is indistinguishable from a bit error and is displayed by the tester as such.

Incorrect signal levels:

It is also possible that a chip that is still running correctly in a computer (just now) is recognized by the tester as defective. If the tester is operated via USB, the supply voltage is usually at 4.7 to 4.8V. There are chips that, when operated at less than 5V, no longer work properly due to aging. This is not ok, because usually +/- 10% are permissible for Vcc.

Another problem that cannot be solved satisfactorily with an ATmega2560 is the determination of the correct voltage level. Since the chips are only addressed digitally, it cannot be determined whether a chip complies with the voltage levels correctly. Theoretically, the analog ports could be used for this, but due to the 16 ports this would only be usable with a few chips and the ports would have to be calibrated, which increases the hardware costs.

Another problem is the different voltage levels of the ATmega2560 and the IC under test: According to the data sheet, a 4164 should deliver a maximum of 0.8V for a LOW, at least 2.4V for a HIGH. At 5V supply voltage, the ATmega2560 recognizes levels below 1.5 V as LOW, levels above 3 V as HIGH. Chips that deliver just over 2.4V at an output may be displayed as faulty by the tester even though they are still within the allowed limits but so close to the upper or lower limits that the chip is marginal at best and should not be used.

More tests:

In principle, further special tests would be useful:

- Short circuits between signal lines or interrupted signal lines.
- Power consumption test
- Test of output currents

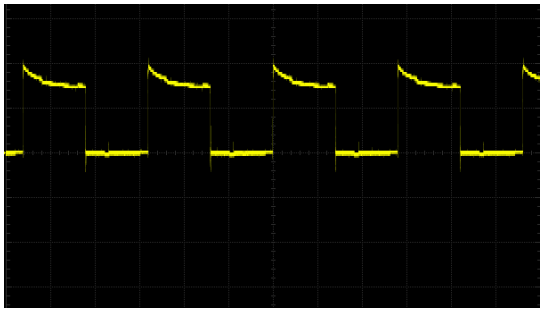
The chip tester detects short circuits between signal lines or interrupted signal lines as bit errors using traditional tests.

A power consumption test cannot be done due to the hardware used. This test assumes that defective chips may have a higher power consumption.

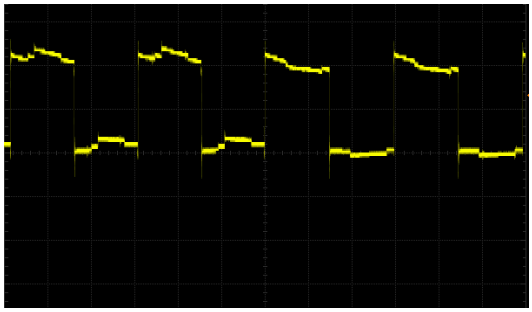
The hardware used cannot test the possible output currents. This test could determine whether outputs still supply enough current to switch inputs.

In the following pictures you can see how different the signals are at the output (e.g. for a 4164):

Reading alternating 0 and 1:

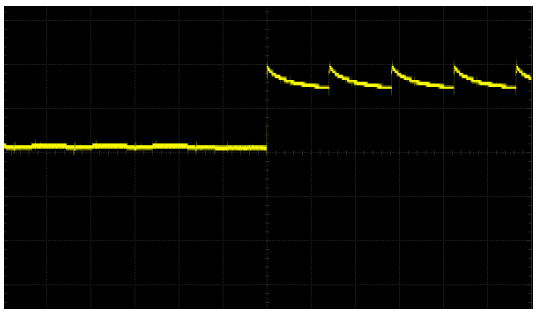


IC 1: D_{out}, level between 3V and 4V

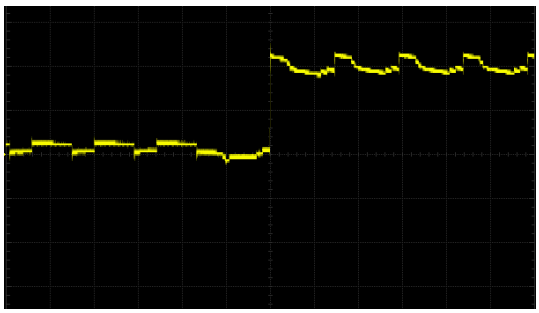


IC 2: D_{out}, level between 4V and 4.5V

Change from “only 0” to “only 1”:



IC 1: D_{out}, level between 3V and 4V



IC 2: D_{out}, level between 3,9V and 4.6V

The voltage level of a HIGH is within the permitted range for both ICs. IC 2 supplies a value of up to 0.5V when reading a LOW. That is above what is allowed. Since the levels at IC 2 vary, tests can sometimes run as error-free, but as a rule the chip should be recognized as faulty.

In a real system (here using an Oric-1 home computer as an example) the signal levels are much worse (although the levels shown in the picture are very good in comparison). The picture shows the signals for RAS (purple), CAS (cyan) and WE (yellow) (4164 DRAM). The memory cycle is clearly visible: RAS becomes active (low), CAS follows 75ns later. RAS stays active for 175ns, CAS stays active for 250ns. CAS goes inactive after RAS just before the next memory cycle begins.

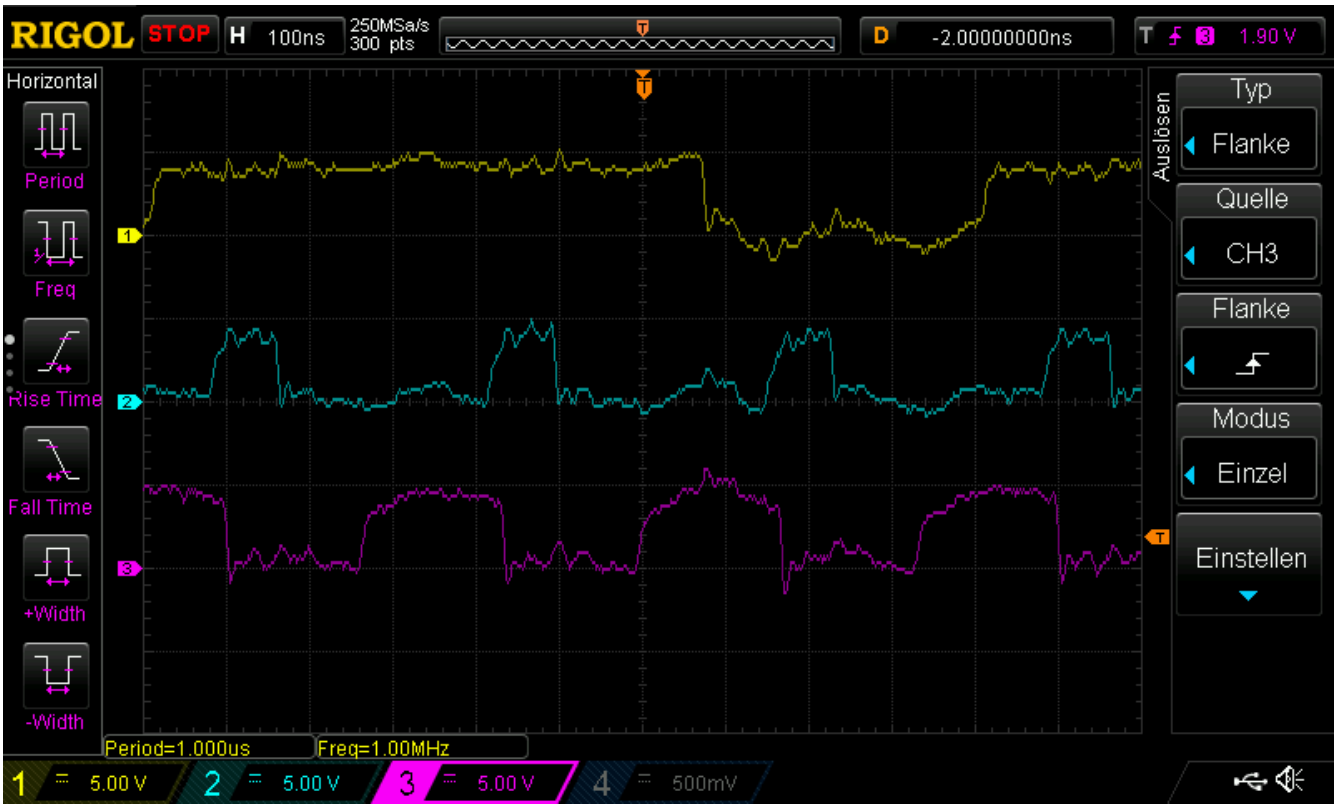


Figure 11.1: RAS/CAS/WE timing (Oric-1)

An IC that is tested as error-free under optimal conditions in the tester could very well have errors in a real system due to the interferences.

General significance of memory tests

The question is always whether certain tests make sense for the intended purpose.

An example:

Let's assume a 120ns SRAM chip like the one used in a ZX Spectrum or C64. After 40 years the memory cells are (hopefully) still 100% OK, but the access time is now 150ns. What should a tester give you as result?

a) "FAIL" because the chip no longer meets the 120ns from the specification? In a ZX Spectrum or C64 it still works very well and without a tester it would probably never have been noticed that this chip is "defective". Yes, exactly: "defective" because the specification of the chip was not met. Since the tester issued a FAIL, will the chip be thrown away?

   or

b) "PASS", because the memory cells are still all in order and the chip continues to work in 99.99% of all cases even without correct timing? If a FAIL is output for defective memory cells, it is definitely defective and can also be disposed of.

You can't just rely on a test result like FAIL or PASS. It depends on the application. In the case of a safety-critical system, on the other hand, there will probably be a requirement that a chip precisely adheres to its technical specification.

If an IC is rated “PASS” by the tester, it is very likely that it is OK. Conversely, an IC rated “FAIL” can still work, but it will probably not work, or at least not in the long term. However, these “borderline case” are relatively rare.

Even two Testers assembled with the same components can come to different results with the same IC if it is a "borderline case". The components used are subject to certain tolerances, so that, for example, voltage drops at the diodes or transistors can be slightly different and thus lead to different results. It is not for nothing that measuring devices are regularly calibrated and use particularly high-quality components, but even then you will never get two identical results.

In addition, timings cannot be accurately tested with low cost hardware. Hardware that measures with an accuracy of 5 ns easily increases the cost of a tester to over 1000 euros (calculated very carefully and possibly a lot more if only a few devices are sold).

If you test 40-year-old chips for their technical specifications, you probably won't find many chips that still meet their technical specification accurately.

Conclusion:

There is no 100% accurate tester, not even with much more expensive hardware, which may correctly detect a failed chip 1-2% more often than the RCT. However, if the RCT reports a chip as defective it is very likely that it is actually defective, especially if it is a common IC that has been well tested by many RCT users.

11.2 Testing “non-volatile static RAM” (NVRAM) (“Zeropower” RAMs)

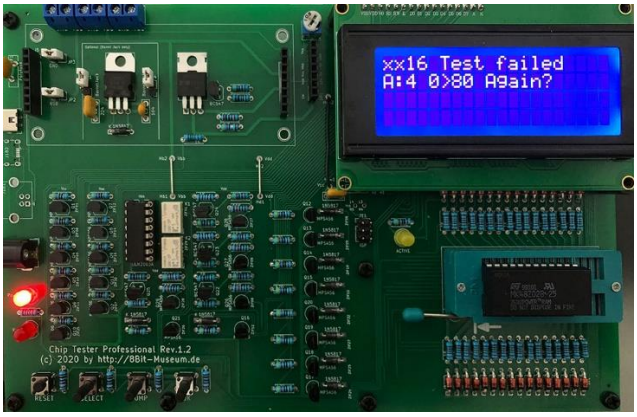
When testing NVRAMs or "Zeropower" RAMs, such as the M48Z02 or M48Z12, the power supply must be observed.

By switching Vcc through a MOSFET and a PNP transistor, a voltage drop of approximately 0.3V is expected. If the tester is supplied with voltage via USB (usually just under 5V), Vcc will finally be in the range from 4.5V to 4.7V. A look at the data sheet shows that with an M48Z02 in the range from 4.5V - 4.75V and with an M48Z12 in the range from 4.2V - 4.5V, the power failure detection becomes active and the chip switches the outputs to tristate mode. This will probably test the chip as defective.

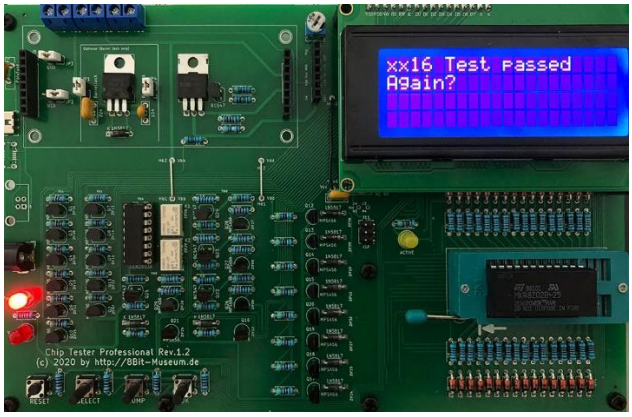
- Automatic power-fail chip deselect and WRITE protection
- WRITE protect voltages
  (V_{PFD} = power-fail deselect voltage):
  - M48Z02: V_{CC}= 4.75 to 5.5 V;
    4.5 V ≤ V_{PFD} ≤ 4.75 V
  - M48Z12: V_{CC}= 4.5 to 5.5 V;
    4.2 V ≤ V_{PFD} ≤ 4.5 V

Extract from the data sheet of the M48Z02 and M48Z12

In this case, the tester should be supplied with 7.5V to 9V via the barrel connector. The linear regulator used in the RCT delivers exactly 5V at the IC, which means a Vcc of approx. 4.7V - 4.8V measured on the ZIF socket. An M48Z02 (and especially the M48Z12) should be able to be tested without errors.



Power supply via USB



Power supply via barrel connector (9V)

11.3 Detection of errors (logic chips and memory ICs)

In principle, the problems described in sec. 11.1 also apply to logic chips. The logic chips are a bit more complicated to test because there must be a separate test for each individual chip.

Most testers for logic chips only test pure logic, including the Retro Chip Tester Pro. However, other (rare) errors can also occur that cannot be detected in this way (especially with old bipolar logic modules):

1. Invalid signal level due to chip aging (e.g. an existing DC voltage offset with a logical LOW, which the tester recognizes as logical HIGH),
2. defective input drivers that require more power due to age,
3. defective output drivers that no longer supply sufficient current (as a rule, the output drivers should supply sufficient current for one to two TTL inputs),
4. other problems.

11.3.1 Invalid signal level

For technical and correct operational reasons the Chip Tester can detect the following levels:

ATmega2560	acc. datasheet	at 5V
Input Low Voltage	max. 0.3 Vcc	max.1.5 V
Input High Voltage	min. 0.6 Vcc	min. 3 V

TTL chips (not CMOS) have the following characteristics:

TTL 5V level	Input	Output
Low Voltage	max. 0.8 V	max. 0.4 V
High Voltage	min. 2.0 V	min. 2.4 V

If, for example, a component actually only supplies 2.4V for a logical HIGH at the output due to its age, this signal may be not recognized as a HIGH by the ATmega2560.

The problem is with all testers that are not explicitly set to the logic level used. The popular China programmer TL866 (old model) uses a 74HC373D as bus driver, so it recognizes the following logic levels:

Symbol	Parameter	Conditions	T_{amb} = 25 °C		
			Min	Typ	Max
74HC373					
V_{IH}	HIGH-level input voltage	V_{CC} = 2.0 V	1.5	1.2	-
		V_{CC} = 4.5 V	3.15	2.4	-
		V_{CC} = 6.0 V	4.2	3.2	-
V_{IL}	LOW-level input voltage	V_{CC} = 2.0 V	-	0.8	0.5
		V_{CC} = 4.5 V	-	2.1	1.35
		V_{CC} = 6.0 V	-	2.8	1.8

Figure 11.2: Logic level of the 74HC373

An IC that only supplies 2.4V for a logical HIGH would still be recognized as such by the TL866, the chip tester will probably recognize this signal as a logical LOW. Conversely, the TL866 recognizes a LOW even at over 1V.

In practice, a TTL IC should be able to deliver a voltage of at least 3.5V (the standard specifies 2.4V), probably a little more due to the low load. Aged ICs can deliver a lower voltage, so that the chip tester no longer reliably recognizes the HIGH signal. In the case of multiple tests, the test sometimes fails, sometimes it is positive (display "L" with an upward arrow = a HIGH was expected, but a LOW was read).

The TL866 II does not use the 74HC373 and sends the signals directly to the ports of the PIC24FJ256GB110 used. This means that the following logic levels are recognized.

PIC24FJ256GB110	acc. datasheet	at 5V
Input Low Voltage (ST)	max. 0.2 Vcc	max.1 V
Input High Voltage (ST)	min. 0.8 Vcc	min. 4 V

The result when testing aged chips with a TL866 II can therefore be different again.

Since the logic level is not exactly adhered to by any tester and basically only the logic is tested, a positive test means that the chip is “most likely OK", while a failed test or a test that randomly fails indicates a "most likely defective" chip. Depending on the situation, the probability of a correct result is sometimes better with one tester and sometimes better with a different tester.

Recommendation: If a test on one Tester - regardless of which one - should fail due to the signal level, but not on another, then the chip should still be rejected because a signal level is obviously no longer optimal (even if 2.4V is conforming to the standard, such a chip will probably not work for a long time).

11.3.2 Defective input or output drivers

The bipolar logic chips are current controlled, i.e. a certain current must be available so that a signal can be reliably recognized. An example of this is described in sec. 6.10.4 (the Chip Tester has a current limiter for protection). Conversely, the Chip Tester only loads the outputs of a logic chip slightly, so that even modules that are no longer able to control one or two TTL inputs can still function correctly in the tester, but no longer can drive another chip in a real circuit.

11.3.3 Timing-Problems

When a memory module is tested several times and the error occurs at different memory cells, then it is possible that the memory is too slow for the test (i.e. the access time has increased over the years). These memory modules can definitely still function in a device. Increased access times are usually not noticeable in a target system, but they are recognized in the tester.

It is also possible that the RCT tests a memory as working, but that memory does not work properly in a device. The memory matrix, the address decoder, the output drivers, etc. are then basically fine, but the memory cannot deal with the fast access times in the target system. Here it should first be checked whether the IC has the correct access time for this system. The RCT always tests memory with conservative timings, which means that it can (in rare cases) happen that memory with strong aging effects that no longer have their nominal access speed but meet the limit values in the RCT are tested as error-free.

11.3.4 "Incorrect" results

Symptom: An IC is tested as OK. However, it does not work in a real circuit.

That can have several reasons. The tester found that, for example in a memory, all memory cells are ok, the input and output drivers are ok, as is the address decoder. But what should be noted is that the ICs was tested in an (almost) perfect environment. The power supply is 5V, is stable and there is de facto no interference from other components. The signals are also almost perfect and have clean flanks. In a real device like a C64, the IC can still behave incorrectly.

Symptom: An IC is tested as FAIL. In a real circuit, however, it works.

There are ICs that no longer meet their specification due to aging effects (e.g. have slower access times than stated) and are therefore shown as faulty in the tester. In a real system, these can still work if access is slower.

In these cases, the tester is right with its result, but the observed behavior may be different.

11.3.5 Other problems

There are other problems that can occur in a real circuit but cannot be tested, including thermal problems that only occur when a chip is, for example, operated at high frequencies and errors only appear after some time when it has warmed up. Any chip that is too hot to touch for longer than 5 seconds is usually marginal and may fail imminently, even if it is working now. Freeze spray can be used to detect this kind of fault. The test result can be OK here, but errors still occur in the real system after some time.

The above applies in principle to all affordable testers for logic chips. Nevertheless, the detection rate is very high and errors can usually be recognized with a very high probability.

11.4 "In circuit" testing

"In circuit" testing may work with some (logic) ICs and it has been reported that it worked fine with the RCT. If soldered ICs are tested, it is essential to do the test in such a way that the board with the IC to be tested is without power supply. In most cases you should also ensure that the Vcc pin of the IC is not connected to the PCB otherwise you will power the board using the RCT.

Be careful with the test results: think about testing a NAND gate where both inputs are connected (so it is working as an inverter). This is a very common circuit on old boards. When the RCT checks the IC the test will fail because it behaves like an inverter and not like a NAND gate.

In circuit testing is not reliable possible. It *may* work but in lots of cases it will not. It depends highly on the circuit. Today's ICs provide a special function for this (boundary scan and test) but ICs from the 1980s do not provide this.

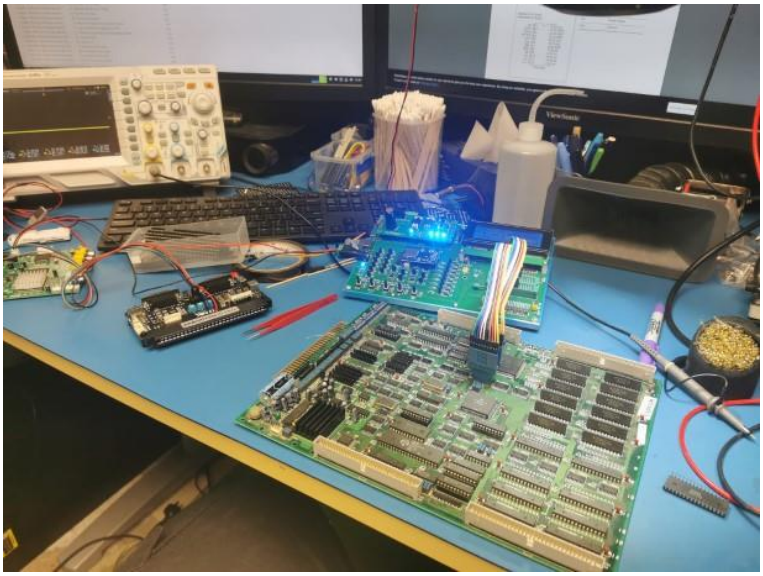


Figure 11.3: RCT in-circuit testing (picture by Darren Jones)

The RCT can test ICs "in circuit" just as well or just as badly as other testers that do so. But the probability that a test fails checking a good IC is very high.

12 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.

12.1 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 12.1: 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).

12.2 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.

12.2.1 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");
```

12.2.2 Switching on/off Vcc

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

12.2.3 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
```

12.2.4 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
```

12.2.5 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
```

12.3 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)).

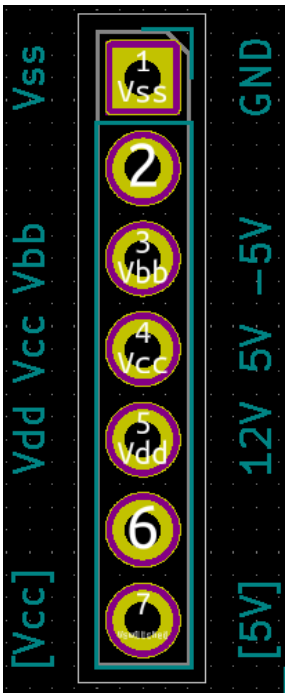
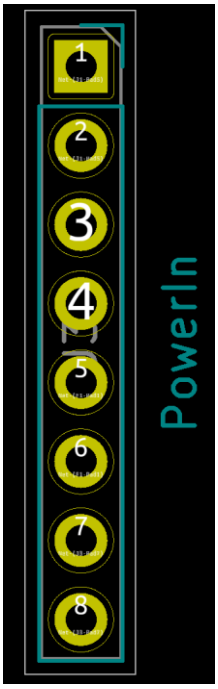


Figure 12.2: Voltage connector

[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.

12.4 Pin assignment of the DC/DC-Converter PCB

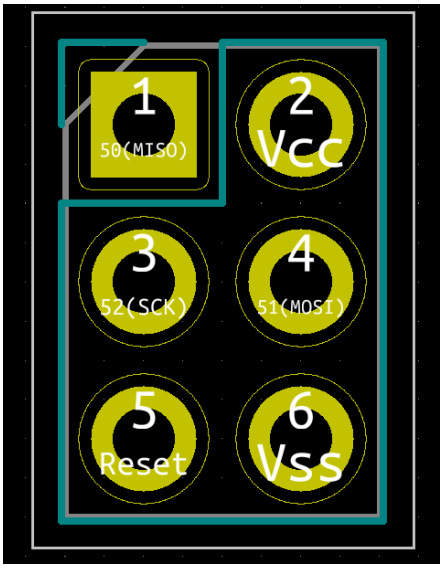


1	Vss
2	Vss
3	nc
4	nc
5	USB 5V
6	USB 5V
7	Barrel Jack (6-9V)
8	Barrel Jack (6-9V)

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

Figure 12.3: DC/DC module pinout

12.5 Pin assignment of the ISP-Header



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

Figure 12.4: ISP pinout

12.6 Pin assignment of the SD-Card-Header

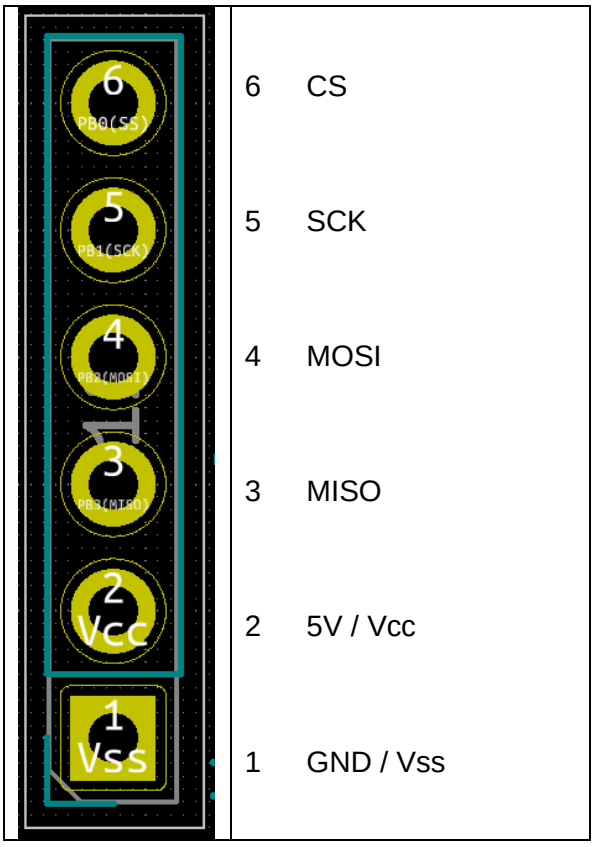


Figure 12.5: SD-Card module pinout

13 Appendix: Troubleshooting

13.1 The ATmega2560 is not recognized during programming

13.1.1 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.

13.1.2 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!

13.1.3 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).

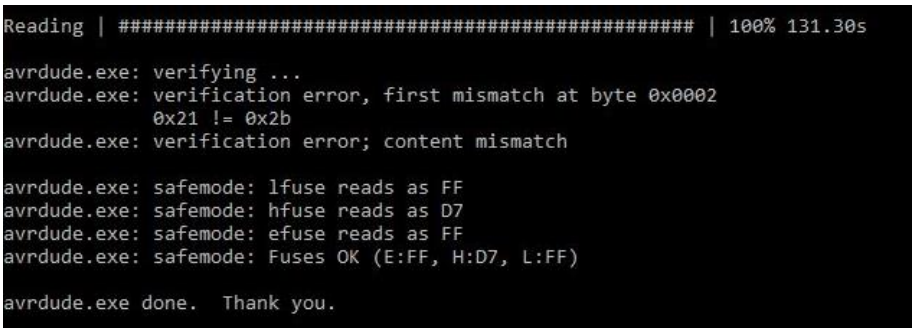


Figure 13.1: “verify error”

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.

13.1.4 How do I find a suitable ISP programmer?

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.

13.1.5 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 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).

13.1.6 Check the connection to the ATmega!

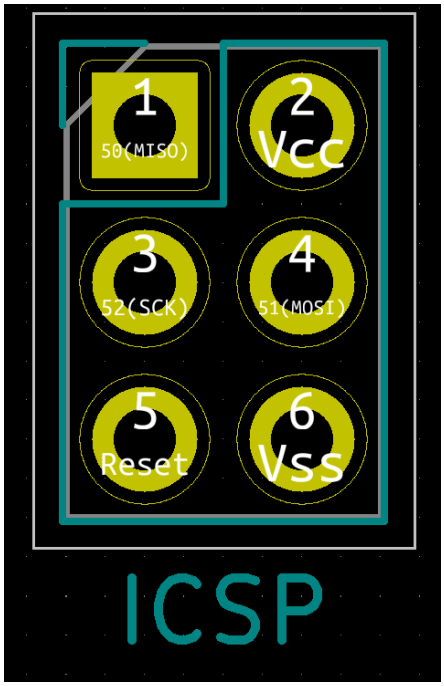
If the ATmega2560 has already been pre-assembled, this test is not necessary!

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):

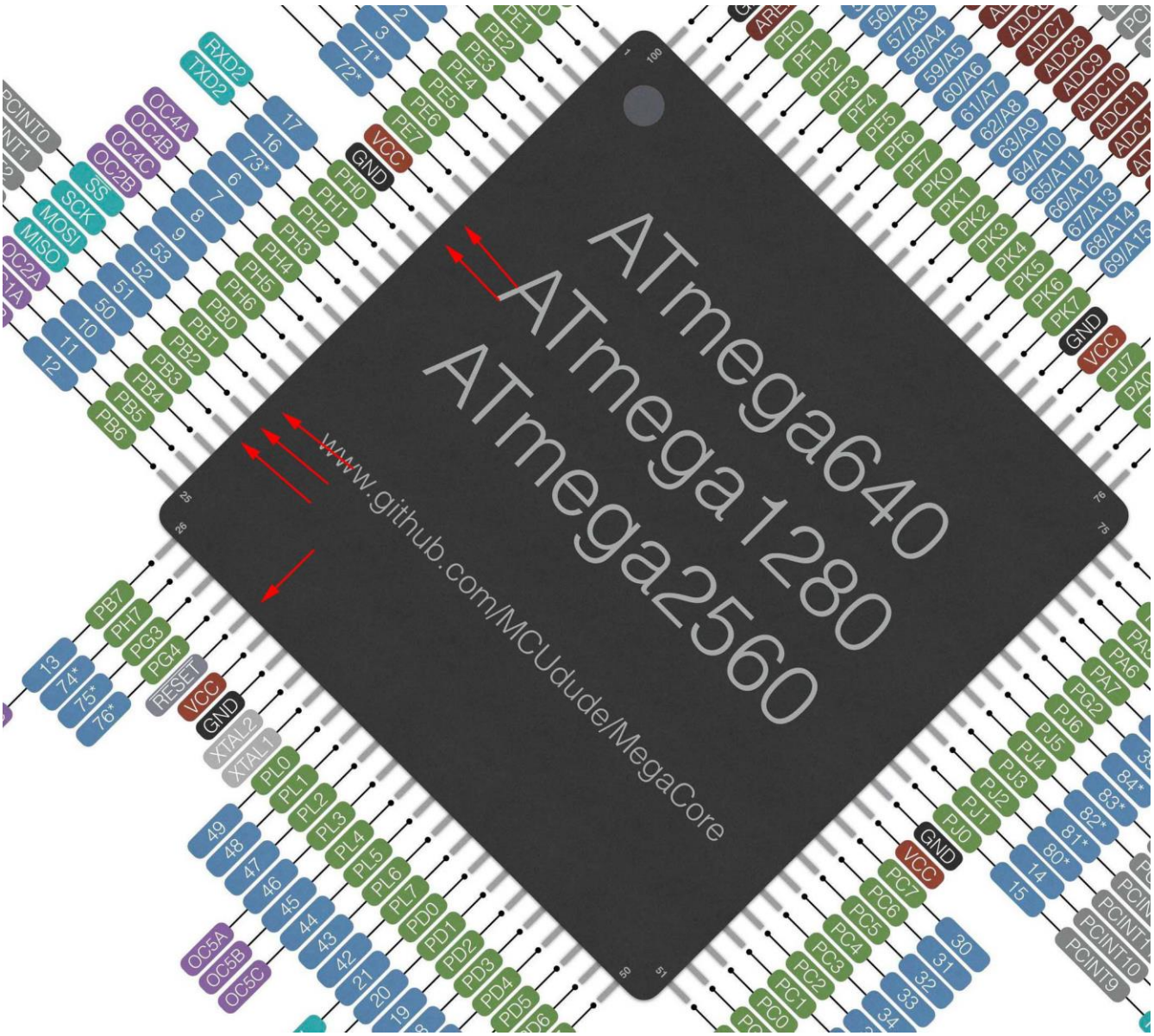


Figure 13.2: ATmega2560 pinout

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)?

13.1.7 “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.

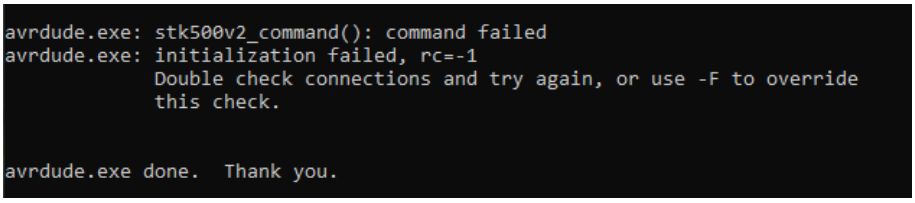


Figure 13.3: AVRDUDE – No communication with the ATmega2560

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 33 (XTAL2) of the ATmega2560. It sounds more complicated than it is.

First you need a clock source. 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.

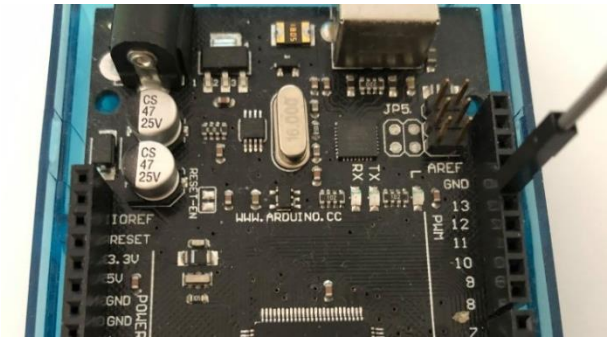


Figure 13.4: Pin 13 on the Arduino Mega

Now you connect pin 13 to XTAL2. The left 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).

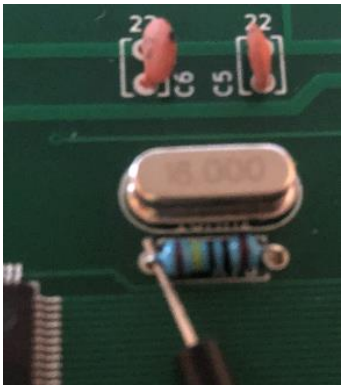


Figure 13.5: XTAL2

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" -B 4 -patmega2560 -cstk500 -PCOM5 -U
lfuse:w:0xff:m -U hfuse:w:0xdf: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.

13.1.8 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
```

13.2 Problems with the display

13.2.1 No text appears on the display

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

13.2.2 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.

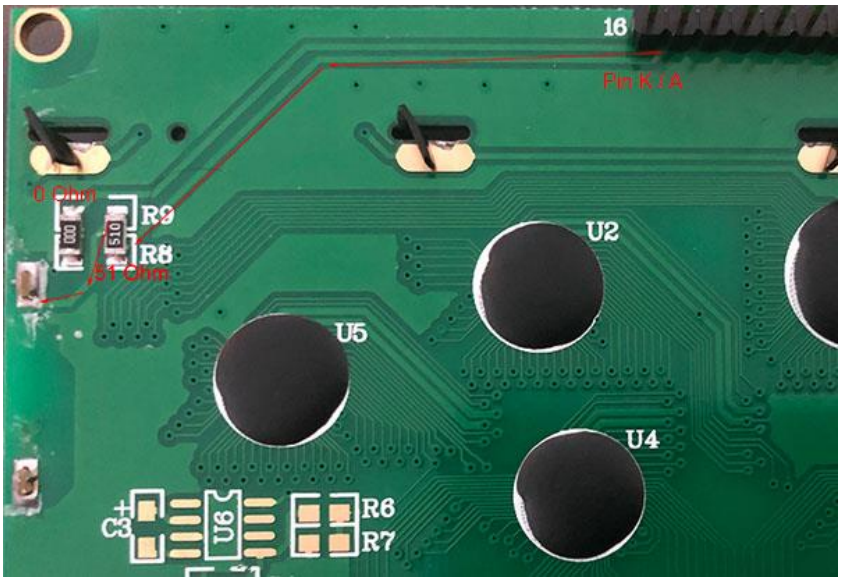


Figure 13.6: 51 Ohm series resistor to the backlight

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

13.2.3 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:

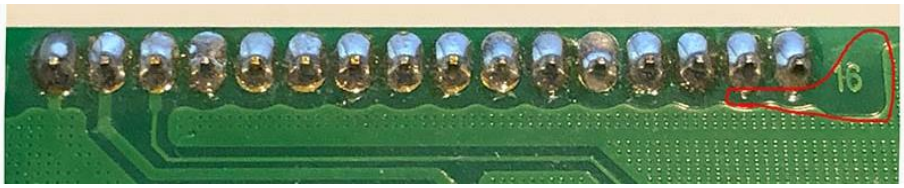


Figure 13.7: Pins 15/16 not connected

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

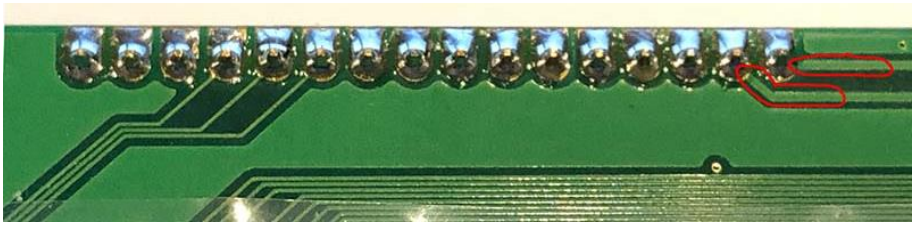


Figure 13.8: Pins 15/16 connected

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.

13.2.4 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:

Figure 13.9: Displayed text is ok

After several resets, lines can suddenly be mixed up:

Figure 13.10: lines are switched

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

Figure 13.11: 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 13.12: backside of the OLED PCB

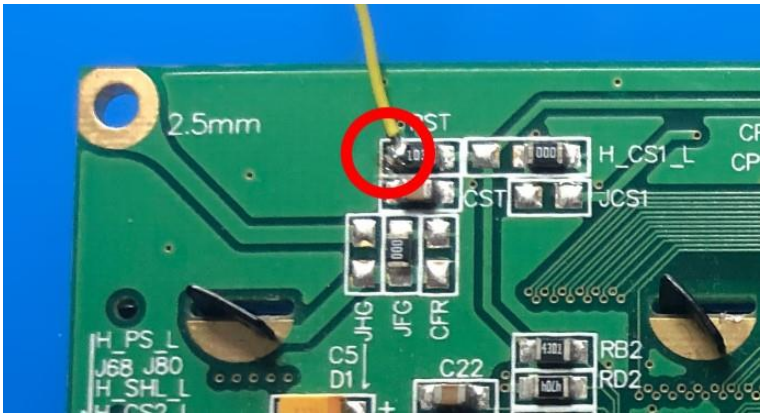


Figure 13.13: left side of the 10k resistor

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).

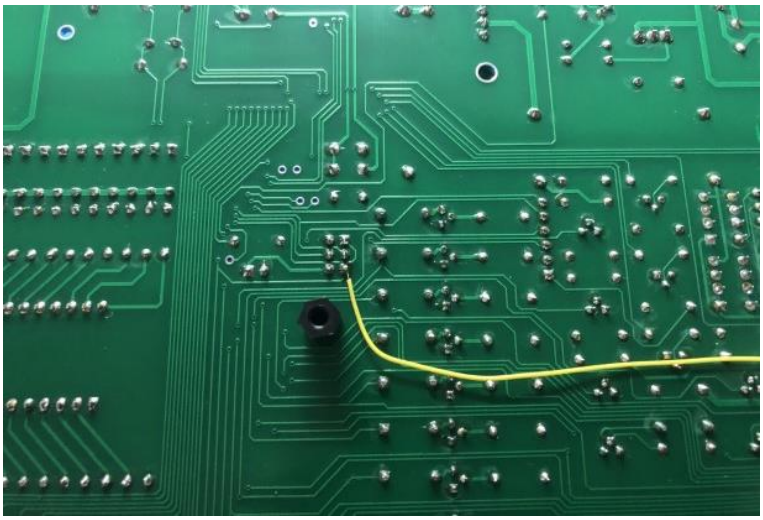


Figure 13.14: Reset at the ISP connector

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

13.2.5 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.

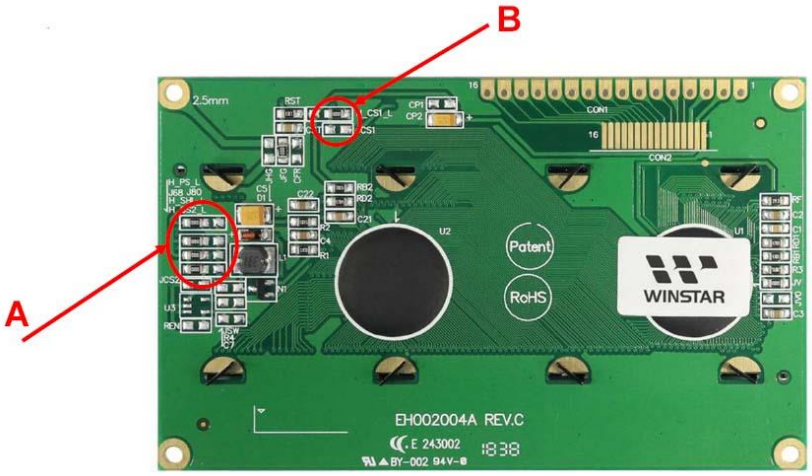


Figure 13.15: correctly set jumpers

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

13.3 Problems with the DC/DC module

13.3.1 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:

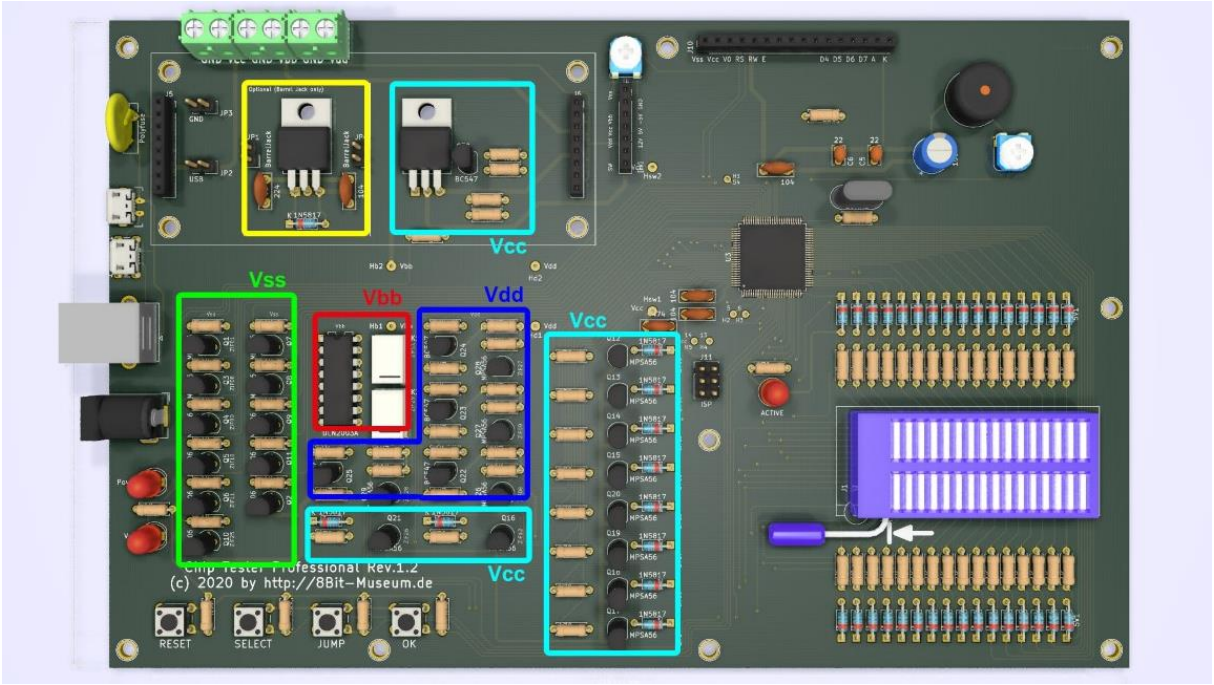


Figure 13.16: PCB overview

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).

13.4 Other Problems

13.4.1 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.

13.4.2 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 13.4.3.
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.

13.4.3 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, see sec. 11.2), required current at the inputs for bipolar ICs (e.g. standard TTLs, see sec. 0).

13.5 Selftest

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.

Figure 13.17: 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.

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

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

13.5.2 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.

13.5.3 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).

13.5.4 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):

Figure 13.18: 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.

13.5.5 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.

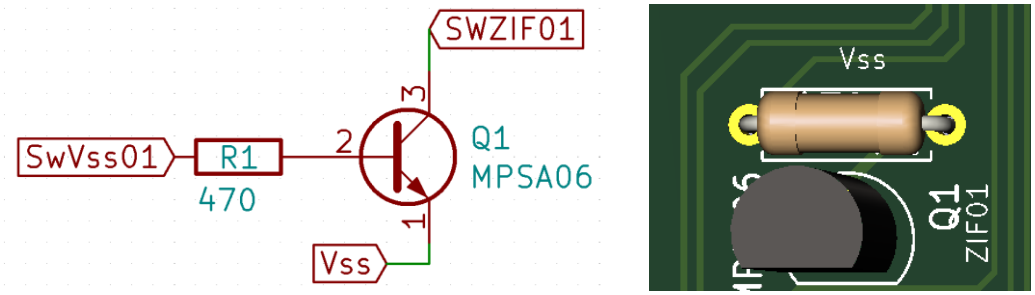


Figure 13.19: Switching of Vss, identification on the board

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.

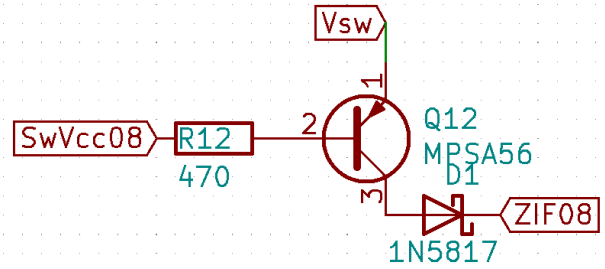


Figure 13.20: Switching Vcc

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.

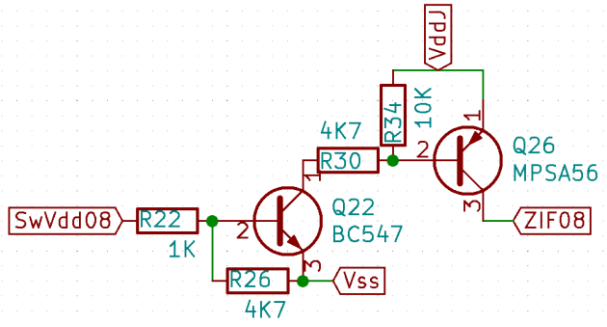
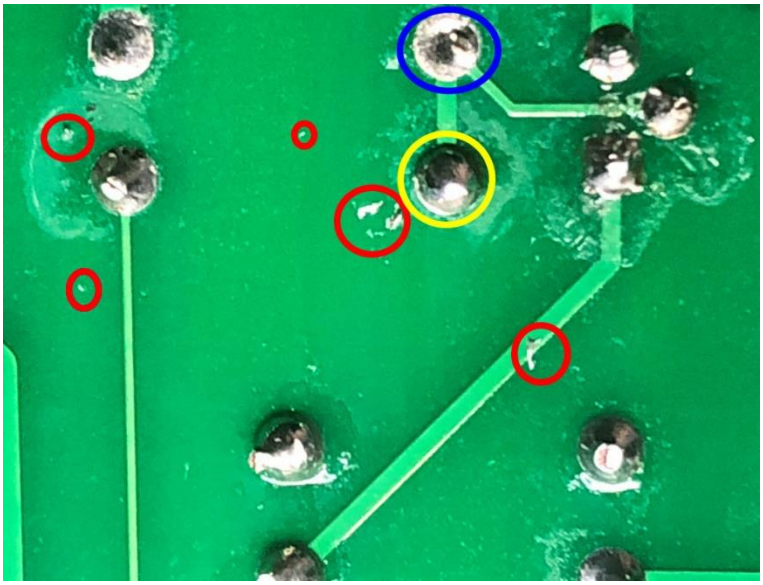


Figure 13.21: Switching Vdd

13.5.6 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)

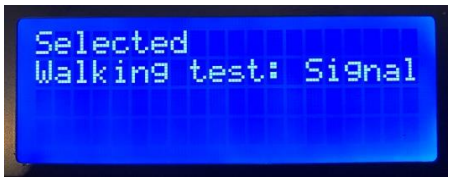
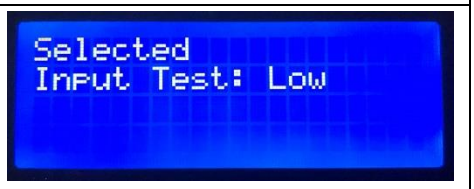
blue: too little solder

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).

13.6 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.	
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.	
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: 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.	
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.	
Input Test: High	When this test is started, any pin can be connected to Vcc. The pin is shown in the display.	

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.	

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).

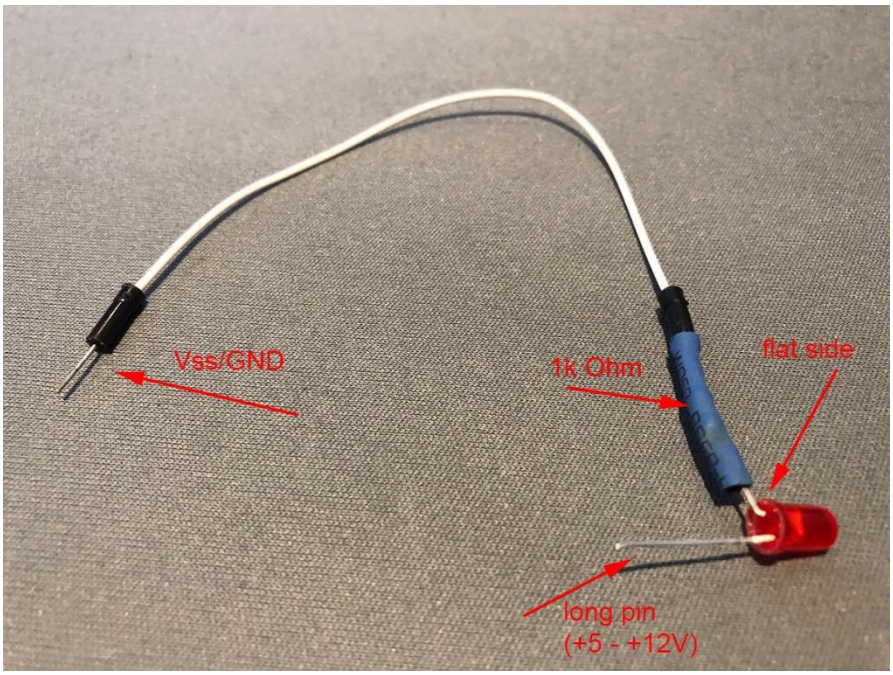
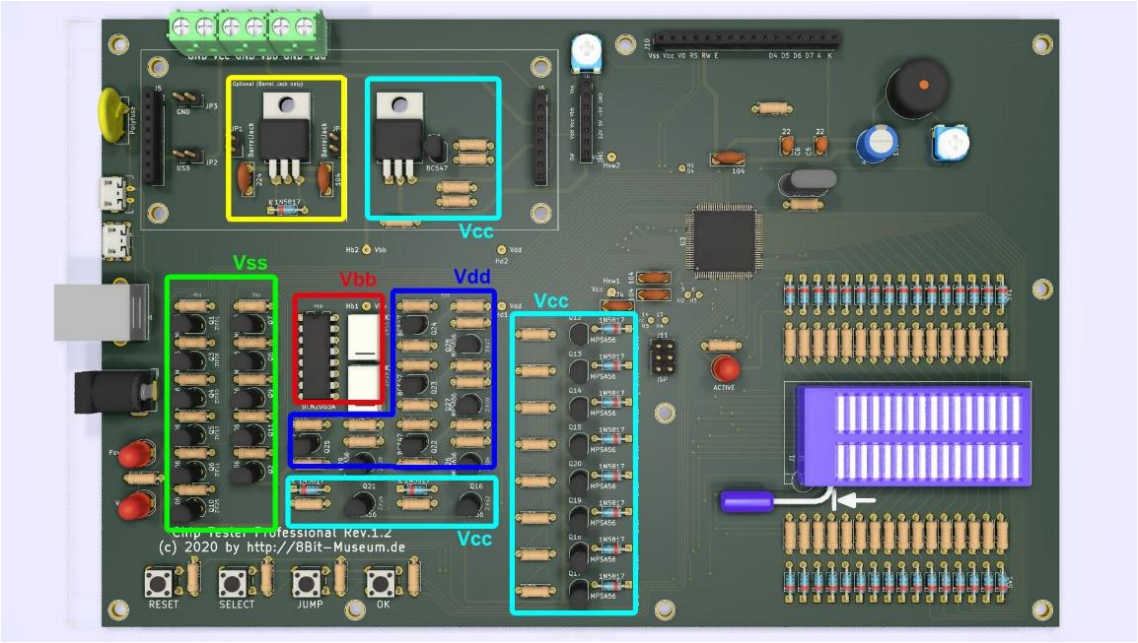


Figure 13.22: A simple logic tester

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:



14 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.

14.1 DIAMEX PROG-S2 ISP-Programmer für AVR / STM32 / NXP / LPC

Figure 14.1: 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.

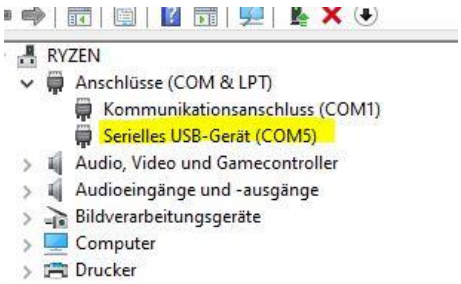


Figure 14.2: Diamex in the Device manager (here: COM5)

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 14.3: 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”):

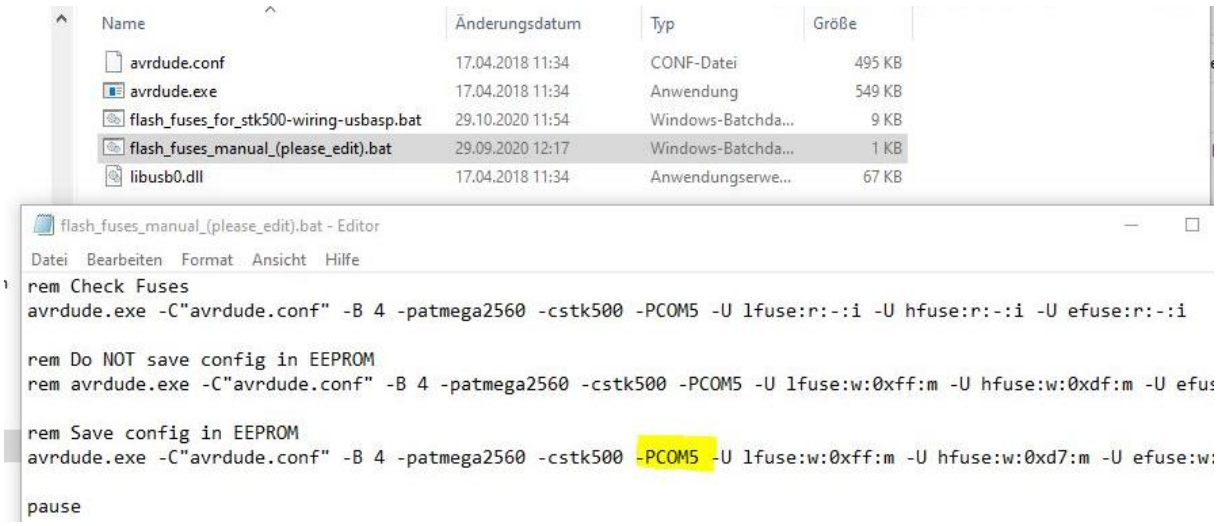


Figure 14.4: Prepared batch file

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“):

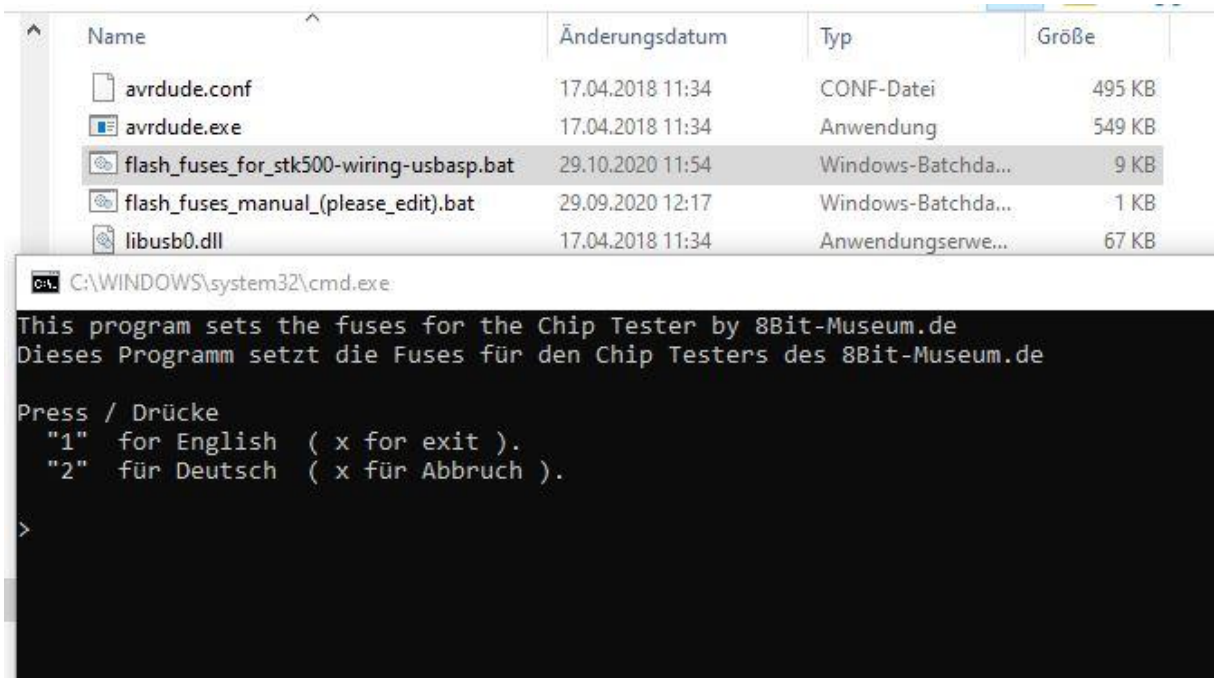


Figure 14.5: Interactive batch file

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.

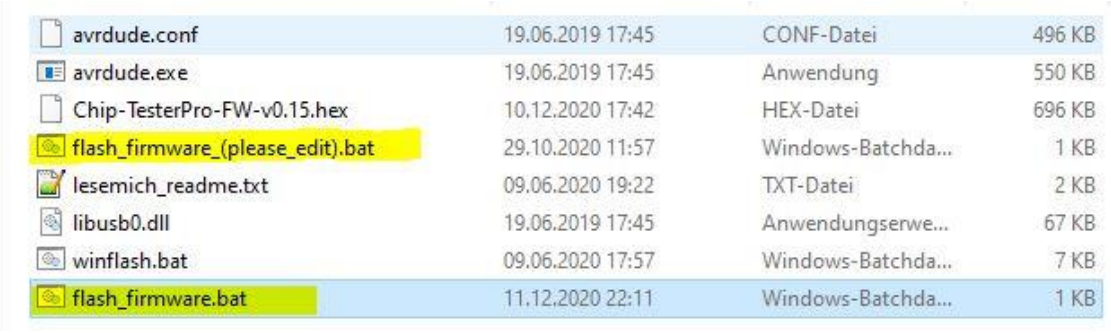


Figure 14.6: Flashing the firmware

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

14.2 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)"[8] is used to program the ATmega2560.

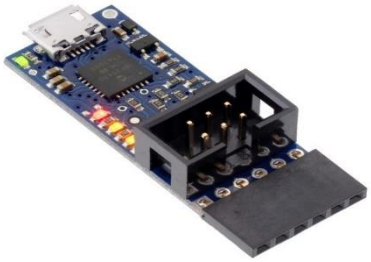


Figure 14.7: Pololu USB AVR Programmer v2.1

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.

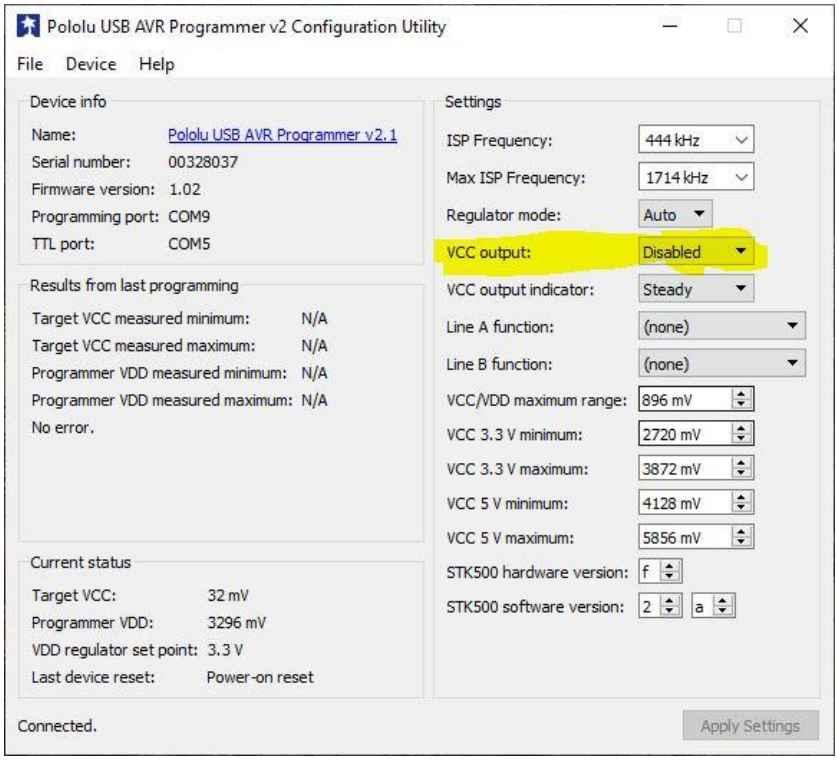


Figure 14.8: Vcc output

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

14.2.1 Settings AVRFuses

Menu: → AVRFuses → Preferences

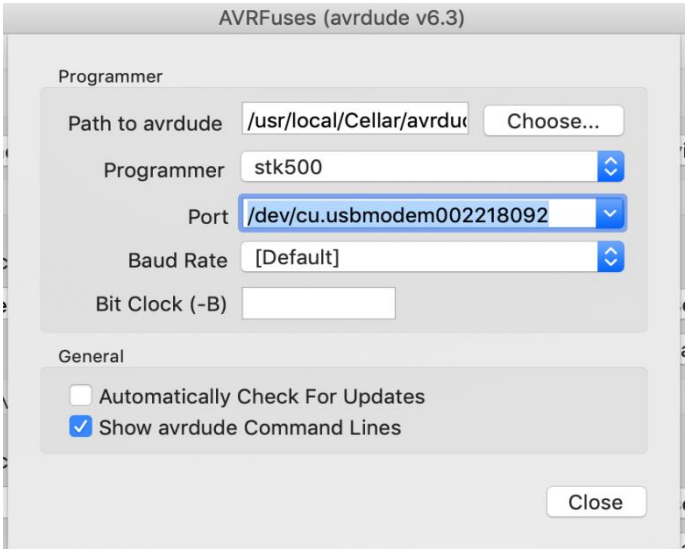


Figure 14.9: AVRFuses

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

14.2.2 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.

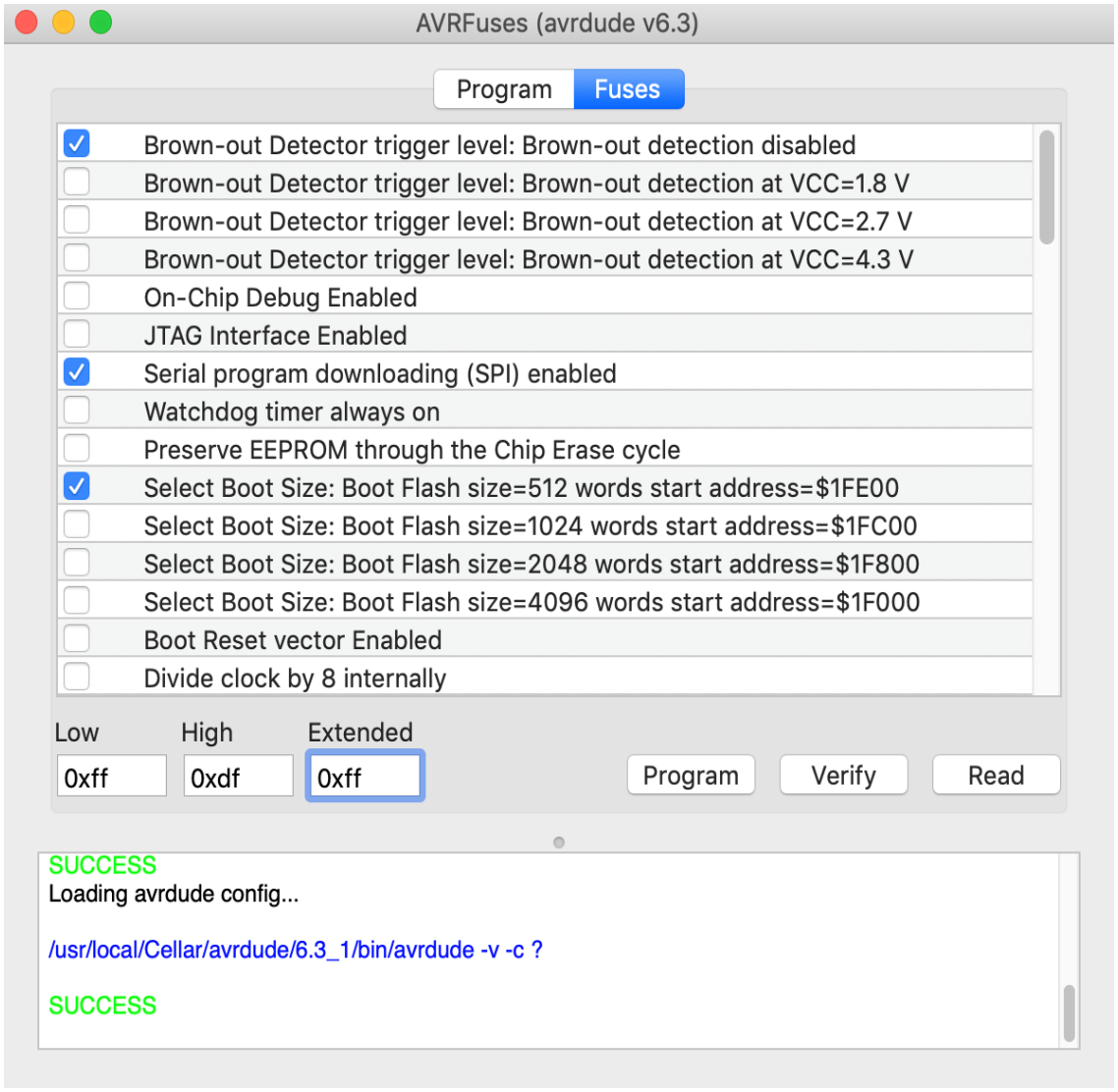


Figure 14.10: Setting the fuses with AVRFuses

Write the values into the ATmega2560 with "Program".

14.2.3 Flash Firmware

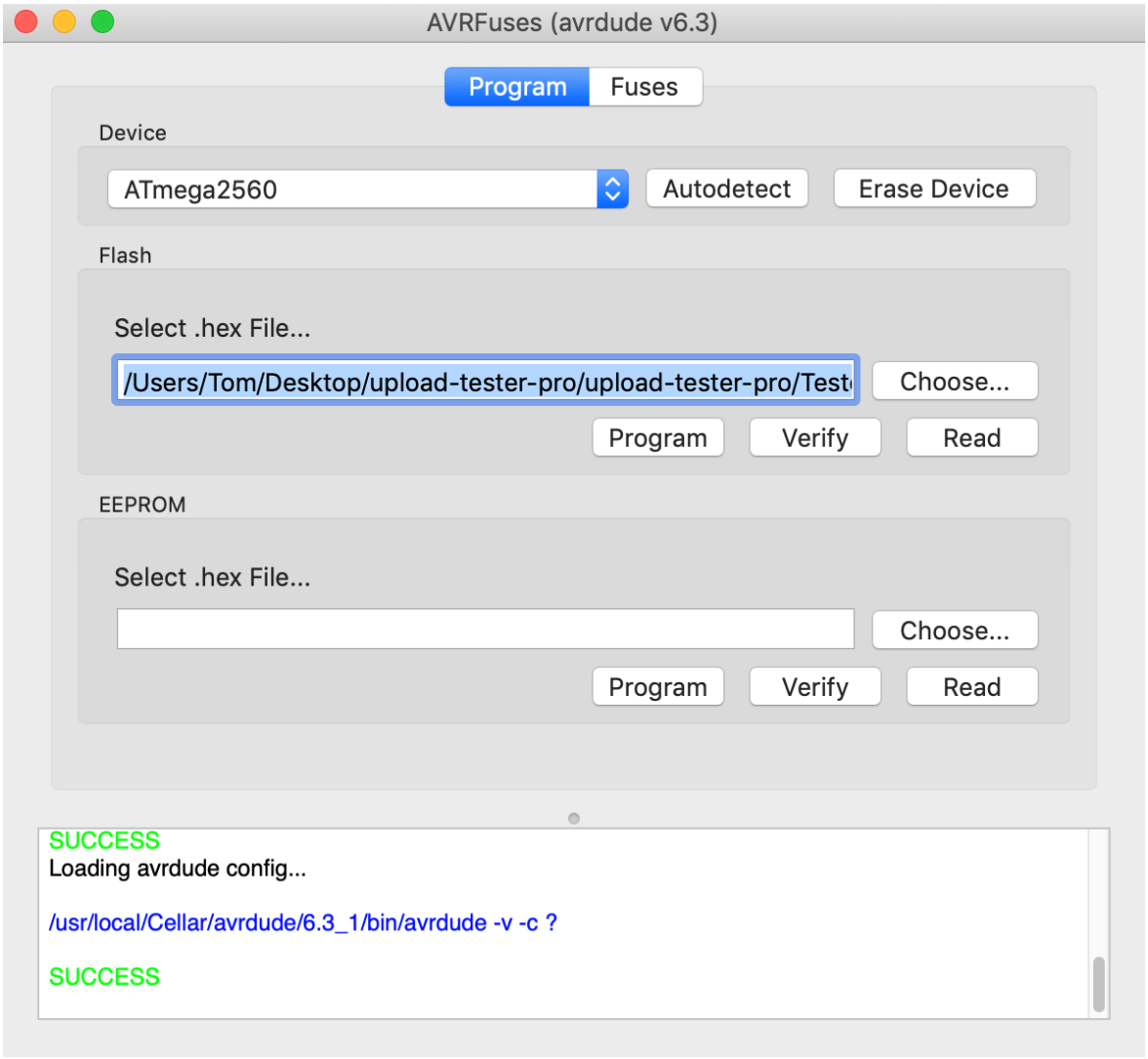


Figure 14.11: Programming with AVRFuses

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

Write the values into the ATmega2560 with "Program".

14.3 Pololu USB AVR Programmer v2.1 / Microsoft Windows

The display must not be attached to the memory tester. The program "AVRDUDESS 2.4"[9] is used to program the ATmega2560.

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.

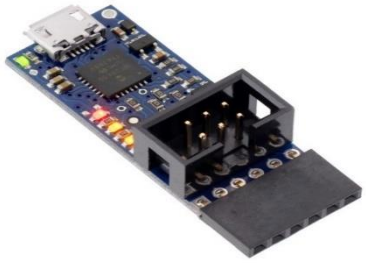


Figure 14.12: Pololu USB AVR Programmer v2.1

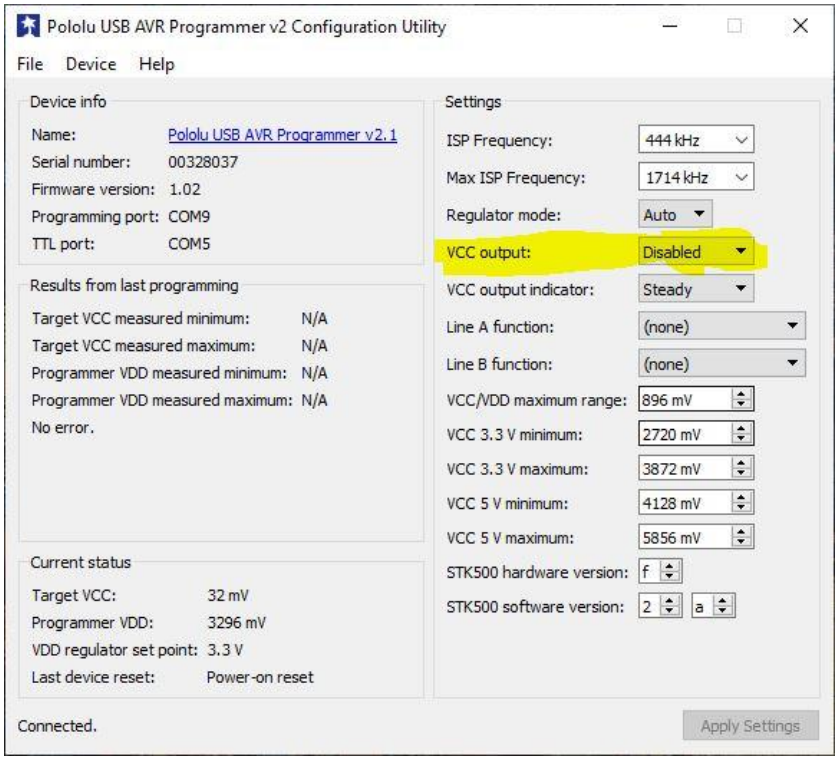


Figure 14.13: Vcc output

The operation of AVRDUDESS is relatively easy:

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

1. Select the programmer (Atmel STK500 compatible)
2. Select the port used (Pololu creates two virtual ports, usually the lower port should be correct)
3. Select the MCU (ATmega2560)
4. Select the firmware to be programmed under Flash (HEX file) with the mode "Write" and format "Auto"
5. Define the fuses "L", "H", "E", "LB"
6. Then "Set fuses" should be checked and these should be set with "Write" (under "Set Fuses").
7. If the fuses are set, the firmware can be written to the ATmega with “Go” (under “Flash”).

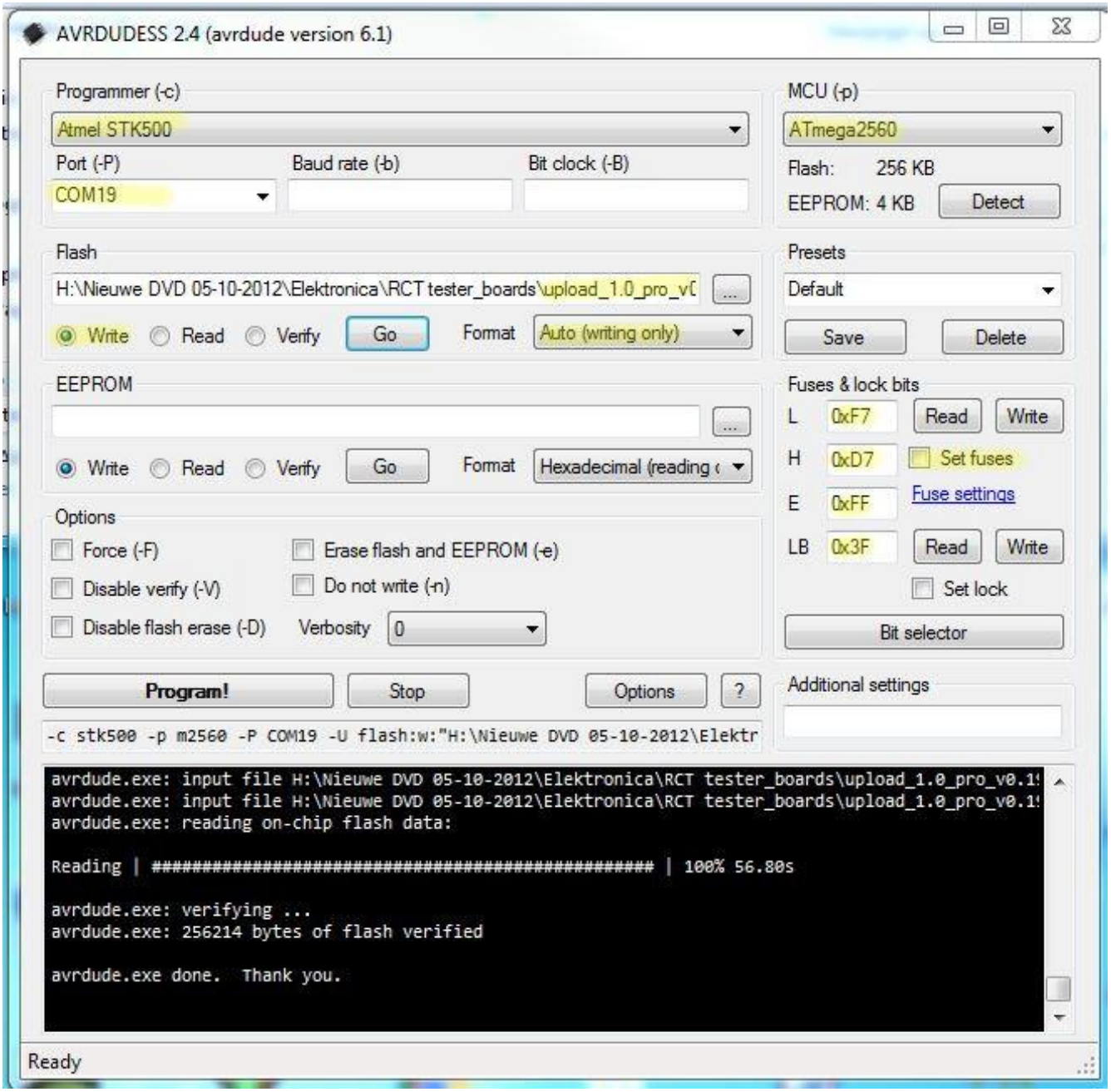


Figure 14.14: AVRDUDESS

14.4 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:

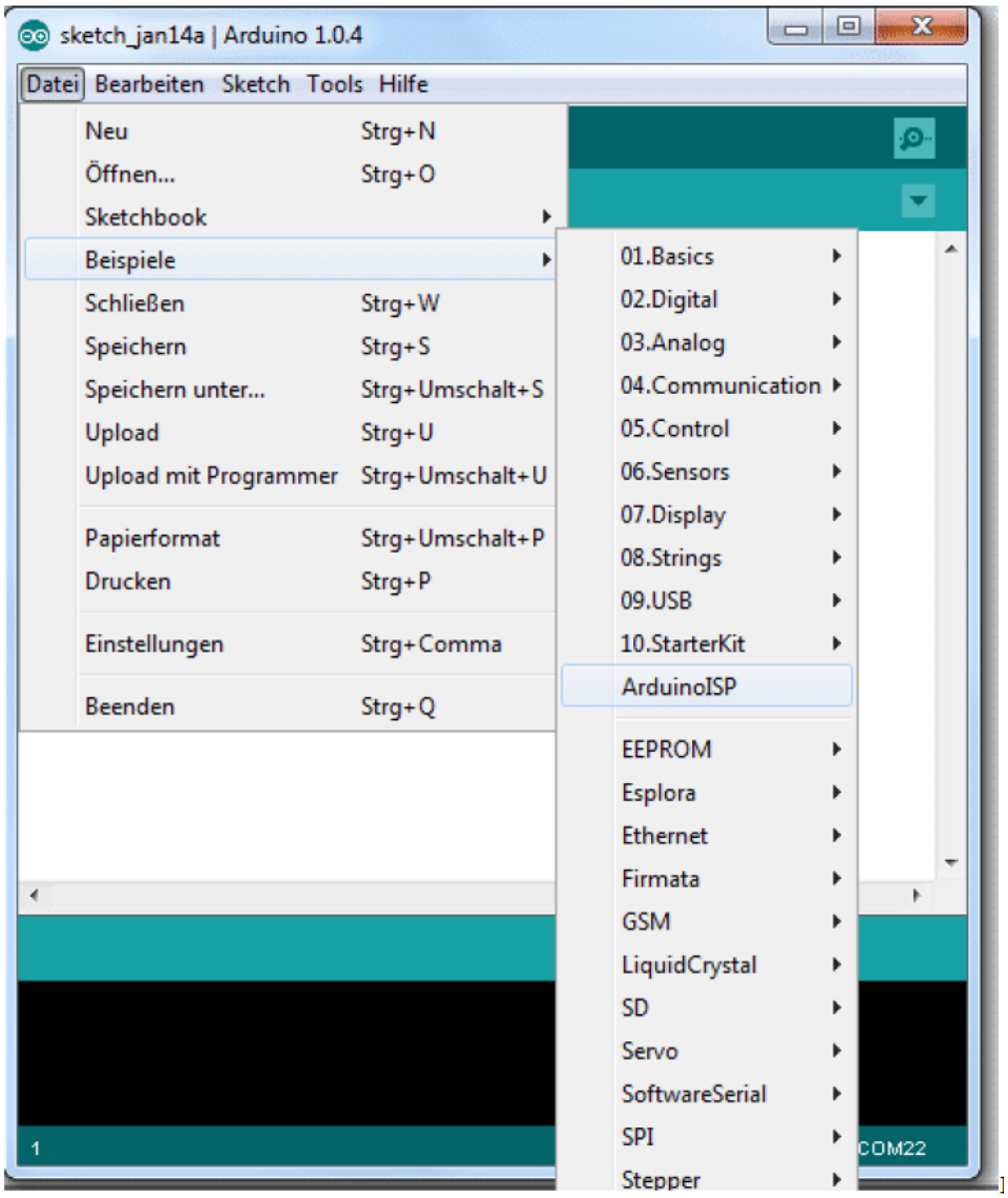
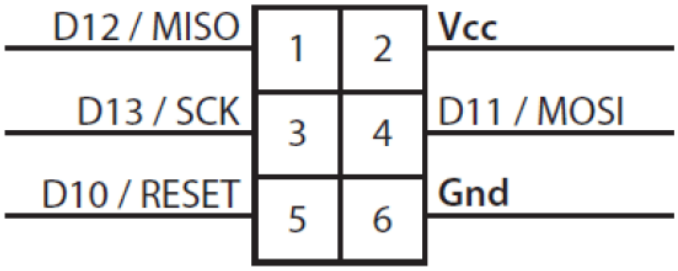


Figure 14.15: Arduino IDE

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.



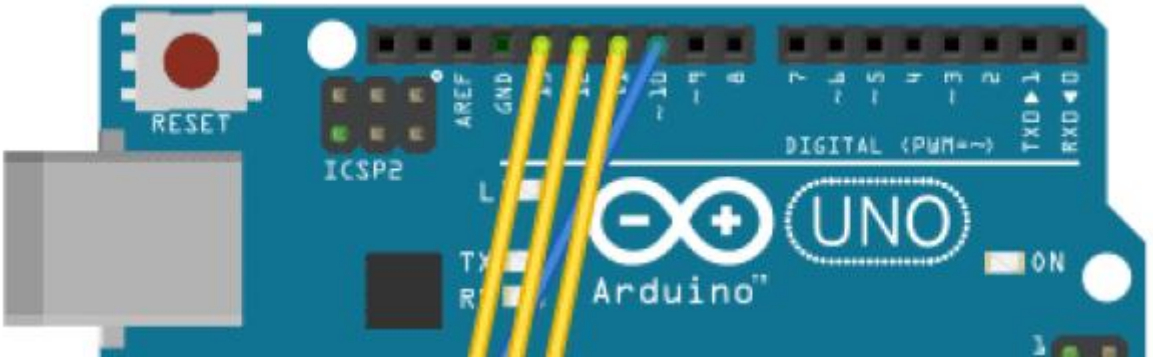
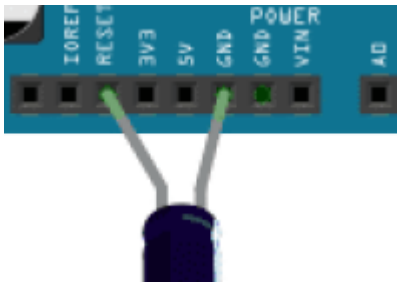


Figure 14.16: Using an Arduino Uno as programmer

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.

14.5 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 14.17: 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".

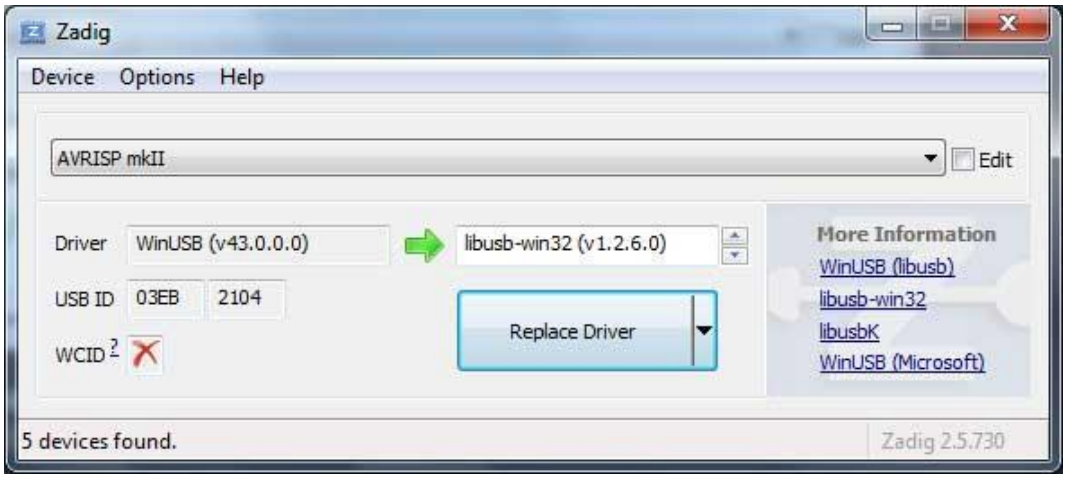


Figure 14.18: ZADIG

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

14.6 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.

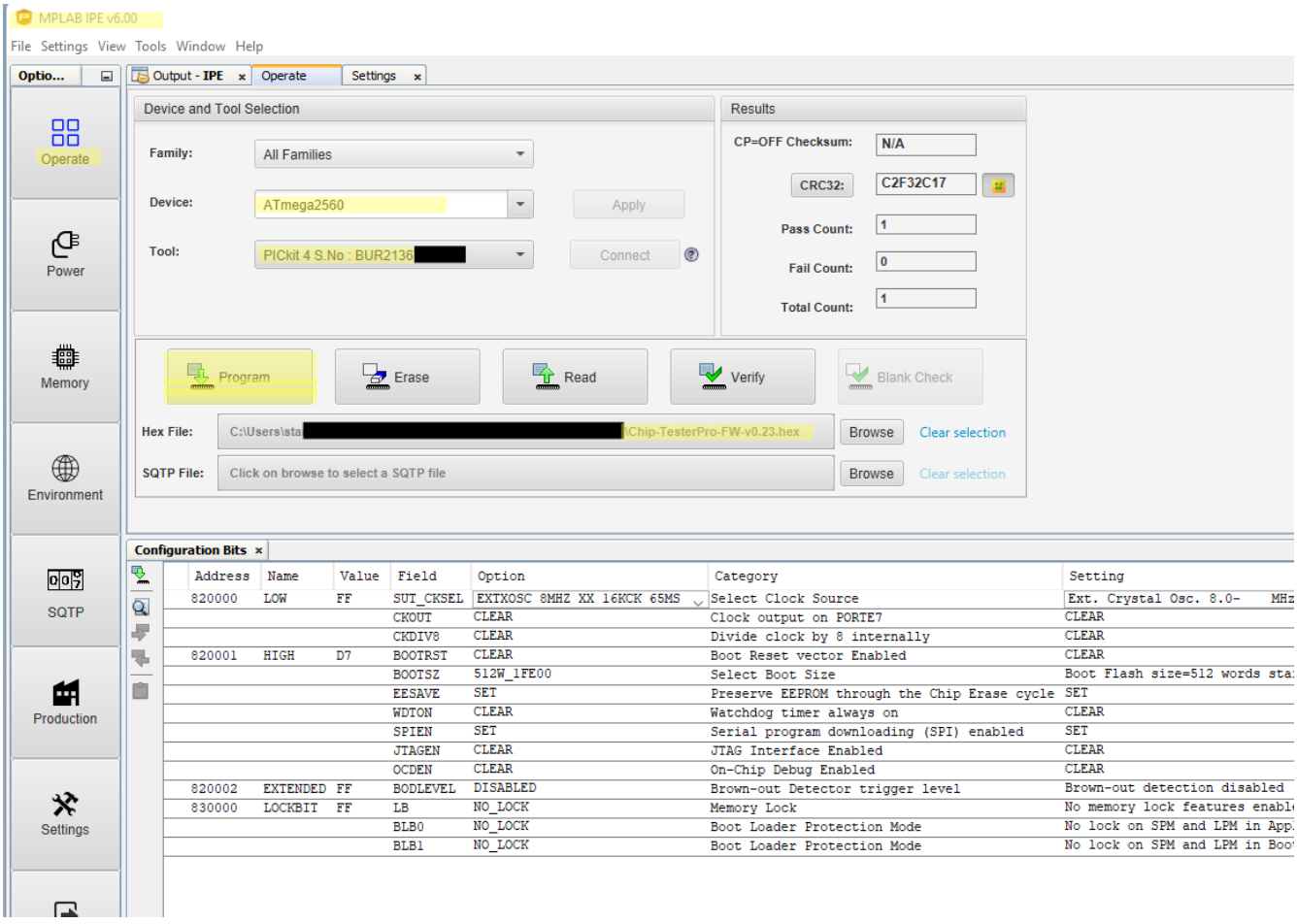
14.7 Microchip MPLABX with IPE Application / PICkit4

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.

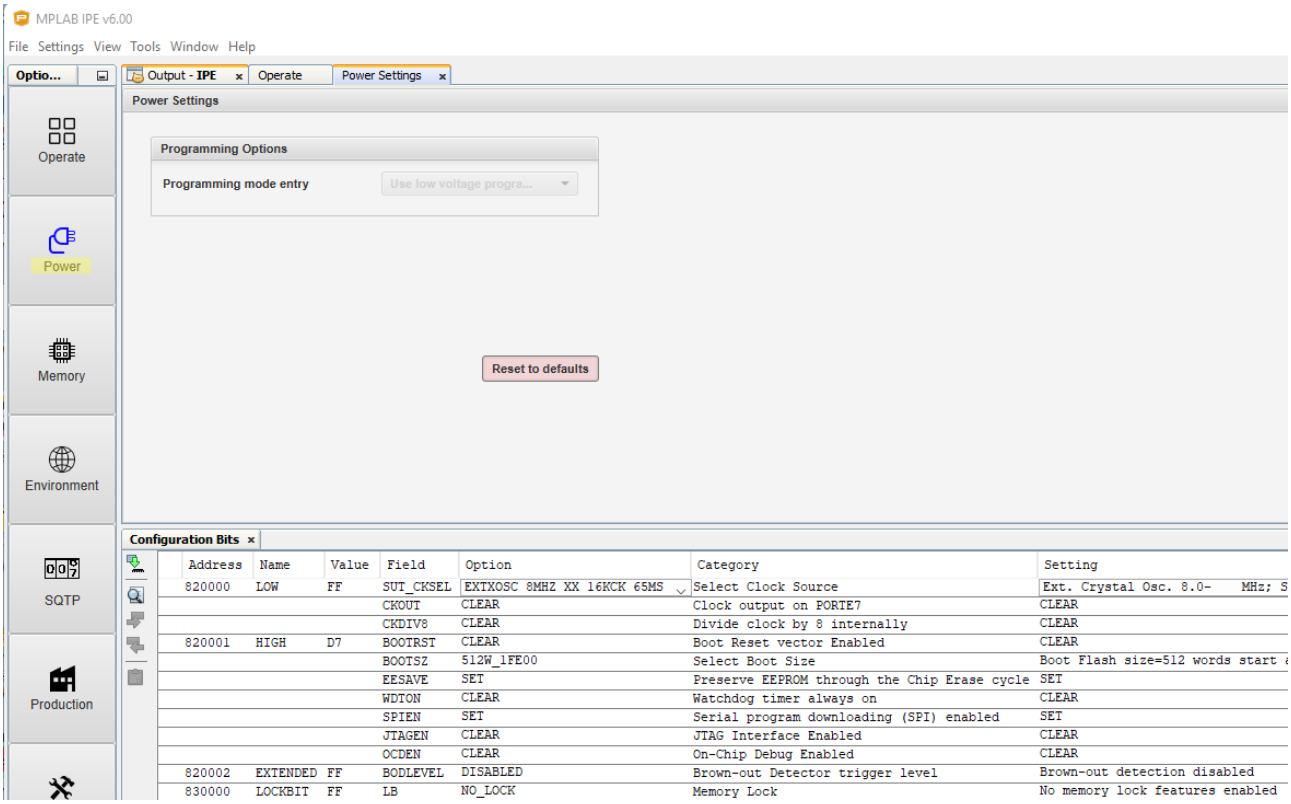
Operate tab:

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



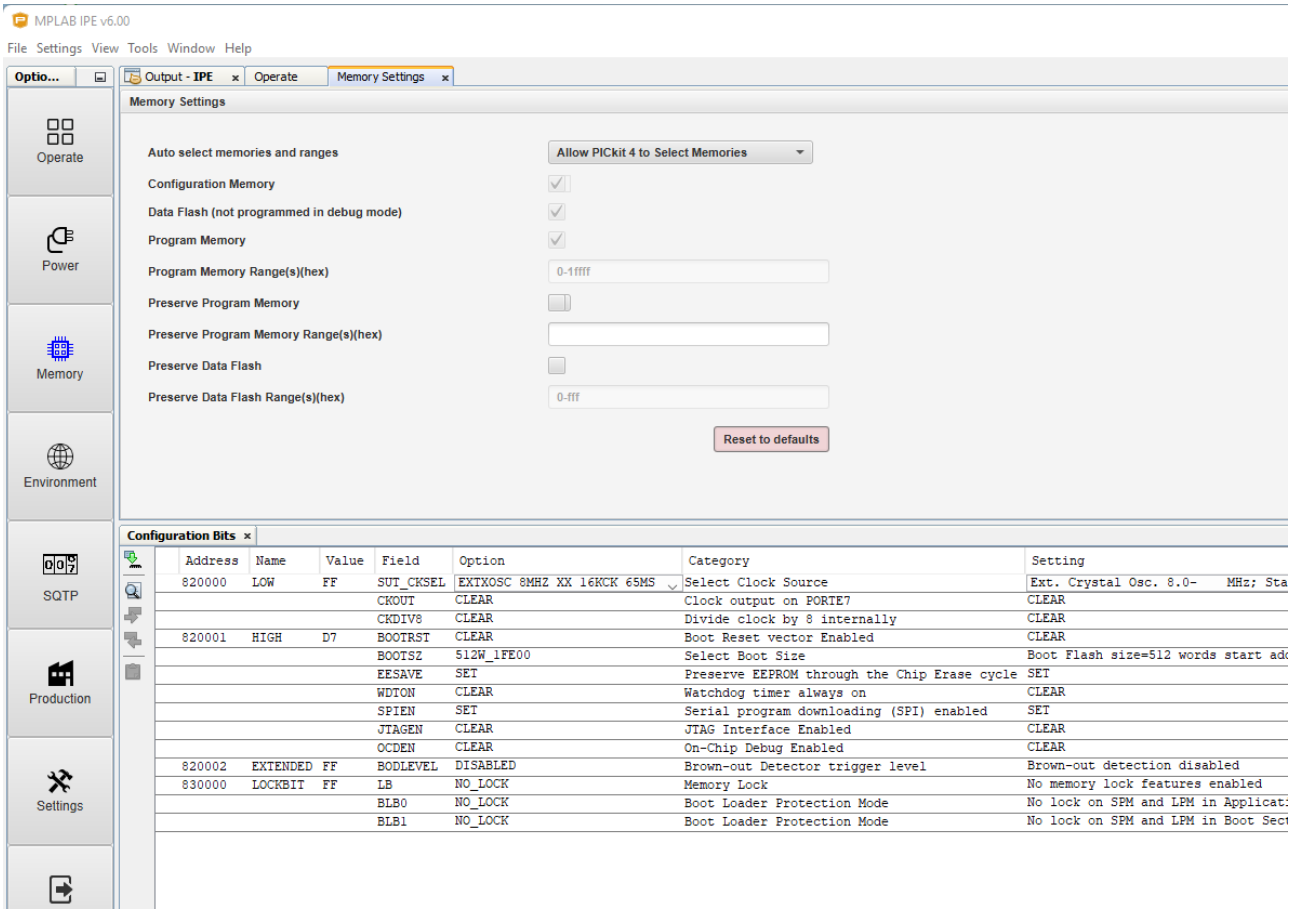
Power tab:

Power is greyed out.



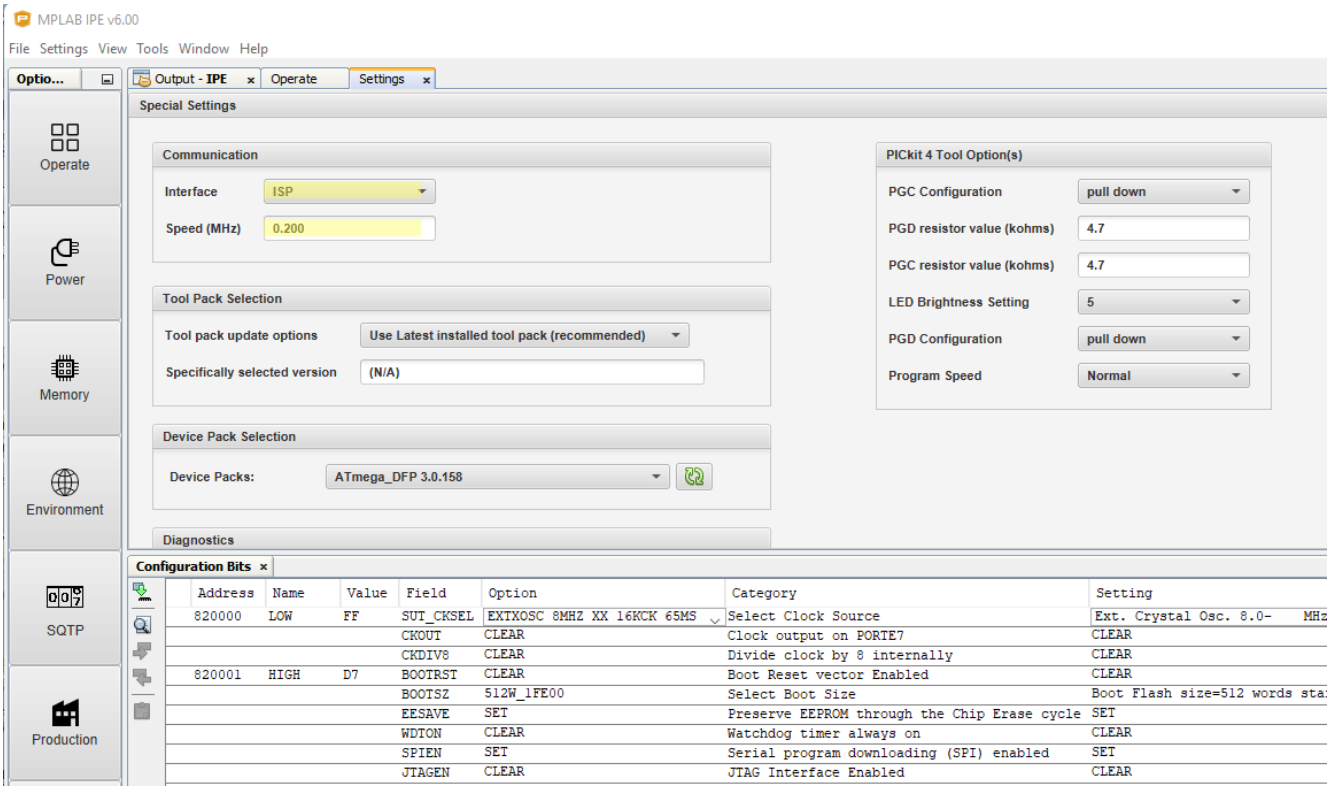
Memory tab:

Select “Allow PICkit 4 to Select Memories”

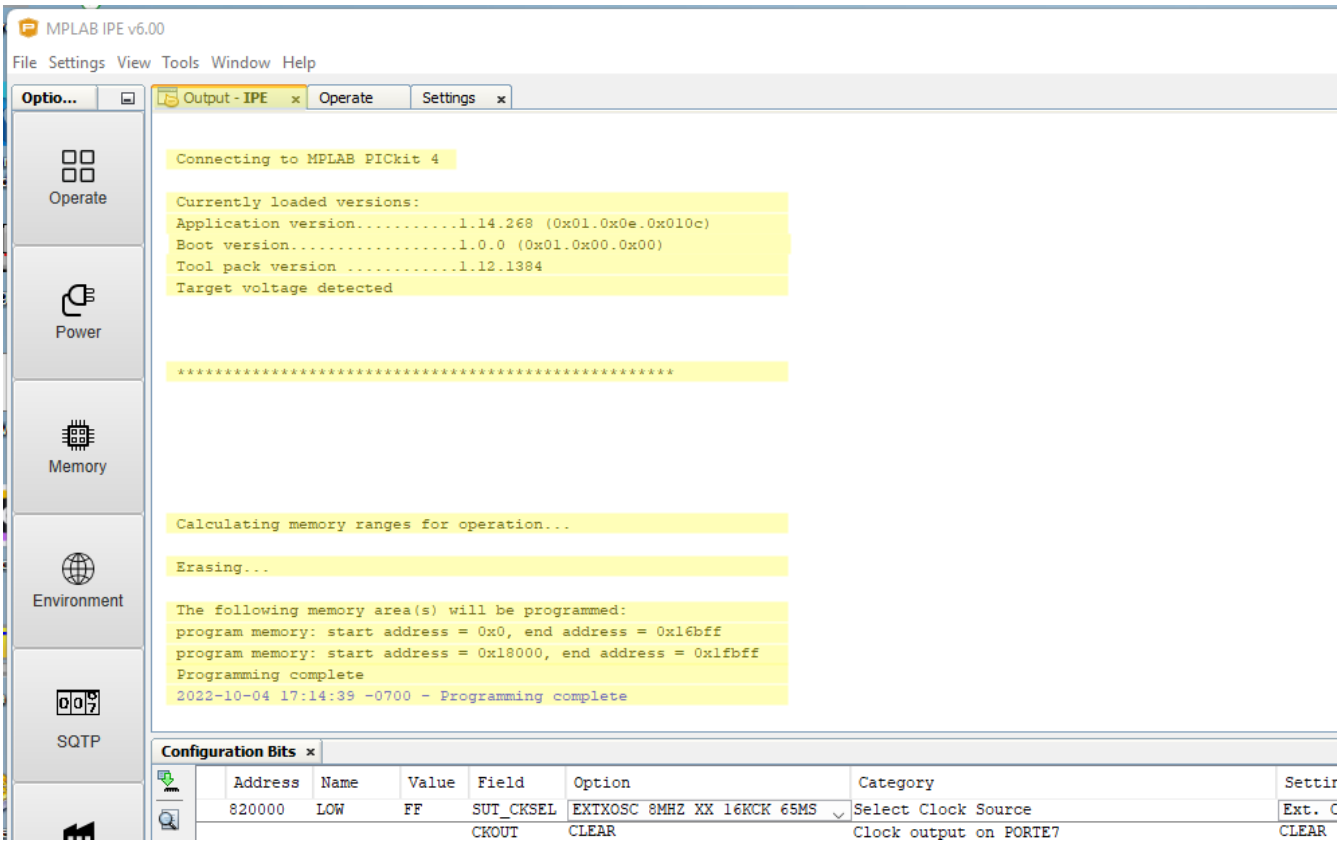


Settings tab:

Choose ISP interface and a speed of 0.200 MHz.



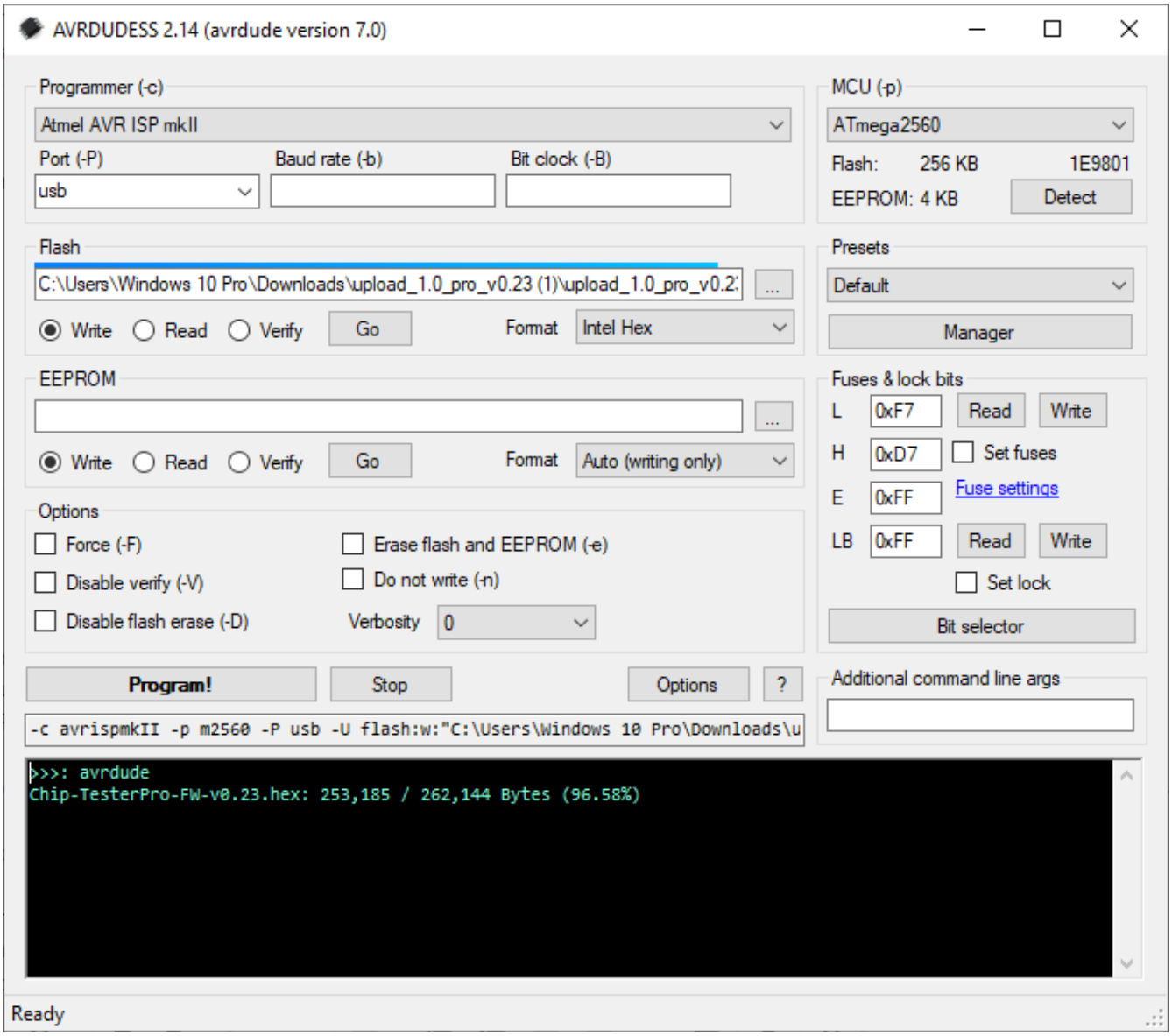
Output IPE tab:



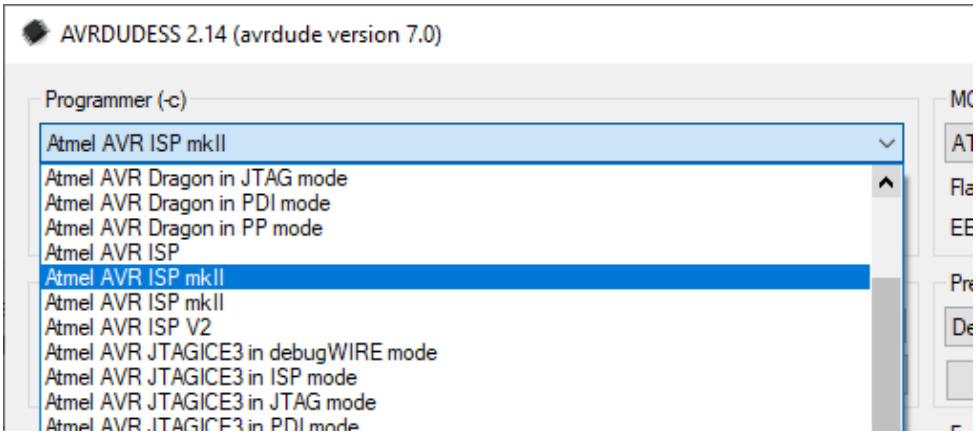
14.8 OLIMEX AVR-ISP-MK2

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).



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".



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.

14.9 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

15 PAL Analysis and Synthesis with the RCT

In chapter 6.12 it is described how a PAL or GAL can be dumped. The underlying logic equations can be reconstructed with very little effort if all possible input combinations are iterated and the outputs are logged and then analyzed. In this way a copy can be made that is logically identical to the original.

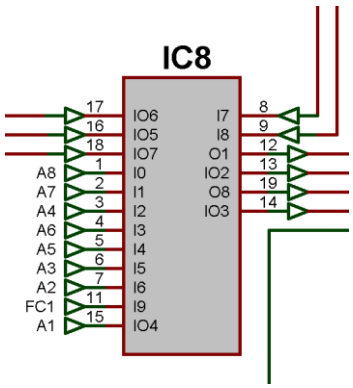
This chapter describes how to do this. The following way has been documented with help of Christophe Escurat. There might be easier solutions.

Following software is required:

- **PALASM 1.5** used to create a JEDEC file
  http://www.pldworld.com/_otherplds/palasm/-engr.uky.edu/_melham01/ee481/software.htm
  It is quite old, it works under MS-DOS, so use **DOSBOX** to execute it.
- **DOSBOX** to execute PALASM
  https://www.dosbox.com/download.php?main=1
- **Simple Solver** to reduce the logic equations
  https://www.simplesolverlogic.com/

Step 1: Analysis

First it is necessary to specify which pins are input and which are outputs. To do this the easiest way is to check the schematics or, if it is not available, do reverse engineering. In this example we analyze a PAL16L8.



The I/O pins are defined in the “PAL (20 pins)“ menu as follows:

```
12345678901234567890
IIIIIIIIIGIOOOIIIIOV
```

Step 2: Saving the PAL content

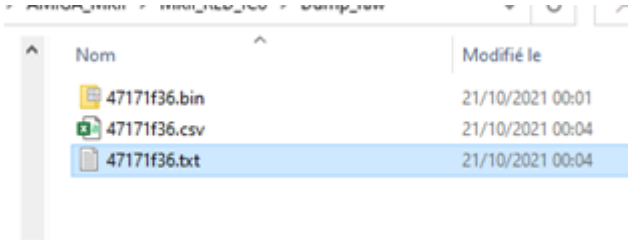
The TI16L8-25 in our example is a PAL16L8 in a PLCC package. We use a PLCC-20 to DIP-20 Adapter to place the PAL into the Retro Chip Tester. Select the menu “Selected (PAL) / PAL 20pin” and ensure that the SD card board is installed and activated in the configuration of the RCT. Set the PIN configuration to the pattern mentioned above. Press and hold the OK button for two seconds.

The content will be written to the SD card now. This may take a while depending of the number of combinations. Time to take a coffee. The RCT checks $4000 = 16384 combinations.

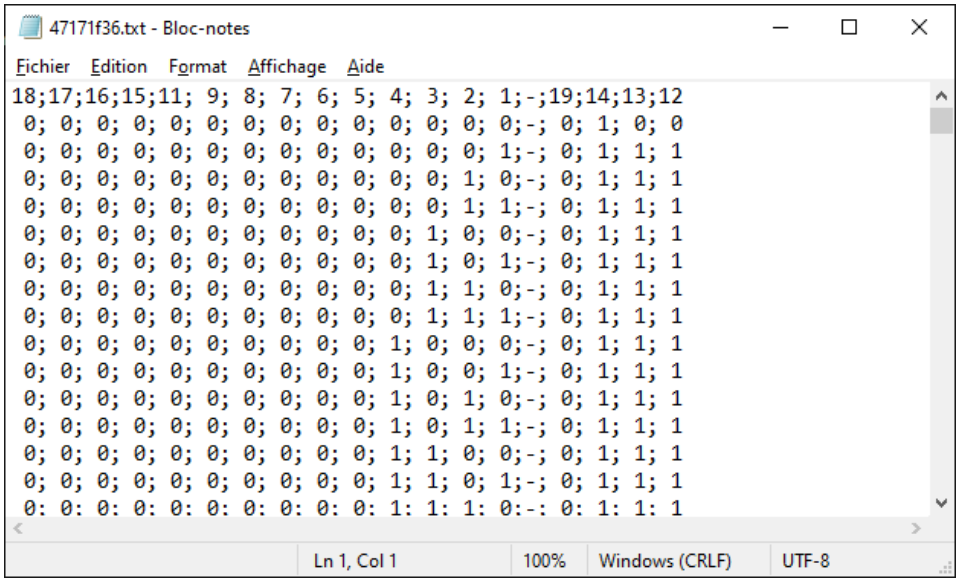
The saving is done in two steps: In the first step, the RCT creates a binary dump file, in the second step this file is converted into a CSV file that can be opened by Excel or a text editor.

Step 3: Check the dump file

After some time (it can take several minutes) the conversion is done. Now check the content of the SD card. Copy the BIN and CSV file to your computer, do a copy of the CSV file and rename it to TXT.



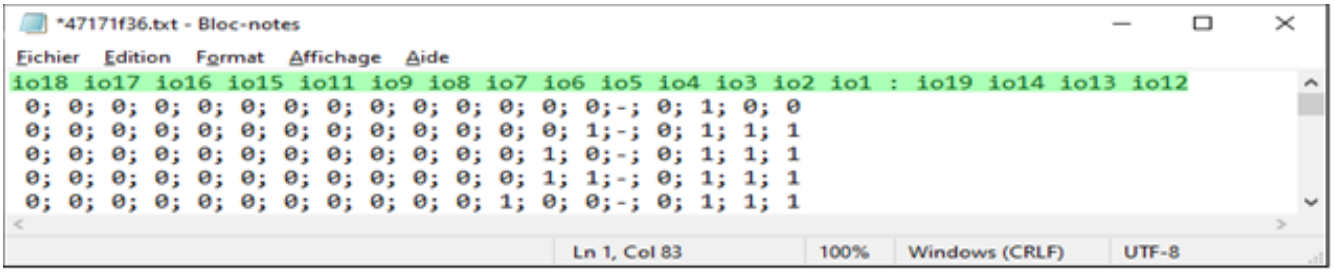
In order to use the text file with a different program, we have to do some changes. Open the file with a text editor and change the first line.



Change the following in the **first line only**:

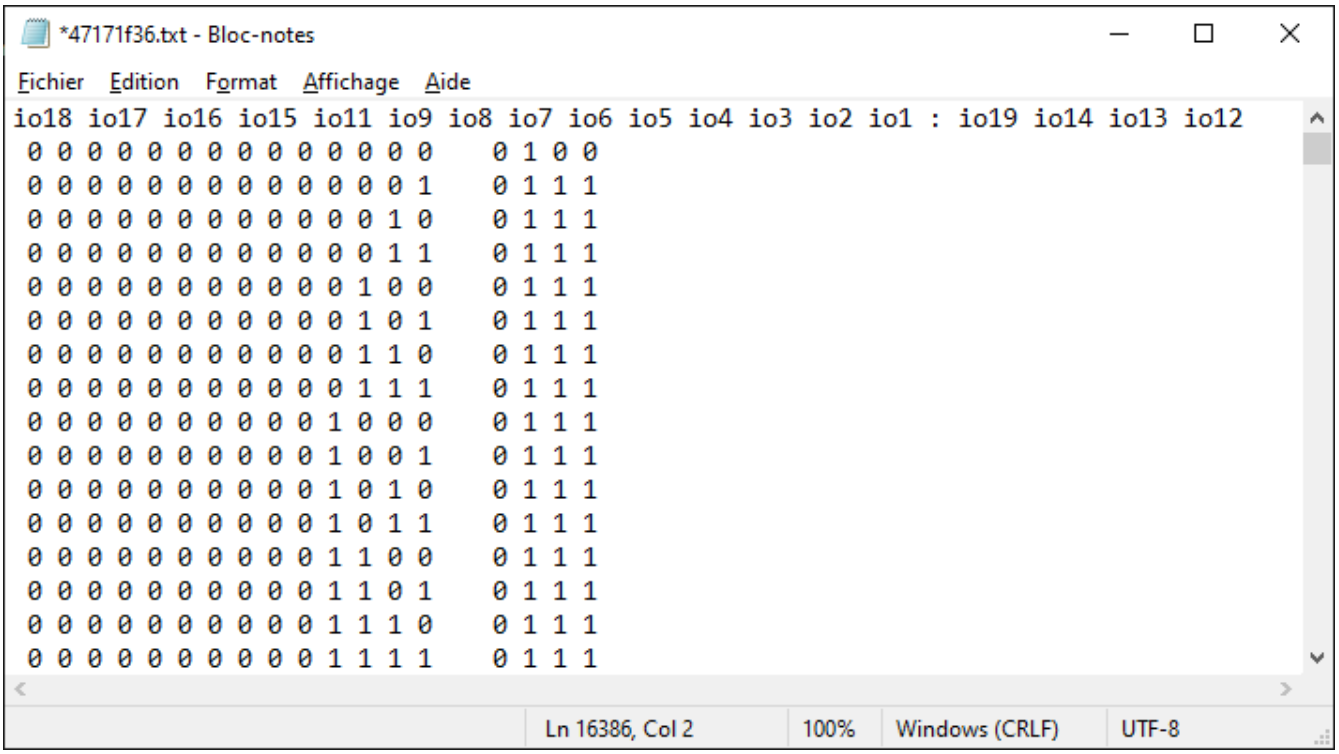
- Add the prefix “io” to the numbers of the I/O pins.
- Remove all ";" (semicolon) so that only one space remains between the values.
- Replace the "-" (minus) with a ":" (colon).

The modifies line looks like this:

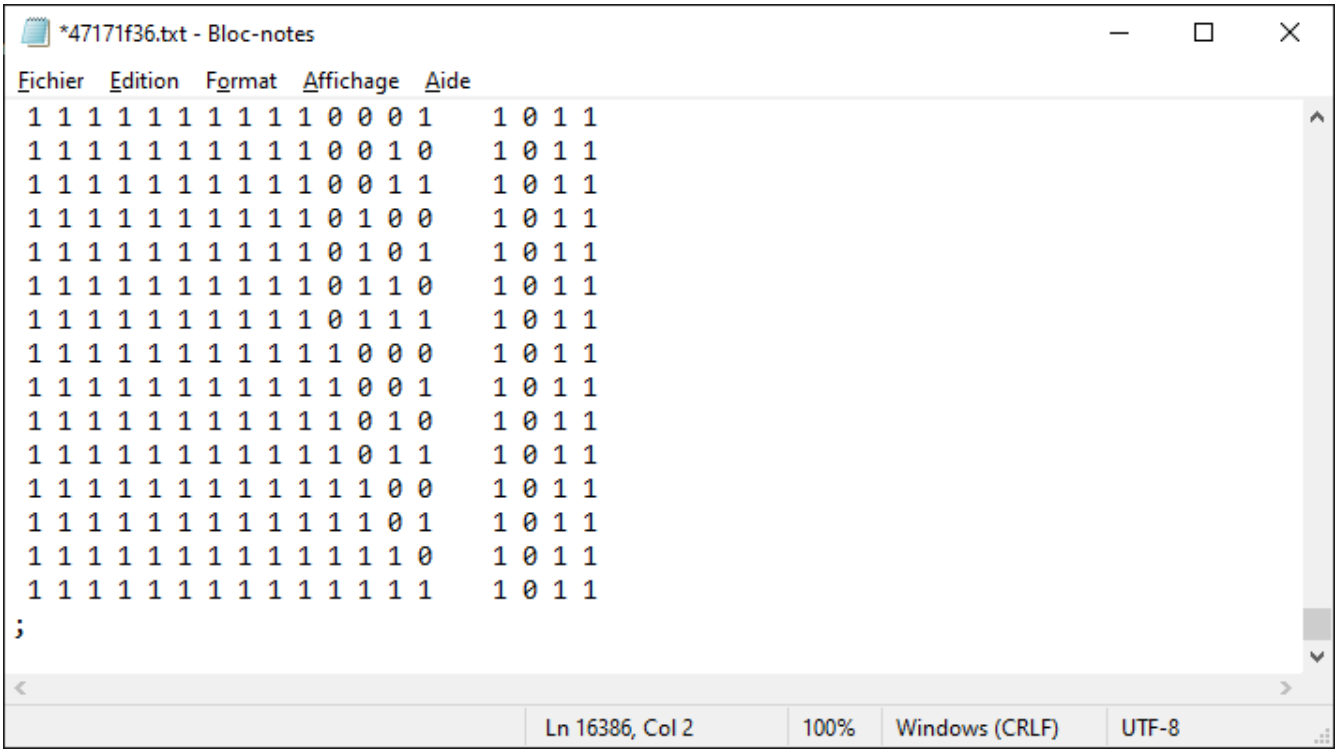


Change the remaining lines as follows:

- Remove all ";" (semicolon).
- Replace the "-" (minus) with a tabulator.



Finally add a ";" (semicolon) to the end of the file and save it.



Step 4: Read the truth table and reduce the equations

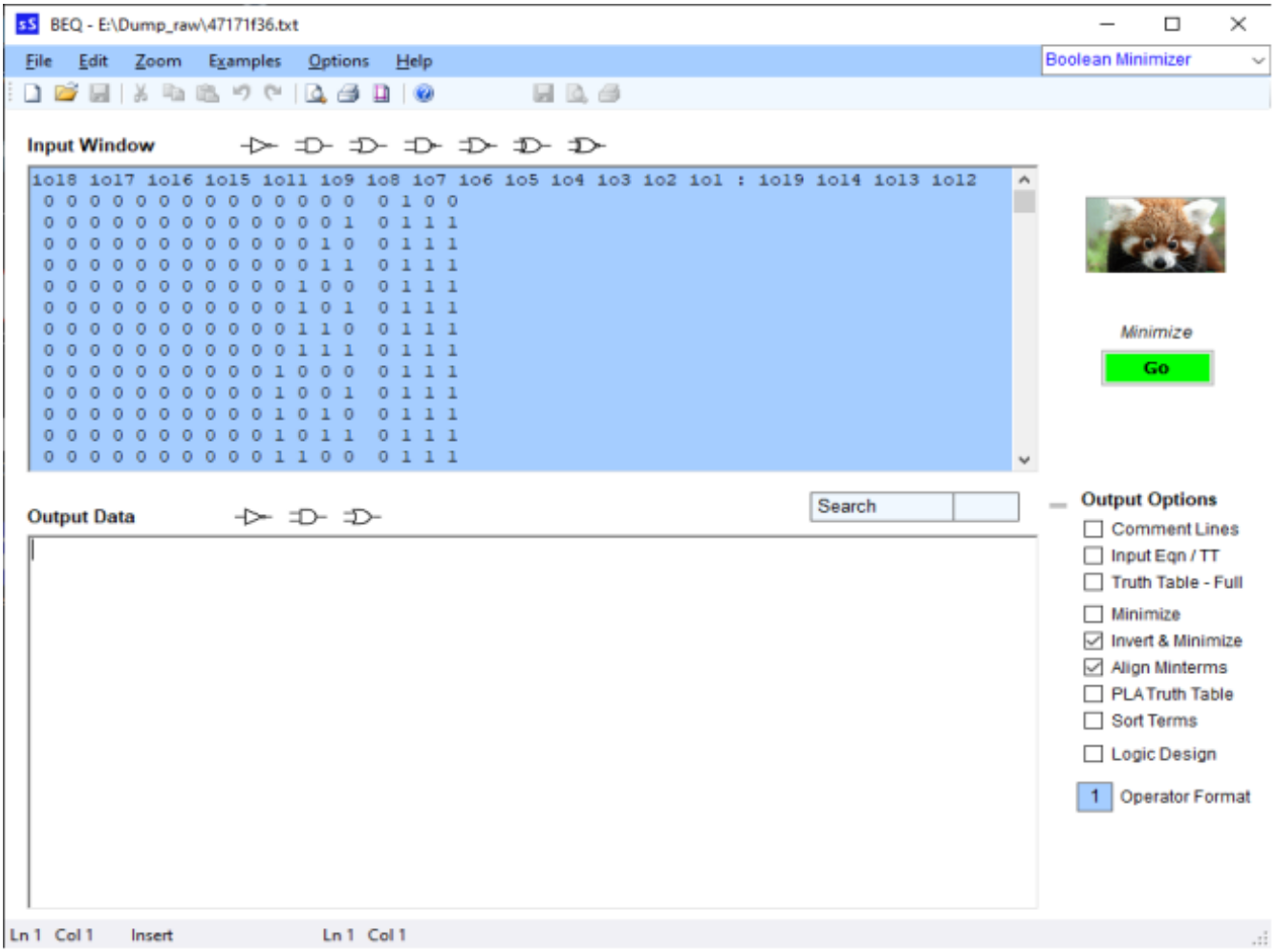
We need the software PALASM. Since it is quite old, it works under MS-DOS, so we use **DOSBOX** to execute it. The installation of DOSBOX is very simple. Use the enclosed configuration file, which can be found at

```
C:\Users\<username>\AppData\Local\DOSBox\dosbox-0.74.conf
```

and add following lines:

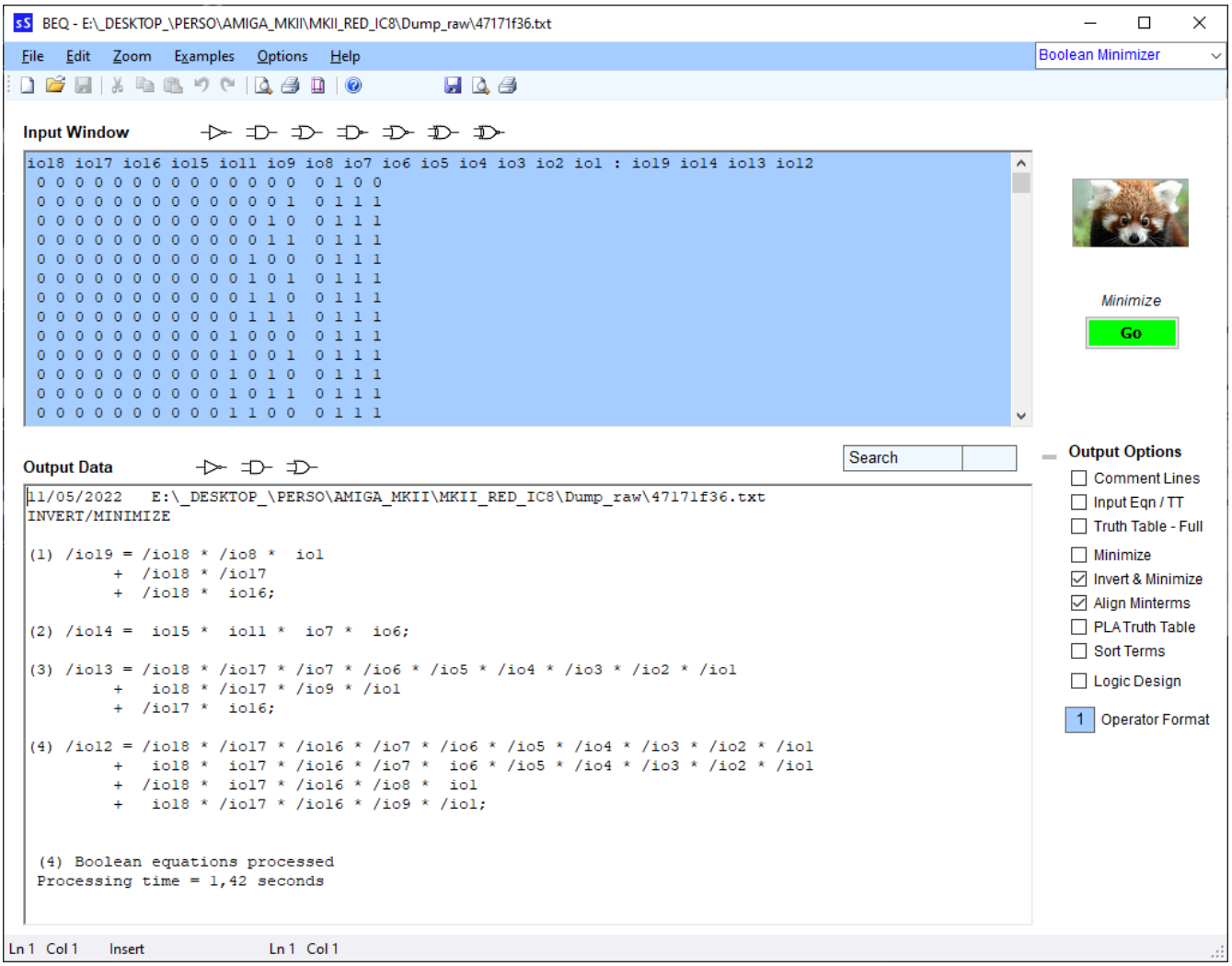
```
[autoexec]
# Lines in this section will be run at startup.
# You can put your MOUNT lines here.
mount C D:\dosprogs

PATH C:\DOSBOX\PALASM\EXE;C:\;"%PATH;C:\DOSBOX\PALASM4\EXE"
SET PALASM=C:\PALASM\

C:
CD C:\PALASM
```

Now load the truth table into Simple Solver (the modified text file).



The options have to be changed to this:

Press GO to analyze the data and reduce the equation



Step 5: Create the PDS file for PALASM

In the PALASM directory (e.g. `D:\dosprogs\PALASM`), create a PDS file and open it with your text editor. Copy following lines into the text file. Please ensure that the INPUT and OUTPUT pin declaration is identical with the original PAL (see above).

47171f36.PDS - Bloc-notes

Fichier Edition Format Affichage Aide

```
;PALASM Design Description

;---------------------------------- Declaration Segment ------------
TITLE    MYPAL1
PATTERN  None
REVISION 0
AUTHOR   Giants
COMPANY
DATE     10/04/2022

CHIP MYPAL1  PALCE16V8                              ; PALCE16V8

;---------------------------------- PIN Declarations ---------------
pin 1 io1       COMBINATORIAL              ; INPUT
pin 2 io2       COMBINATORIAL              ; INPUT
pin 3 io3       COMBINATORIAL              ; INPUT
pin 4 io4       COMBINATORIAL              ; INPUT
pin 5 io5       COMBINATORIAL              ; INPUT
pin 6 io6       COMBINATORIAL              ; INPUT
pin 7 io7       COMBINATORIAL              ; INPUT
pin 8 io8       COMBINATORIAL              ; INPUT
pin 9 io9       COMBINATORIAL              ; INPUT
pin 11 io11     COMBINATORIAL              ; INPUT
pin 12 io12     COMBINATORIAL              ; OUTPUT
pin 13 io13     COMBINATORIAL              ; OUTPUT
pin 14 io14     COMBINATORIAL              ; OUTPUT
pin 15 io15     COMBINATORIAL              ; INPUT
pin 16 io16     COMBINATORIAL              ; INPUT
pin 17 io17     COMBINATORIAL              ; INPUT
pin 18 io18     COMBINATORIAL              ; INPUT
pin 19 io19     COMBINATORIAL              ; OUTPUT
;---------------------------------- Boolean Equation Segment ------

EQUATIONS
```

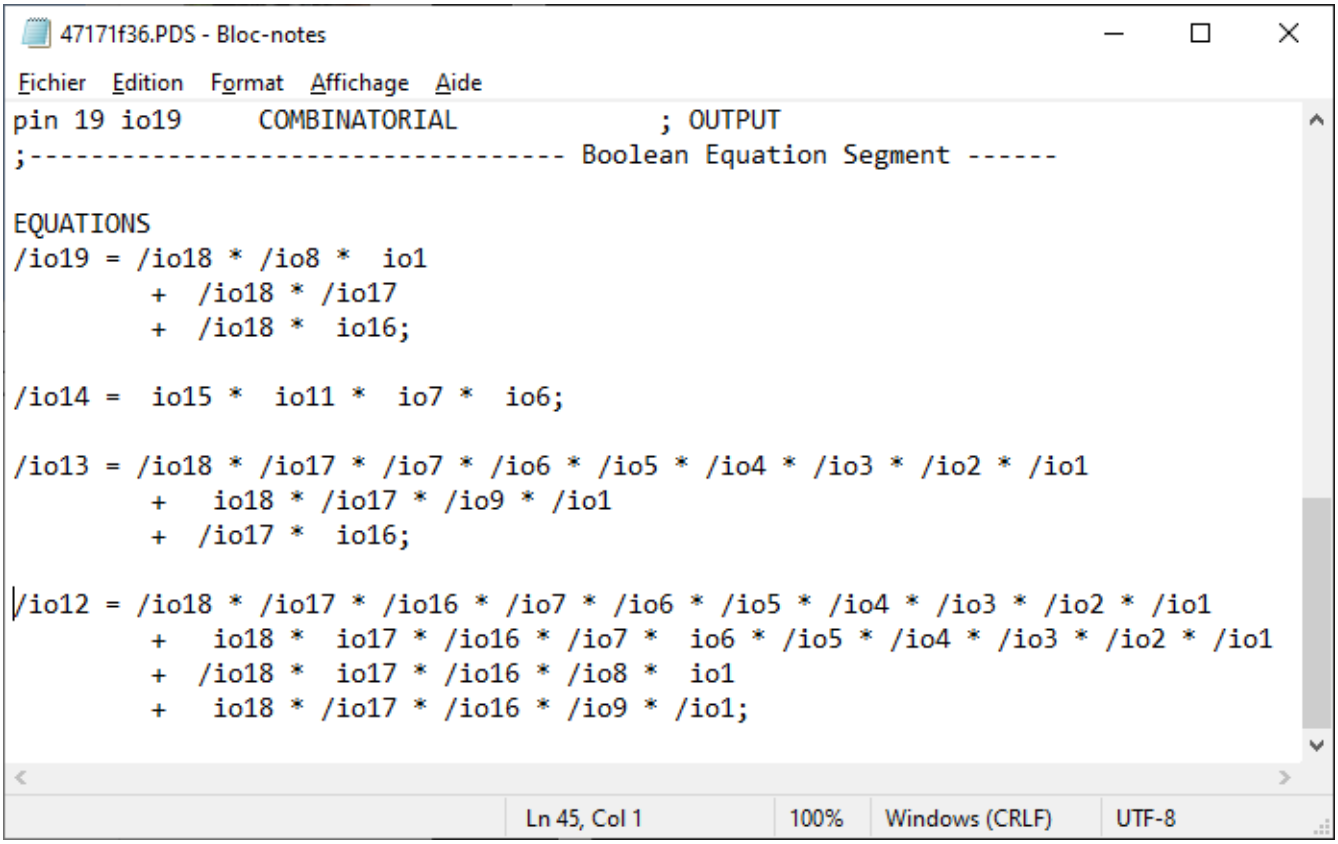
Ln 4, Col 16 100% Windows (CRLF) UTF-8

Some important information:

```
CHIP PYPAL1 PALCE16V8
```

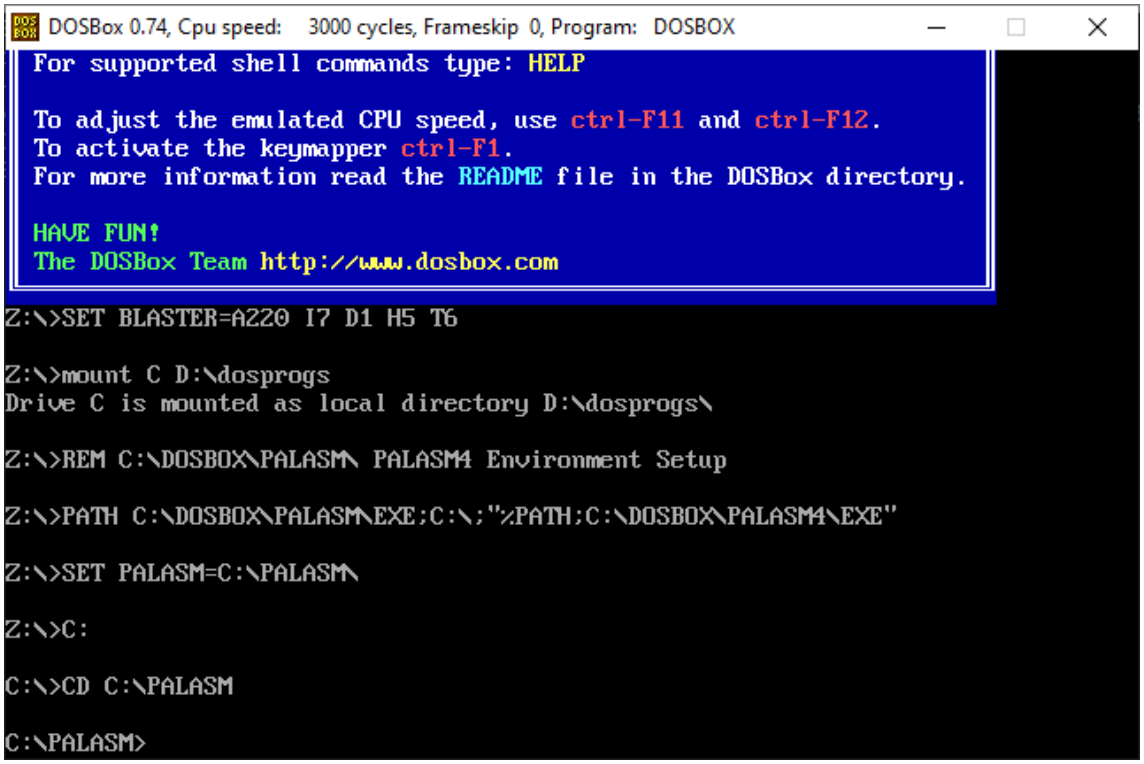
PALCE16V8 is used for the GAL16V8 (see PALASM documentation). If you use another “pattern”, make sure it is changed according your equation. It is necessary to configure each pin (except GND and Vcc, Pin 10 and Pin 20 for GAL16V8). As we changed the name of the pins in the text file to “io<number>” we have to use the same name.

In the EQUATIONS section we have to copy our minimized equations (the equations only), previously created with Simple Solver. Save the file.



Step 6: Compile and create the JEDEC file

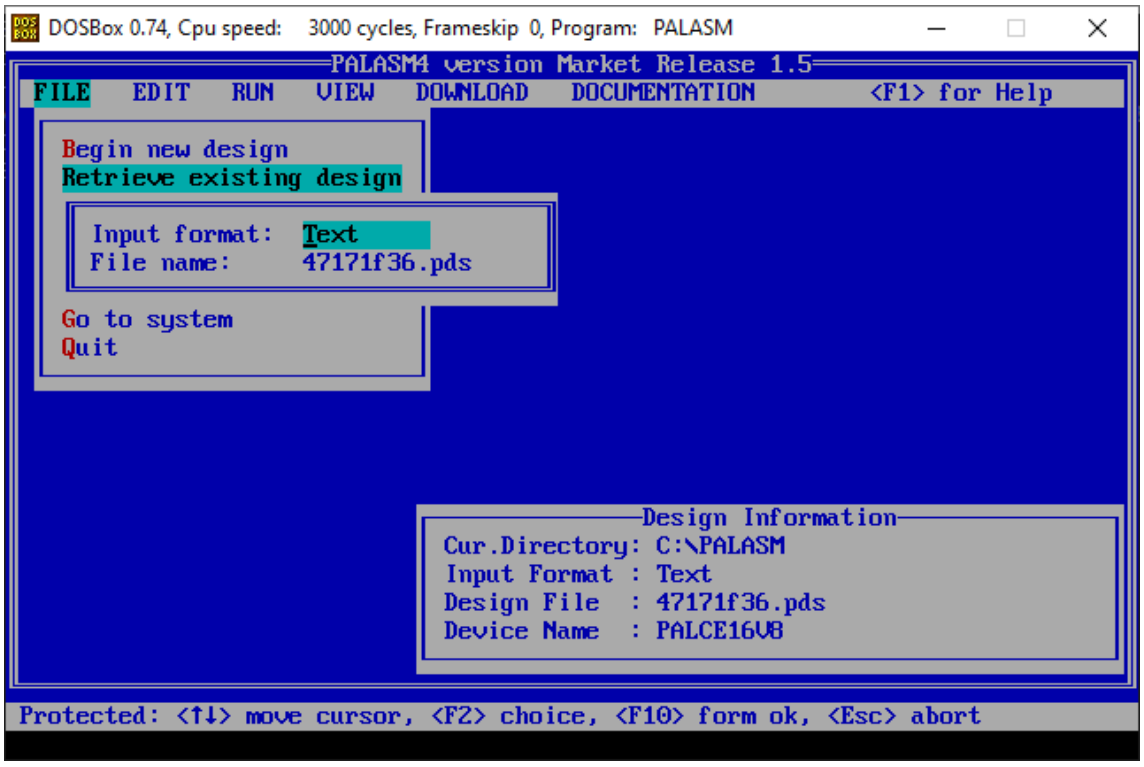
Start DOSBOX and execute PALASM.



Load the PDS file.

`FILE → Retrieve existing design → Input Format Text / Filename <your file>.pds`

Press F10 to accept the changes.



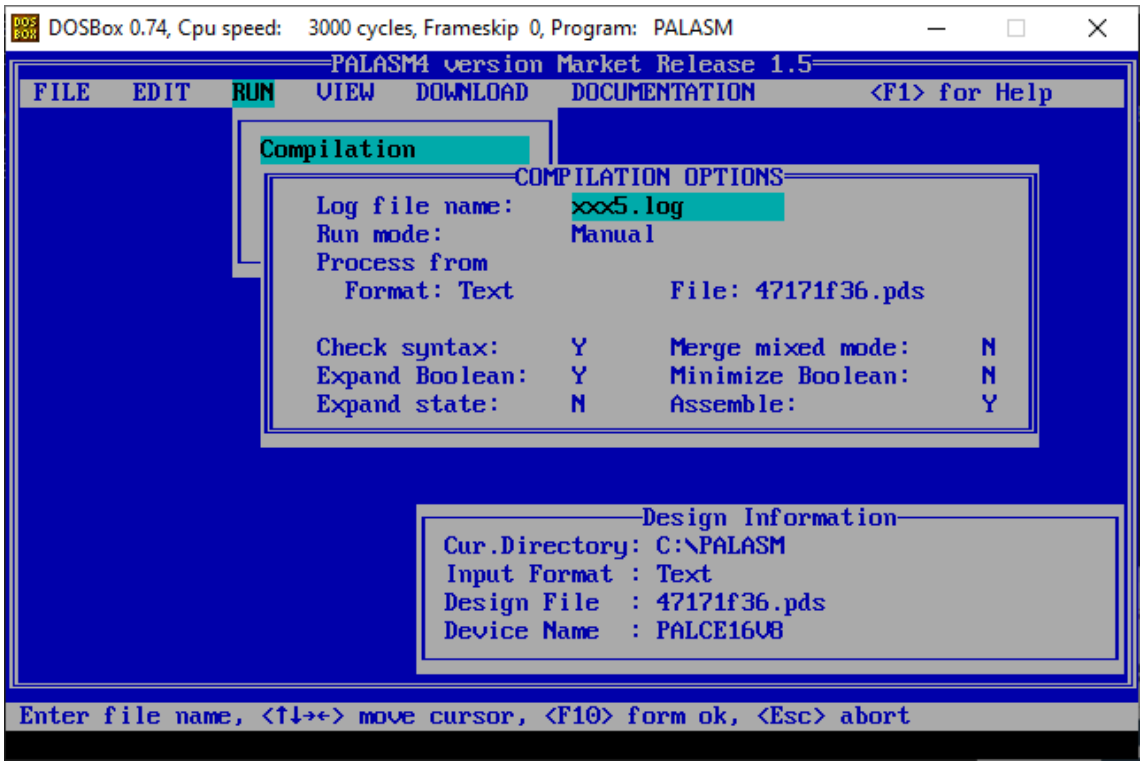
The file can be displayed using:

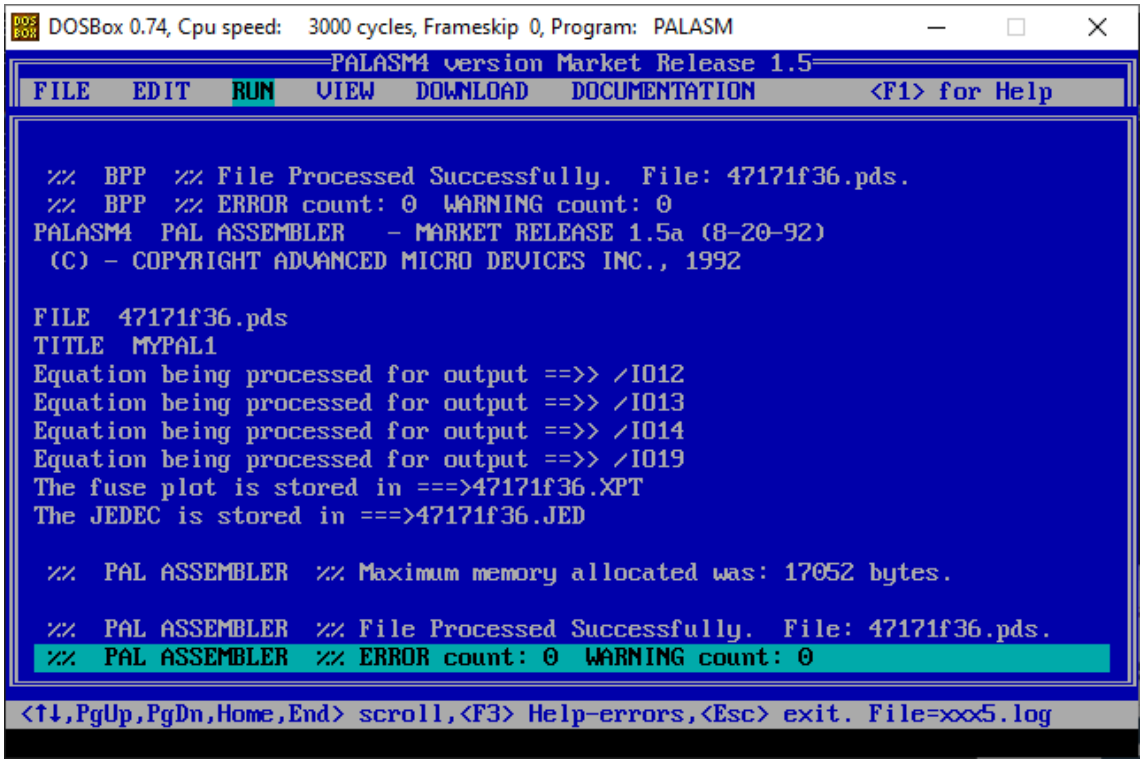
`EDIT → TEXT file → [Press ESC Key]`

You can navigate through the file to see if it is correctly loaded.

Press [F3] to quit the editor and compile the file:

`RUN → Compilation → (check the options and press F10)`

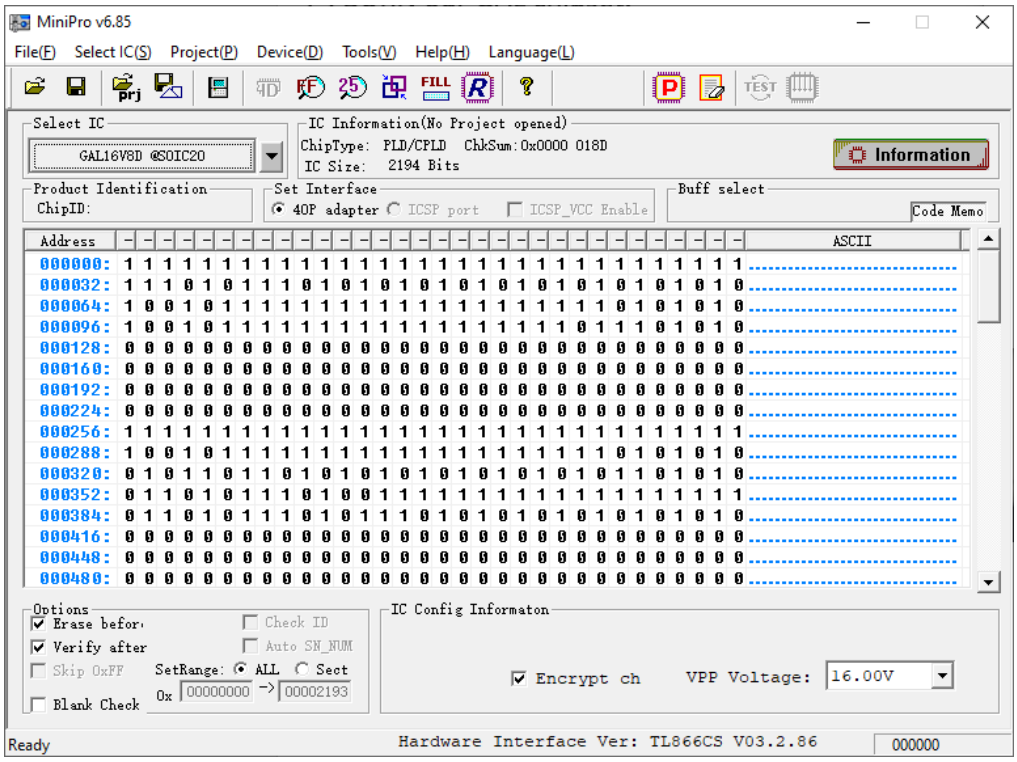




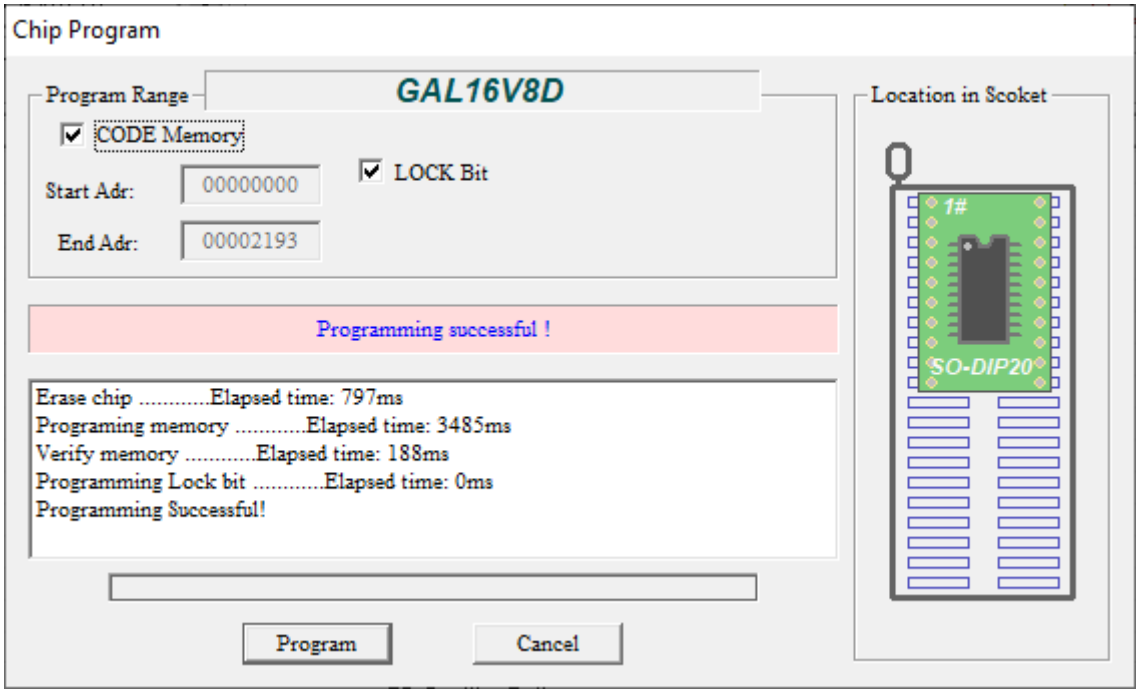
When no error appears a JED file has been created. Quit PALASM and DOSBOX.

Step 7: Burn the GAL

Insert the GAL16V8 in a DIP-20 Adapter and program it e.g. using the MiniPro programmer. Select GAL16V8D, load the JEDEC file and program it.



After a few seconds the GAL should be programmed.



Gratulations. That's all.

16 Appendix: Tweaks

This chapter contains some collected ideas.

16.1 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.

16.1.1 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.

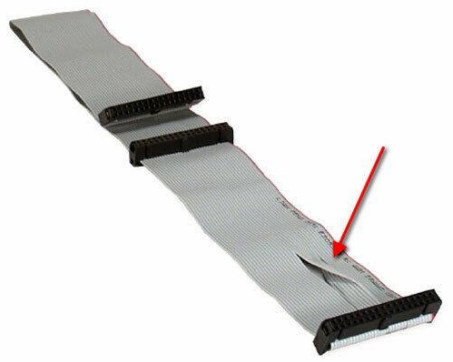


Figure 16.1: Floppy Cable

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

16.1.2 Extension via Jumper Cables

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

Figure 16.2: Jumper Cable (male-female)

16.1.3 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.

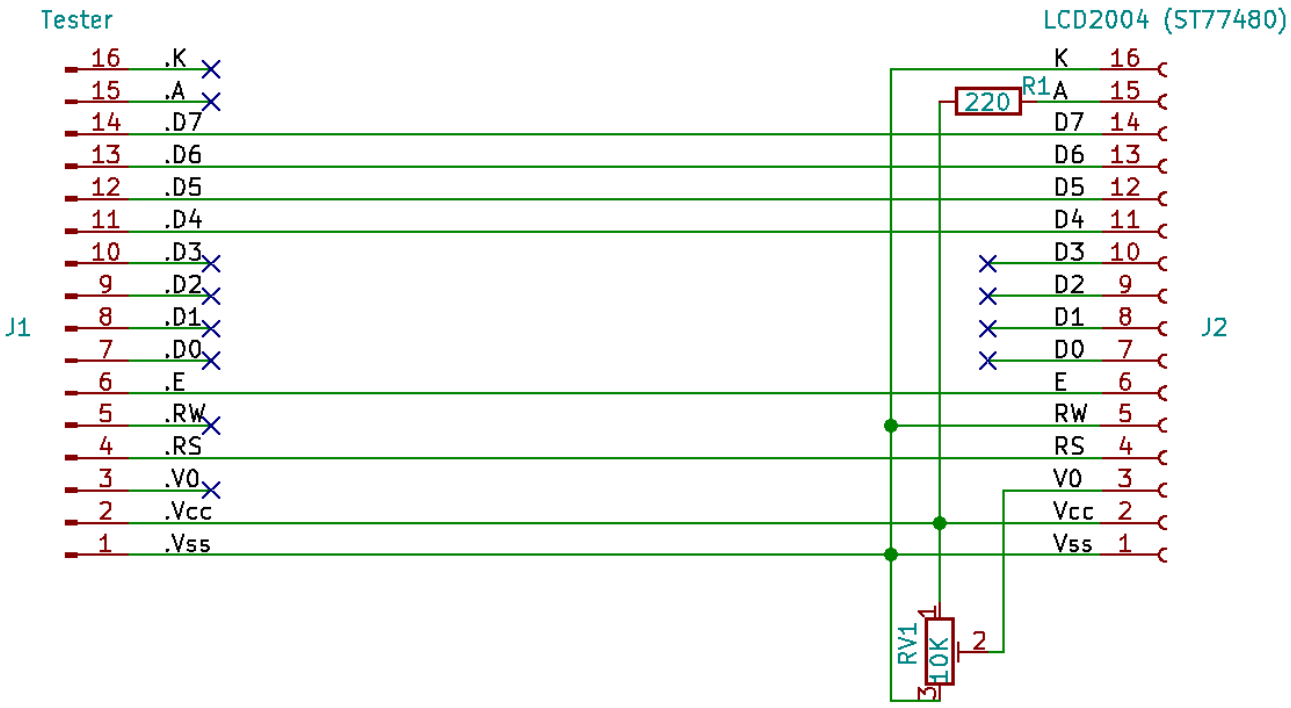


Figure 16.3: Wiring of the display

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.

16.2 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.

16.3 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.

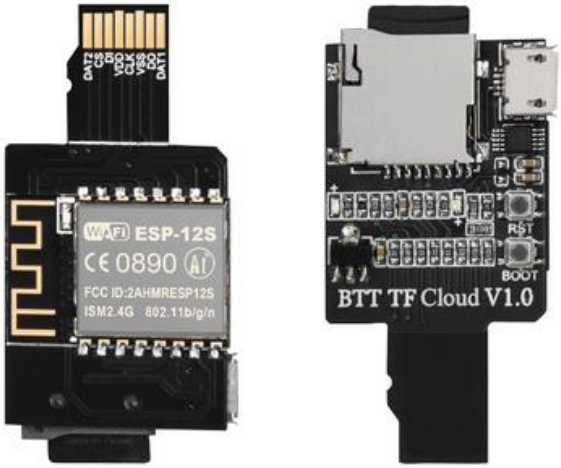


Figure 16.4: TF Cloud V1.0 adapter

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.

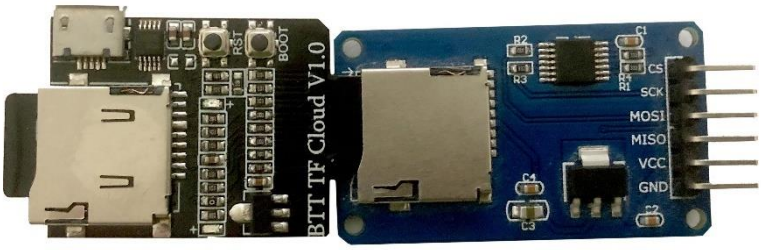


Figure 16.5: TF Cloud V1.0 Adapter (left), Micro-SD card adapter (right)

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".

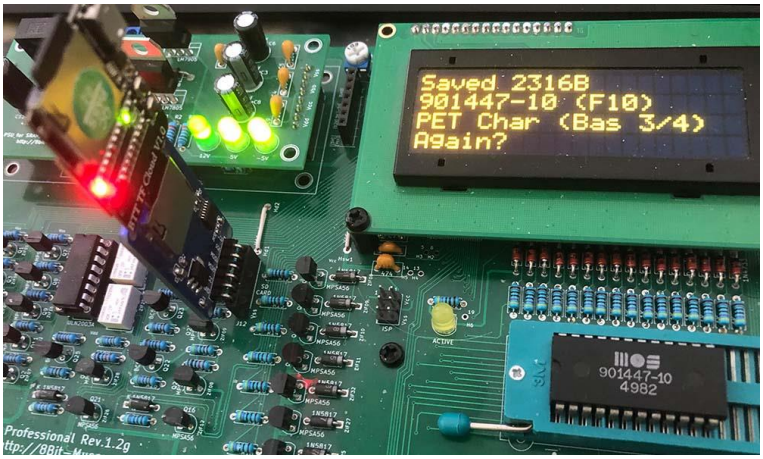


Figure 16.6: TF Cloud V1.0 adapter in use

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:

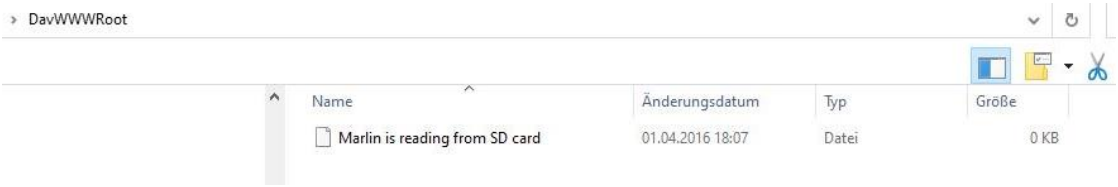


Figure 16.7: TF Cloud V1.0 adapter in use

otherwise the content of the SD card is displayed.

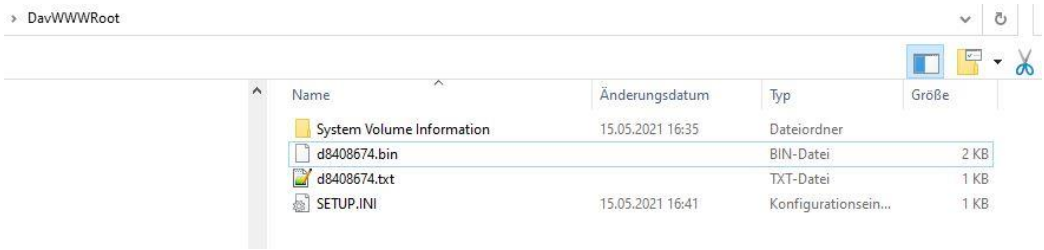


Figure 16.8: Contents of the SD card

The data can now be easily copied.

16.4 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):

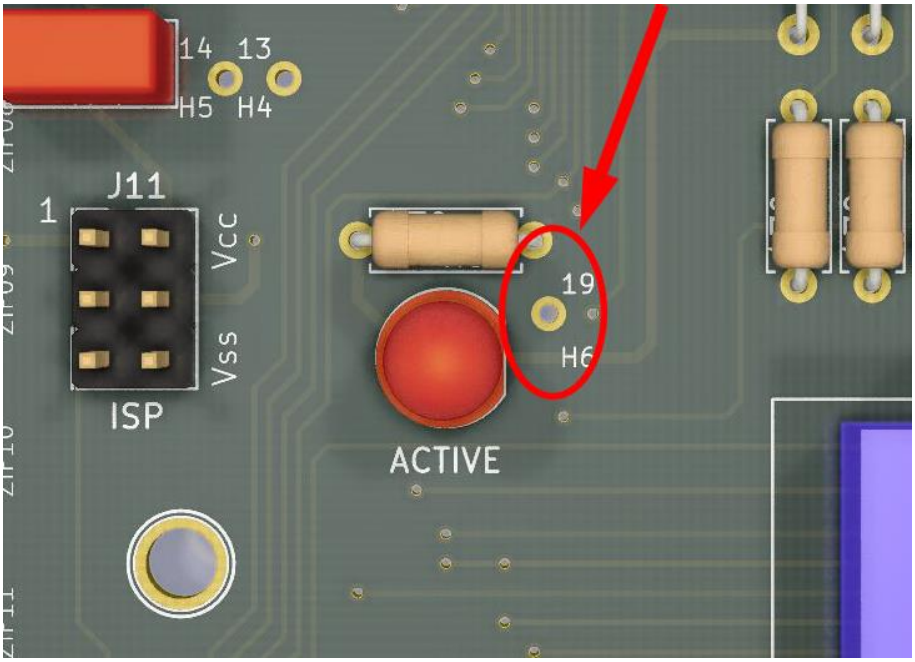


Figure 16.9: Connecting the push button (19/H6)

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[10]
		X		Jump to main menu[11] or enter submenu
			X	Test selected chip
			keep pressed	Alternative mode in some cases

[10] in submenu only
[11] in submenu only