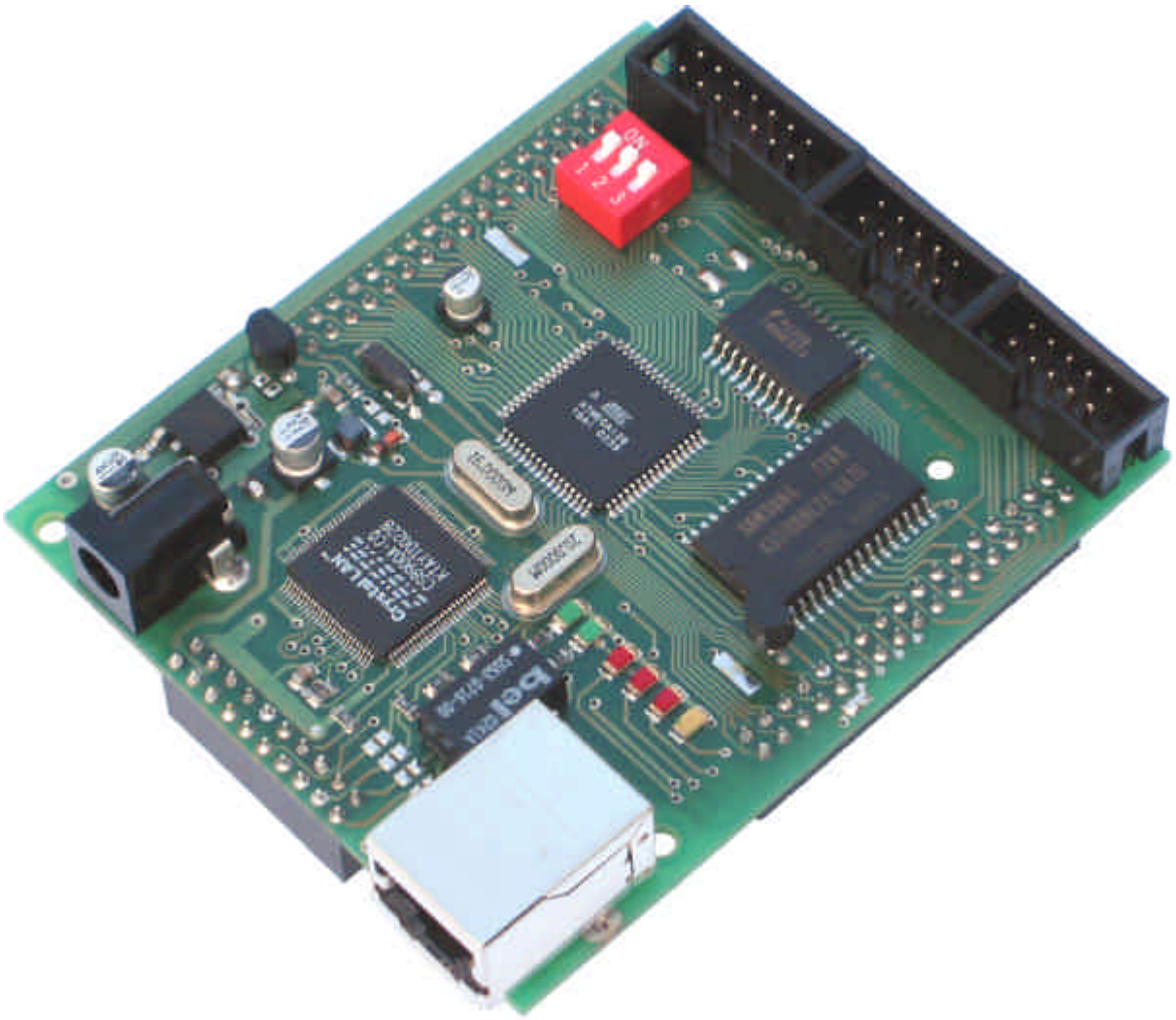


easyToWeb-AVR

Module with ATmega128® and CS8900A

Hardware Guide (Hardware-Version 1.3)

Forschungs- und Transferzentrum Leipzig
Wächterstr. 13
D-04107 Leipzig
Germany
phone +49 341 3076 1136
fax +49 341 3076 1220
info@easyToWeb.net
www.easyToWeb.net

In this manual are descriptions for copyrighted products that are not explicitly indicated as such. The absense of the trademark (™) and copyright (©) symbols does not imply that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual.

The information in this document has been carefully checked and is believed to be entirely reliable. However, FTZ Leipzig assumes no responsibility for any inaccuracies. FTZ Leipzig neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. FTZ Leipzig reserves the right to alter the information contained herein without prior notification and accepts no resposibility for any damages which might result.

Additionally, FTZ Leipzig offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. FTZ Leipzig further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

Forschungs- und Transferzentrum Leipzig
Udo Jakobza / Thomas Minner
Wächterstr. 13
D-04107 Leipzig
Germany
phone +49(0)341-3076-1136
fax +49(0)341-3076-1220
e-mail: info@easyToWeb.net
http://www.easyToWeb.net

table of contents

History

- Version 1.3a
  - 28.11.2002: adaption of hardware guide to hardware-version 1.3
  - 20.12.2002: new picture
  - 29.01.2003: complete the data of Prog-Adapter (version1.4)
  - 04.02.2003: englisch version

Features of easyToWeb-AVR module

- module for prototypes
- based on the micocontroller ATmega128
- **Memory:**
  - 128kByte Flash Code-Speicher (within ATmega128®)
  - 4 kByte EEPROM (within ATmega128®; Endurance: 100,000 Write/Erase Cycles
  - 4 kByte SRAM (within ATmega128®)
  - 28 kByte or 60 kByte external SRAM
  - 8 Byte ROM Unique 64-bit registration number
  - 8 Byte PROM for additional informations (MAC-Address)
  - 32 Byte EEPROM for critical data
  - 2Mbit ... 32Mbit DataFlash® (serial Flash)
- **External Interface:**
  - 10Base-T (10 Mbit/s, Twisted-Pair-Kabel, RJ45-Steckverbinder, with isolation transformer)
  - eTW-Interface for internal signals (2 x 34 Pins und 1 x 20 Pins)
  - all PORT-signals of ATmega128
  - whole Memory-Interface (A0 ... A15; D0 ... D7; AD0 ... AD7)
  - power connector
  - 2 x serial UART's
    2 x UART with TTL-Level (max. 2MBaud) or
    2 x UART with RS232-Level (max. 115kBaud)
- **Power:**
  - 3.3V and 5V Version available
  - only one power supply-voltage
  - internal poweradapter with 8 .. 12V input voltage - with polarity protection
  - power interface via eTW-Interface
  - separated voltage interface for microcontroller and the other parts of the modul (eg. accu-buffering for ATmega128)
  - power consumption:
    - 145mA with 5V, 16MHz ATmega128-Clock, ATmega128-Activ-Mode
    - 110mA with 5V, 16MHz ATmega128-Clock, ATmega128-Idle-Mode
    - 45mA with 5V, 16MHz ATmega128-Clock, ATmega128-Idle-Mode, CS8900-Sleep
    - 85mA with 3.3V, 7.3728MHz ATmega128L-Clock, ATmega128L-Activ-Mode
- **resources:**
  - the Ethernet-Controller (CS8900A) is addressed complete via external memory access (address: 0xF000 .. 0xFFFF)
  - PortB.0 - Chip-Select-Line for internal DataFlash®
  - PortD.7 - serial number + PROM + EEPROM
- **Software:**
  - free available C-Sorces (download: see www.easyToWeb.net)
  - TCP/IP-Server adapted from extra booklet [1]
  - completed with extensive software-library (low-level- and test-routines)
  - Debug-Interface with simple handling via terminal program and serial interface (UART0)
- physical dimensions: 85,7mm x 73,7mm

Introduction

The concept „easyToWeb“

This concept is a kicking combination of microcontroller and software. For the simple communication bound, in this case ethernet, an appropriate controller is additionally required. It is furthermore impotant to reduce the necessary procoll extend to a minimal size and to adapt the abilities of the microcontroller i.e. Rechenpower and ressources. The result has been a TCP/IP protocoll stack that fullfill the functions extend of a simple webserver, and, that provides a solid base for further experiments and expansions in the aim of specifical applications.

The microcontroller

The most efficient member of the AVR-family, the Atmega128(L), has been chosen for this project. It is a further development of the earlier ATmega103 with 128Kbyte code memory and 4KByte main storage. The basic parameters have not been changed, but relevant improvements have been carried out by the firm ATMEL. Some important aspects that are treated by the project „easyToWb-AVR“ should not be left unstated:

- A confident internal reset-logic with adjustable reset-levels,
- a improved bus interface with extended possibilities to tune the timing parameters (slight changed timing),
- improved Port F - I/O-Port with enlarged ADC and
- JTAG-interface with enlarged functionality (On-Chip Debug-System).

The minimal duration of a command cycle has been reduced to 62,5 ns. In order to enable a scaleable solution, 3 additional pin-compatible Atmega types with variable capacities (ATmega64, ATmega32 und ATmega103) have been made available.

The Ethernet-Controller

The most important aspects are the complete integration of ethernet-interface (MAC and PHY), transmit- and receive-Buffer and a simple with a simple bus interface. There is only a few types available (Davicom - DM9000; SMSC - LAN91C96 oder LAN91C111; Cirrus Logic - CS8900A).

The Hardware

The following paragraph presents the basic constitution of the webserver modul. This element is principally constituted of the ethernet- and microcontroller so as of suitable external output. The diagram gives you an overview on it’s structure (figure 1).

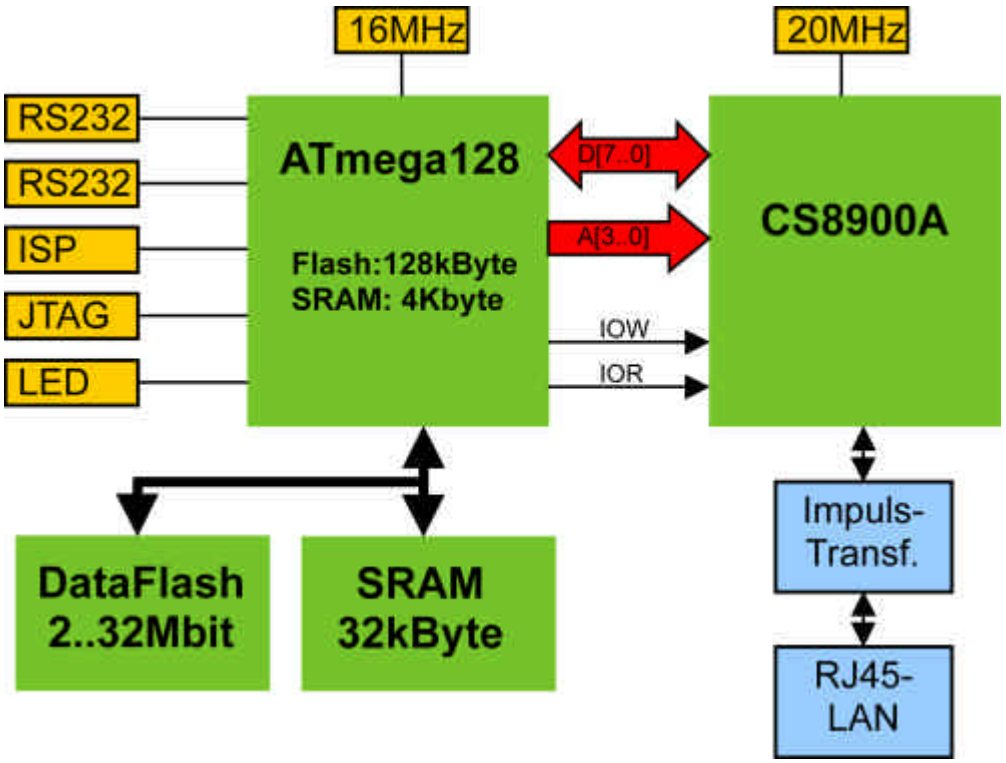


figure 1: *block diagram of "easyToWeb-AVR" module*

ATmega128(L) and CS8900A

The ATmega128(L) has an external bus interface. It builds up the register of the CS8900A through a simple logic in the area 0xF000 .. 0xFFFF of the ATmega128(L). The ethernet-controller operates in »I/O«-Modus, which requires only 8 data- and 6 control-transmission lines The external analog Beschaltung of the CS8900A relies principally on producer's recommendations. The controller sustains different physical ethernet-media (10Base-T, 10Base-2, AUI), but this realisation is limited to the nowadays most used »10Base-T«. The connection to the hub occurs here over a standard patch cable with RJ45-plugs (also known as Western-plug).
The impulstransformator »S553-0716-00« of the firm Bel Fuse Inc. Is used for an obligatory galvanic seperation of the network and of the webserer-element. It could be replaced by similar types. In this case, the voltage-transfer- ratio should be noticed.
The ethernet-controller signals over LEDs the connection to ethernet (LINK-LED) and the read and write activities (LAN-LED).
The other wiring of the microcontroller ATmega128(L) encloses a 32 kByte SRAM, a debug-Interface, two RS232-serial interfaces and a crystal. A serial DataFlash® has been conceived for permanent storage of datas. A package-version that enables a flexible equipment with 2 to 32 Mbit variants has been chosen.

Programing and debugging

For the in-circuit-programming and debugging, the circuit possesses a JTAG-interface or ISP-interface. The required signals are available over a 14-pin- connector that is conceived for the connection of a debug-adapter. It is therefore possible to use the ATmega128(L) over the ISP-programm-adapter or JTAG-adapter. 4 additional LED at port D are available for support of entering in functions and of tests of personal applications,.

Electricity supply

Two internal supply networks have been realised on the „easyToWeb-AVR" server, so as to enable a continual supply of the ATmega128(L) and of the remaining components, even when seperated. This configuration has the advantage that the ATmega128(L) can function with a

32 kHz-clockquarz as RTC or the ATmega128(L) can be set in a sleep-mode (only with a few µA as electrical supply). The electrical supply takes places either over the external interface with 5V or 3.3V or over a jack plug plug with about 8V....12V (~145 mA@5V and ~85mA@3.3V).

Interface

All portpins of the ATmega128(L) and supply potentials are externally available for other uses over a connector. When necessary, the board »piggy-back« can be fixed on the mother circuit board or can directly be integrated to an application. The already assembled board is shown in figure 2.
In the aim to enable an universal use of the board, two additional RS232-interfaces have been incorporated. The RXD- and TXD-signals are each accessible over a 10-pin- connector.

The tools

A very important part of the "easyToWeb-AVR"-solution is the software that has been used. From the variable available instruments of the AVR family, the CodeVisionAVR C-Compiler has been chosen. Besides a stable, easy to manage and complete integrated development environment (figure 2), the good relation price-capacities have been considered for the choice of this compiler. A further comfortable property is the transparent integration of the programm-interface into CodeVisionAVR. Other informations on the CodeVisionAVR C-Compiler are to be found under "http://www.hpinfotech.ro/".

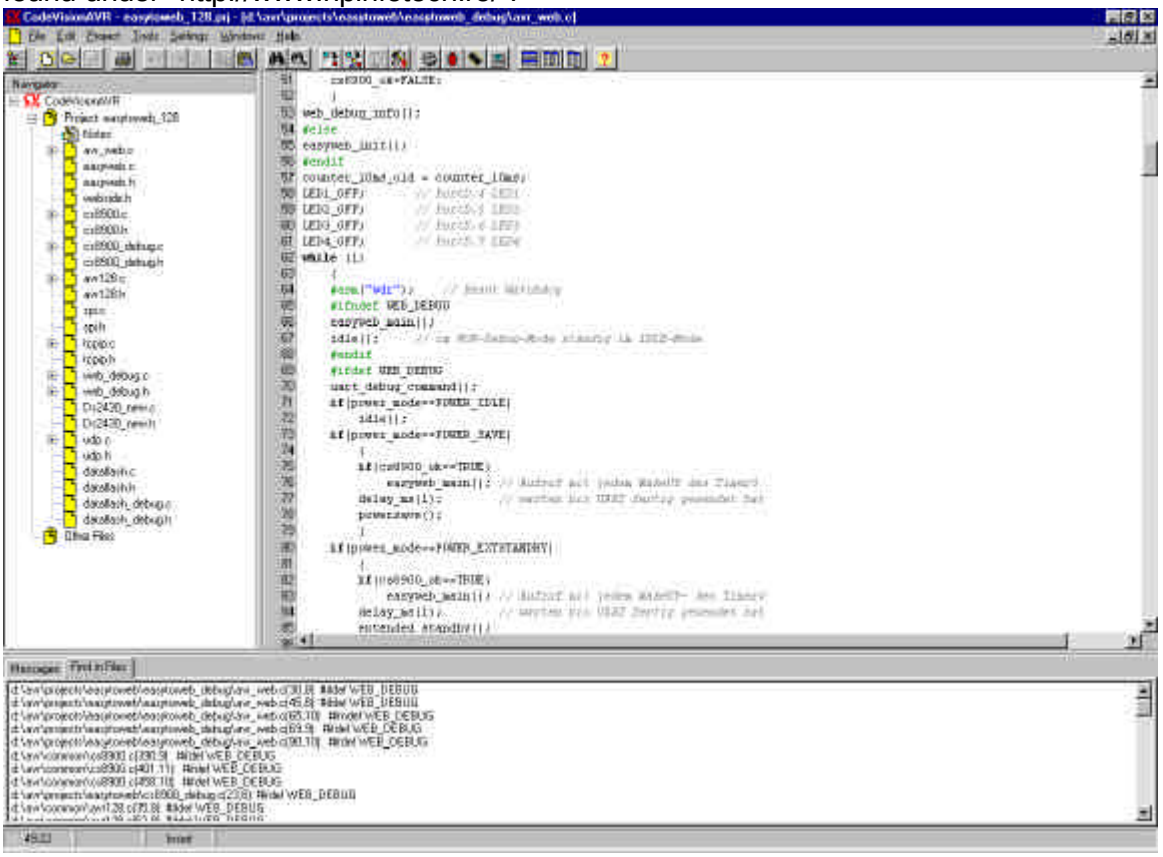


figure 2 *enviroment for developping CodeVisionAVR*

The „easyToWeb-AVR“-solution

With the "easyToWeb-AVR"-module whitch is presented in figure 3, a hardware-base that can be used and developed as wished by the user is available. Contrary to other solutions that are available on the market, the entry in the embedded internet's world (starterkit version) so as a gradual construction of a complicated solution can take place with one and the same „easyToWeb-AVR“- module. This module can also directly be used to construct tools because of it's smaller sizes 86 x 74 mm and flexible possibilties of equipment.

The principal components of the software-interface have been made compatible to known solutions from [1]. For this means, the software has been inserted into the ATmega128(L) and into the CodeVisionAVR C-compiler. With the idea „easytoWeb“ in mind, we took care to preserve to a large extend a homogeneous sotware-structure and not to modify the functions of the existing API. The documentation from [1] can therefore also be used for the "easyToWeb-AVR". The complete source-text is included in the starterkit and offers to the interested beginner an overview into the working way of the protokoll stack. For the advanced programmer, it is the base for the integration of additional protocolls and functions and to develop a own solution.
With in mind – Take it easy!

figure 3 *http-side of "easyToWeb-AVR" module*

eTW-Interface (easyToWeb-Interface) and User-Interface

The "easyToWeb-AVR" board offers variable possibilities for the usage of the interfaces of the ATmega128. Thus, the internal signals of the board have been drained over 3 Buchsenleisten (X1_1; X1_2; X1_3), so that the whole functionality can be used on board specifical for custommers. Moreover, the additional interfaces are directly available to the user over a standard Steckverbinder. The UART1- and UART2-interfaces (JP5, JP6), the debug-port (JP1); the power-link (P1) and the ethernet-link (JP9) belong to it.

Location of the external interfaces

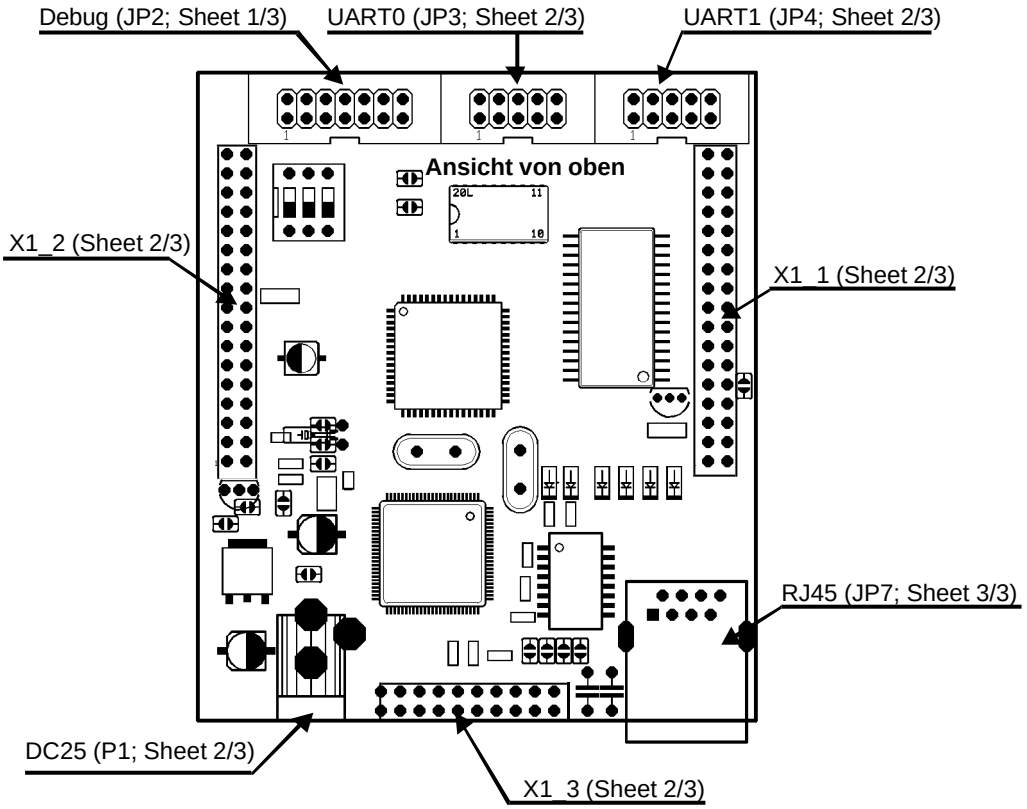


figure 4: *external connections - top-side*

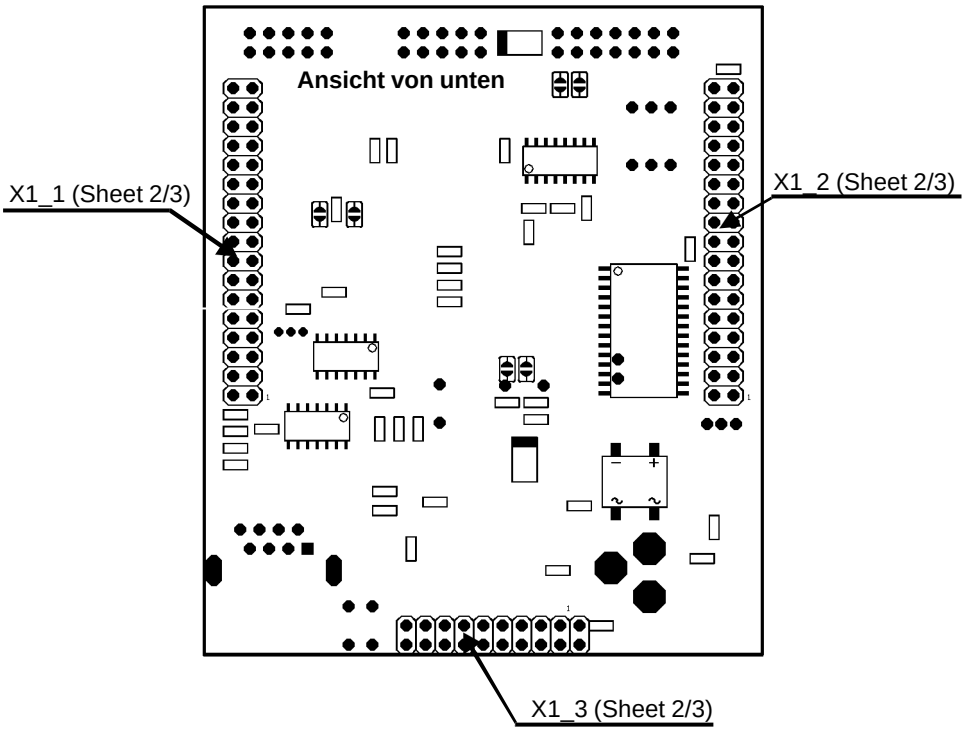


figure 5: *external connections - bottom-side*

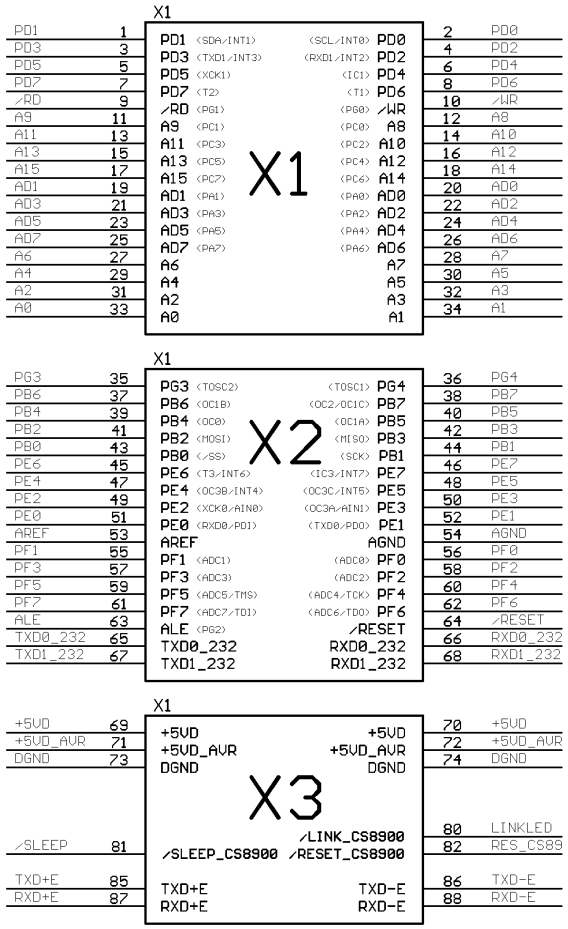


figure 6: *connections of eTW-Interface (X1, X2, X3)*

delivery of eTW interface

- X1, X2 mounted with 2 x 17 pin female connector and 2,54mm grid;
- X3 mounted with 2 x 10 pin female connector im 2,54mm Raster;
- JP5, JP6 mounted with 10 pin male connector;
- JP7 mounted with 8 pin RJ45-Connector;
- JP1 mounted with 14 pin male connector
- X4 mounted with 2,1mm jack (socket)

eTW-Interface and User-Interface

The follow tables is given a general view to the external terminations ot the "easyToWeb-AVR" module. There are two main groups:

- *eTW-Interface* (easyToWeb-Interface),
- *User-Interface*.

The *eTW-Interface* contains the pin female connectors X1_1 (2x17 pin), X1_2 (2x17 pin) and X1_3 (2x10 pin). The *User-Interface* contains JP2 (debug interface), JP3 (UART0), JP4 (UART1), P1 (power connection) and JP7 (RJ45 ethernet female connector).

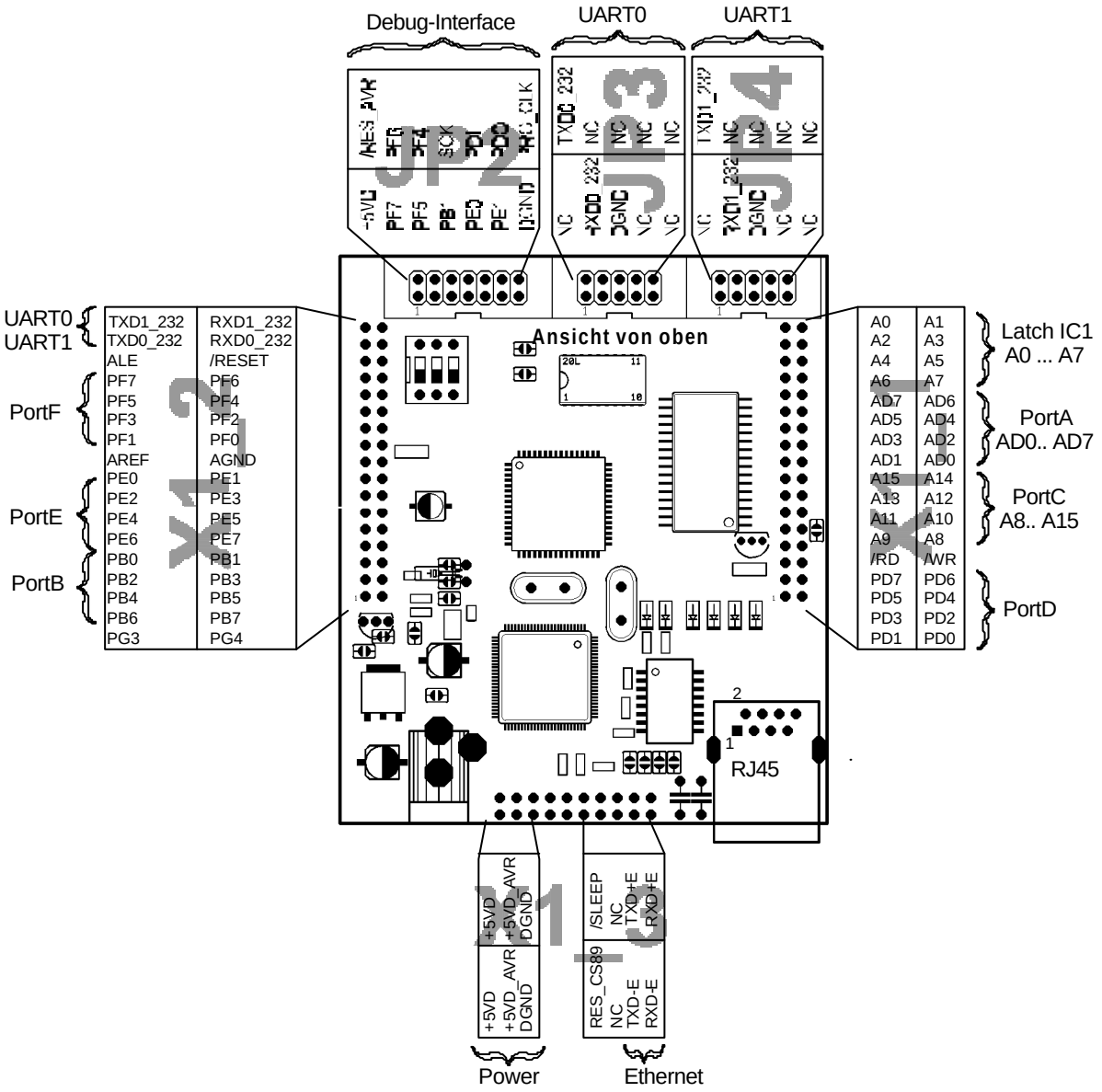


figure 7: *eTW-Interface and User-Interfaces*

eTW-Interface

eTW-Interface	signal	ATmega128-Pin	description
X1_1.1	PD1	26	PortD.1 (SDA/INT1)
X1_1.2	PD0	25	PortD.0 (SCL/INT0)
X1_1.3	PD3	28	PortD.3 (TXD1/INT3)
X1_1.4	PD2	27	PortD.2 (TXD1/INT2)
X1_1.5	PD5	30	PortD.5 (XCK1)[1]
X1_1.6	PD4	29	PortD.4 (IC1)[1]
X1_1.7	PD7	32	PortD.7 (T2)[1]
X1_1.8	PD6	31	PortD.6 (T1)[1]
X1_1.9	/RD	34	/RD (PortG.1)[1]
X1_1.10	/WR	33	/WR (PortG.0)[1]
X1_1.11	A9	36	A9 (PortC.1)[1]
X1_1.12	A8	35	A8 (PortC.0)[1]
X1_1.13	A11	38	A11 (PortC.3)[1]
X1_1.14	A10	37	A10 (PortC.2)[1]
X1_1.15	A13	40	A13 (PortC.5)[1]
X1_1.16	A12	39	A12 (PortC.4)[1]
X1_1.17	A15	42	A15 (PortC.7)[1]
X1_1.18	A14	41	A14 (PortC.6)[1]
X1_1.19	AD1	50	AD1 (PortA.1)[1]
X1_1.20	AD0	51	AD0 (PortA.0)[1]
X1_1.21	AD3	48	AD3 (PortA.3)[1]
X1_1.22	AD2	49	AD2 (PortA.2)[1]
X1_1.23	AD5	46	AD5 (PortA.5)[1]
X1_1.24	AD4	47	AD4 (PortA.4)[1]
X1_1.25	AD7	44	AD7 (PortA.7)[1]
X1_1.26	AD6	45	AD6 (PortA.6)[1]
X1_1.27	A6		A6 of Latch IC1.13 (74VHC573)[1]
X1_1.28	A7		A7 of Latch IC1.12 (74VHC573)[1]
X1_1.29	A4		A4 of Latch IC1.15 (74VHC573)[1]
X1_1.30	A5		A5 of Latch IC1.14 (74VHC573)[1]
X1_1.31	A2		A2 of Latch IC1.17 (74VHC573)[1]
X1_1.32	A3		A3 of Latch IC1.16 (74VHC573)[1]
X1_1.33	A0		A0 of Latch IC1.19 (74VHC573)[1]
X1_1.34	A1		A1 of Latch IC1.18 (74VHC573)[1]
X1_2.35	PG3	18	TOSC2 (PortG.3)[3]
X1_2.36	PG4	19	TOSC1 (PortG.4)[3]
X1_2.37	PB6	16	PortB.6 (OC1B)
X1_2.38	PB7	17	PortB.7 (OC2/OC1C)
X1_2.39	PB4	14	PortB.4 (OC0)
X1_2.40	PB5	15	PortB.5 (OC1A)
X1_2.41	PB2	12	MOSI (PortB.2)[1]
X1_2.42	PB3	13	MISO (PortB.3)[1]
X1_2.43	**PB0**	**10**	***/SS* (PortB.0)[2]**
X1_2.44	*PB1*	*11*	*SCK (PortB.1)[1, 5]*
X1_2.45	PE6	8	PortE.6 (T3/INT6)
X1_2.46	PE7	9	PortE.7 (IC3/INT7)
X1_2.47	PE4	6	PortE.4 (OC3B/INT4)
X1_2.48	PE5	7	PortE.5 (OC3C/INT5)
X1_2.49	PE2	4	PortE.2 (XCK0/AIN0)
X1_2.50	PE3	5	PortE.3 (OC3A/AIN1)
X1_2.51	*PE0*	*2*	*RXD0 (PDI/PortE.0)[4, 5]*
X1_2.52	*PE1*	*3*	*TXD0 (PDO/PortE.1)[4, 5]*
X1_2.53	AREF	62	reference-voltage for ADC of ATmega128
X1_2.54	AGND	63	analog-ground of ATmega128
X1_2.55	PF1	60	PortF.1 (ADC1)
X1_2.56	PF0	61	PortF.0 (ADC0)
X1_2.57	PF3	58	PortF.3 (ADC3)
X1_2.58	PF2	59	PortF.2 (ADC2)
X1_2.59	PF5	56	PortF.5 (ADC5/TMS)
X1_2.60	PF4	57	PortF.4 (ADC4/TCK)
X1_2.61	PF7	54	PortF.7 (ADC7/TDI)
X1_2.62	PF6	55	PortF.6 (ADC6/TDO)
X1_2.63	ALE	43	ALE (PortG.2)
X1_2.64	/RESET	20	/RESET[3]
X1_2.65	TXD0_232		transmit output IC8.7 (MAX3232)
X1_2.66	RXD0_232		receive-input IC8.8 (MAX3232)
X1_2.67	TXD1_232		transmit output IC8.14 (MAX3232)
X1_2.68	RXD1_232		receive input IC8.13 (MAX3232)
X1_3.69	+5VD		+5V (CS8900, SRAM and so on)

eTW-Interface	signal	ATmega128-Pin	description
X1_3.70	+5VD		+5V (CS8900, SRAM and so on)
X1_3.71	+5VD_AVR	52, 21	+5V ATmega128
X1_3.72	+5VD_AVR	52, 21	+5V ATmega128
X1_3.73	DGND	53, 22	DGND
X1_3.74	DGND	53, 22	DGND
X1_3.75 .. X1_3.80	NC		not connected
X1_3.81	/SLEEP		/SLEEP-Pin of CS8900A
X1_3.82	RES_CS89		RESET-Pin of CS8900A
X1_3.83	NC		not connected
X1_3.84	NC		not connected
X1_3.85	TXD+E		transmit output IC11.87 (CS8900A)[3]
X1_3.86	TXD-E		transmit output IC11.88 (CS8900A)[3]
X1_3.87	RXD+E		receive input IC11.91 (CS8900A)[3]
X1_3.88	RXD-E		receive input IC11.91 (CS8900A)[3]

Table 1: *Pins of eTW-Interface sorted with* ***Pin-Numbers***

eTW-Interface	signal	ATmega128-Pin	description
X1_1.33	A0		A0 of Latch IC1.19 (74VHC573)[1]
X1_1.34	A1		A1 of Latch IC1.18 (74VHC573)[1]
X1_1.31	A2		A2 of Latch IC1.17 (74VHC573)[1]
X1_1.32	A3		A3 of Latch IC1.16 (74VHC573)[1]
X1_1.29	A4		A4 of Latch IC1.15 (74VHC573)[1]
X1_1.30	A5		A5 of Latch IC1.14 (74VHC573)[1]
X1_1.27	A6		A6 of Latch IC1.13 (74VHC573)[1]
X1_1.28	A7		A7 of Latch IC1.12 (74VHC573)[1]
X1_1.12	A8	35	A8 (PortC.0)[1]
X1_1.11	A9	36	A9 (PortC.1)[1]
X1_1.14	A10	37	A10 (PortC.2)[1]
X1_1.13	A11	38	A11 (PortC.3)[1]
X1_1.16	A12	39	A12 (PortC.4)[1]
X1_1.15	A13	40	A13 (PortC.5)[1]
X1_1.18	A14	41	A14 (PortC.6)[1]
X1_1.17	A15	42	A15 (PortC.7)[1]
X1_1.20	AD0	51	AD0 (PortA.0)[1]
X1_1.19	AD1	50	AD1 (PortA.1)[1]
X1_1.22	AD2	49	AD2 (PortA.2)[1]
X1_1.21	AD3	48	AD3 (PortA.3)[1]
X1_1.24	AD4	47	AD4 (PortA.4)[1]
X1_1.23	AD5	46	AD5 (PortA.5)[1]
X1_1.26	AD6	45	AD6 (PortA.6)[1]
X1_1.25	AD7	44	AD7 (PortA.7)[1]
X1_2.54	AGND	63	analog-ground of ATmega128
X1_2.63	ALE	43	ALE (PortG.2)
X1_2.53	AREF	62	reference-voltage for ADC of ATmega128
X1_3.73	DGND	53, 22	DGND
X1_3.74	DGND	53, 22	DGND
X1_3.75 .. X1_3.80	NC		not connected
X1_3.83	NC		not connected
X1_3.84	NC		not connected
X1_2.43	**PB0**	**10**	**/SS (PortB.0)[2]**
X1_2.44	*PB1*	*11*	*SCK (PortB.1)[1, 5]*
X1_2.41	PB2	12	MOSI (PortB.2)[1]
X1_2.42	PB3	13	MISO (PortB.3)[1]
X1_2.39	PB4	14	PortB.4 (OC0)
X1_2.40	PB5	15	PortB.5 (OC1A)
X1_2.37	PB6	16	PortB.6 (OC1B)
X1_2.38	PB7	17	PortB.7 (OC2/OC1C)
X1_1.2	PD0	25	PortD.0 (SCL/INT0)
X1_1.1	PD1	26	PortD.1 (SDA/INT1)
X1_1.4	PD2	27	PortD.2 (TXD1/INT2)
X1_1.3	PD3	28	PortD.3 (TXD1/INT3)
X1_1.6	PD4	29	PortD.4 (IC1)[1]
X1_1.5	PD5	30	PortD.5 (XCK1)[1]
X1_1.8	PD6	31	PortD.6 (T1)[1]
X1_1.7	PD7	32	PortD.7 (T2)[1]
X1_2.51	*PE0*	*2*	*RXD0 (PDI/PortE.0)[4, 5]*
X1_2.52	*PE1*	*3*	*TXD0 (PDO/PortE.1)[4, 5]*
X1_2.49	PE2	4	PortE.2 (XCK0/AIN0)
X1_2.50	PE3	5	PortE.3 (OC3A/AIN1)
X1_2.47	PE4	6	PortE.4 (OC3B/INT4)
X1_2.48	PE5	7	PortE.5 (OC3C/INT5)
X1_2.45	PE6	8	PortE.6 (T3/INT6)

eTW-Interface	signal	ATmega128-Pin	description
X1_2.46	PE7	9	PortE.7 (IC3/INT7)
X1_2.56	PF0	61	PortF.0 (ADC0)
X1_2.55	PF1	60	PortF.1 (ADC1)
X1_2.58	PF2	59	PortF.2 (ADC2)
X1_2.57	PF3	58	PortF.3 (ADC3)
X1_2.60	PF4	57	PortF.4 (ADC4/TCK)
X1_2.59	PF5	56	PortF.5 (ADC5/TMS)
X1_2.62	PF6	55	PortF.6 (ADC6/TDO)
X1_2.61	PF7	54	PortF.7 (ADC7/TDI)
X1_2.35	PG3	18	TOSC2 (PortG.3)[3]
X1_2.36	PG4	19	TOSC1 (PortG.4)[3]
X1_3.82	RES_CS89		RESET-Pin of CS8900A
X1_3.87	RXD+E		receive-input IC11.91 (CS8900A)[3]
X1_3.88	RXD-E		receive-input IC11.91 (CS8900A)[3]
X1_2.66	RXD0_232		receive-input IC8.8 (MAX3232)
X1_2.68	RXD1_232		receive-input IC8.13 (MAX3232)
X1_3.85	TXD+E		transmit-output IC11.87 (CS8900A)[3]
X1_3.86	TXD-E		transmit-output IC11.88 (CS8900A)[3]
X1_2.65	TXD0_232		transmit-output IC8.7 (MAX3232)
X1_2.67	TXD1_232		transmit-output IC8.14 (MAX3232)
X1_1.9	/RD	34	/RD (PortG.1)[1]
X1_2.64	/RESET	20	/RESET[3]
X1_3.81	/SLEEP		/SLEEP-Pin of CS8900A
X1_1.10	/WR	33	/WR (PortG.0)[1]
X1_3.69	+5VD		+5V (CS8900, SRAM and so on)
X1_3.70	+5VD		+5V (CS8900, SRAM and so on)
X1_3.71	+5VD_AVR	52, 21	+5V ATmega128
X1_3.72	+5VD_AVR	52, 21	+5V ATmega128

Table 2: *Pins of eTW-Interface sorted with* ***Pin-Names***

1 internal and external signal (see power of driver and delay of driver);

2 only internal signal;

3 internal signal + **external** signal with solder strap (note: reduced internal functionality);

4 external signal + **internal** signal with solder strap (note: reduced external functionality);

5 signals are necessary for programming adapter (note: the *Prog-Adapter uses a data multiplexer (74HC4053)* -- reduced electric functionality;

User-Interface

Debug-Interface JP2

Debug-Interface	signal	ATmega128-Pin	description
JP1.1	+5VD		+5 power voltage
JP1.2	/RES_AVR	20	/Reset of ATmega128
JP1.3	PF7	54	PortF.7 (ADC7/TDI)
JP1.4	PF6	55	PortF.6 (ADC6/TDO)
JP1.5	PF5	56	PortF.5 (ADC5/TMS)
JP1.6	PF4	57	PortF.4 (ADC4/TCK)
JP1.7	PB1		X2.10 eTW-Interface
JP1.8	SCK	11	SCK (PortB.1)
JP1.9	PE0		X2.17 eTW-Interface
JP1.10	PDI	2	RXD0 (PDI/PortE.0)
JP1.11	PE1		X2.18 eTW-Interface
JP1.12	PDO	3	TXD0 (PDO/PortE.1)
JP1.13	DGND		GND
JP1.14	PRG_CLK	24	Programmer-Clock for ATmega128, see SJ3-solder strap

Table 3: *Pins of Debug-Interface*

UART0-Interface JP3

UART0-Interface	signal	description
JP5.1	NC	not connected
JP5.2	TXD0_232	transmit-signal from RS232-driver
JP5.3	RXD0_232	Receive-signal to RS232-driver
JP5.4	NC	not connected
JP5.5	DGND	Digital-GND
JP5.6 .. JP5.10	NC	not connected

Table 4: *UART0 interface*

UART1-Interface JP4

UART0-Interface	signal	description
JP6.1	NC	not connected
JP6.2	TXD1_232	transmit-signal from RS232-driver
JP6.3	RXD1_232	receive-signal to RS232-driver
JP6.4	NC	not connected
JP6.5	DGND	Digital-GND
JP6.6 .. JP6.10	NC	not connected

Table 5: *UART1 interface*

Power-Interface P1

A Klinkenstecker with 2,1mm Stift has been intended here. The following voltage supply works with polarity protection so that any jack (socket) with 2,1mm and 2,5mm (functions in most cases) can be used with the corresponding voltage (8V .. 12V).

RJ45-Ethernet-Buchse JP7

RJ45	signal	description
JP9.1	TD+	TDX+ signal from transformer
JP9.2	TD-	TDX- signal from transformer
JP9.3	RD+	RDX+ signal from transformer
JP9.4	NC	not connected
JP9.5	NC	not connected
JP9.6	RD-	RDX- signal from transformer
JP9.7	NC	not connected
JP9.8	NC	not connected
JP9.S1	GNDS	Schirm-Potential
JP9.S2	GNDS	Schirm-Potential

Table 6: *ethernet interface*

LED-Functions

The ATmega128 uses the ports PD4 to PD7 to trigger. The LED's are switched on with the L-level and off with the H-level at the port.
REMARK: The high power driver of the port output of the ATmega128 (>20mA) enables the parallel running of the LED and an external triggering.

position of LED's on the easyToWeb-Module

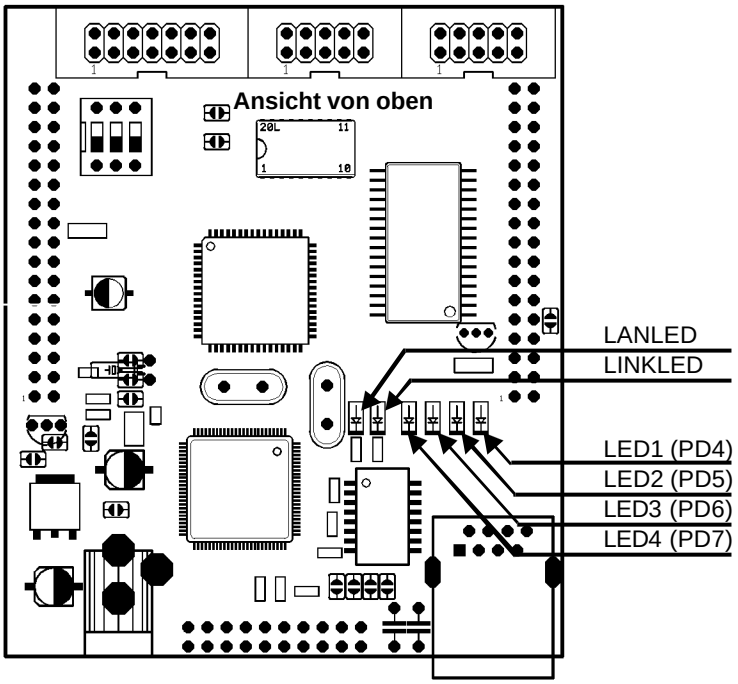


figure 8: *position of LED's*

Functions description

The LED1 to LED4, see figure 8, can freely be employed by the user.
The pre-installed "easyToWeb"-software defines following functions for the LED1 to LED4.

- **LED1 - LED4:** are activated in the start-phase of the "easyToWeb"-software and go on for a few milliseconds. When the main loop is reached, the LEDs are switched off.. (File: "avr_128.c"; "avr_web.c").
- **LED1 (gelb):** flashing of the LED in the loop of the main-routine (File: "avr_web.c")
- **LED2 (rot):** goes on when accessing to the implemented HTTP-server (File: "easyweb.c") in the function " easyweb_HTTPServer()".
- **LED3 (rot):** this LED can be used for some measurements
- **LED4 (rot):**
  Hardware-Version 1.1:
  the LED can be used for one's own measurements;
  Hardware-Version > 1.2:
  PortD.7 is used for the DS2430 and is not available for one's own measurements or only as 1wire-bus;

The LANLED and LINKLED, see figure 8, are triggered by the ethernetcontroller and signals the function-mode of the CS8900A. The LED's are switched with the L-level.

- **LANLED (grün):** signals the sending, receiving and the identification of collisions with an impuls of at least 6ms.
- **LINKLED (grün):** is activated at each reception of a valid 10Base-T Link-Puls. This LED can also be employed by the user. For this aim, the BitE (HCB0) and BitC (HC0E) in register15 (SelfControl) should be adequately triggered.

Software-definition

Macros are defined in the file "avr128.h" for an usage with the C-compiler CodevisionAVR.

LED power consumption

LED1 until LED4 with 1kOhm and 5 Volt - about 3mA per LED
LANLED and LINKLED with 560 Ohm and 5V - about 5mA per LED

Reset- circuits of the easyToWeb-AVR

The brand new generation of the Atmel AVR-Microcontroller (ATmega-Serie) possesses an ameliorated internal reset-logic. The problems (90AT1200 usw.), which solutions required external circuits with reset-controller, belong now definetly to the past.
For special applications is meanwhile an external Beschaltung of the reset-pins neccessary.
Figure 9 shows a port of schematic of the easyToWeb-AVR-module.
Remark: The ATmega128 has no bidirectional reset-pin. This means, the microcontroller regenerates no external reset-signal.

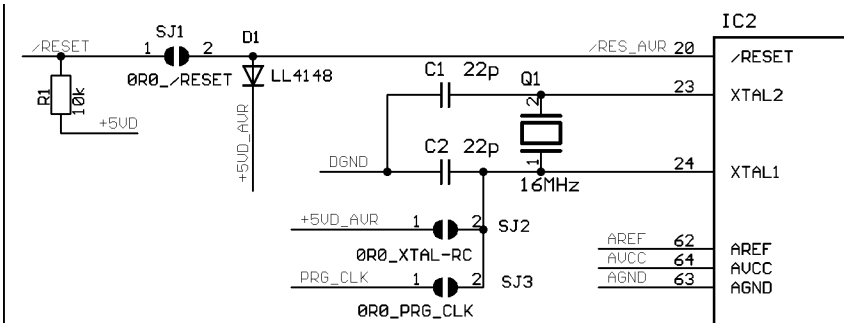


Bild 9: *reset schematic of ATmega128*

positions of components and descriptions

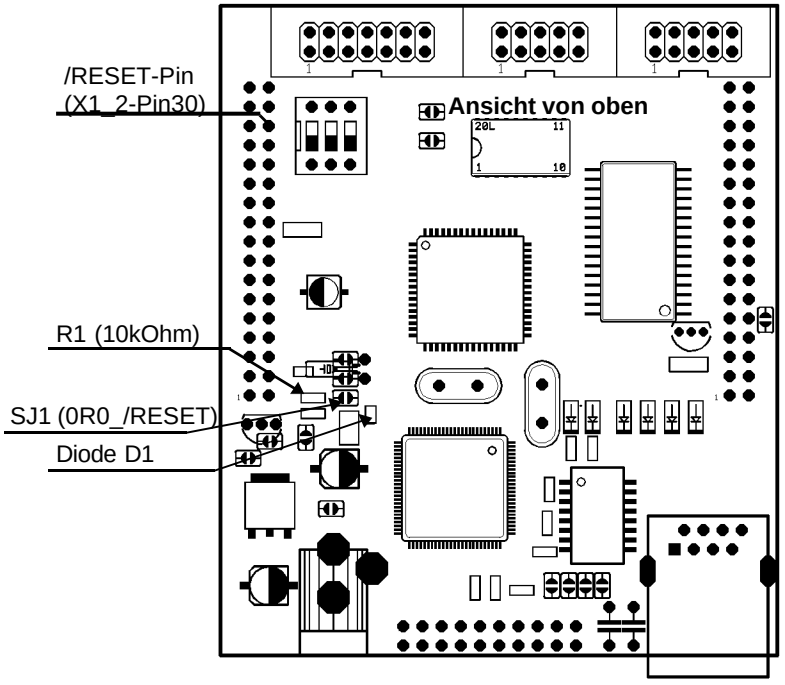


figure 10: *placement options of reset-wiring*

- SJ1-solder strap activate the external reset function. It is connected Pin30 of X1_2 with the Reset-Pin of ATmega128.
- The diode D1 is a protection for a better EMV protection. This one protects the lacking internal diode which abducts the over – voltage to the Vcc.
  The reset-pin has a double function. In programm-mode of the ATmega128, the parallel

programm function is activated by applying a voltage of 12V.
Caution: The diode D1 must be removed for the usage of this function.

- The resistance R1 is used as an additional pull-up to the internal pull-up and spares an eventually external wiring. The internal pull-up resistance is 33kΩ ... 100kΩ worth. Detailed explanations on timing and set-up in [2] are given in the paragraph "System Control and Reset".

delivery of Reset- circuits

- SJ1: open (SMD 0805);
- D1: unmount (SMD LL-34; 1N4148);
- R1: unmount (SMD 0805);

A later usage of the external reset-function require the closing of the SJ1 solder strap (0 Ohm-resistance of the SMD-Bauform 0805 or with tin solder).

Debug-Interface

Debug-Interface plugged : S1.1, S1.2, S1.3 open
Debug-Interface removed : S1.1, S1.2, S1.3 closed (ON)

For an easier debugg-acces to the ressources of the ATmega128, a debug-interface has been realised on the "easyToWeb-AVR" module. For this aim, following functions have been modified:

- Surer separation of the ISP-signals of the ATmega128 from the program-port (here parallel-port of PC);
- Possibility of connection of the JTAG-Port from ATmega128 (if possible JTAG ICE connection occupancy);
- smaller efforts at the easyToWeb-Module;

The entire realisation of this functions isn`t practicable on the "easyToWeb-AVR" module. For this reason, the external **Prog-Adapter** has been conceived. Meanwhile, some components have been realised on the "easyToWeb-AVR module:

- the real debug-junction (JP1) - a 14-pin-male connector (see paragraph User-Interface),
- DIP switch (S1) for the disactivation of the ISP-signals (SCK, PDI, PDO) of the ATmega128.

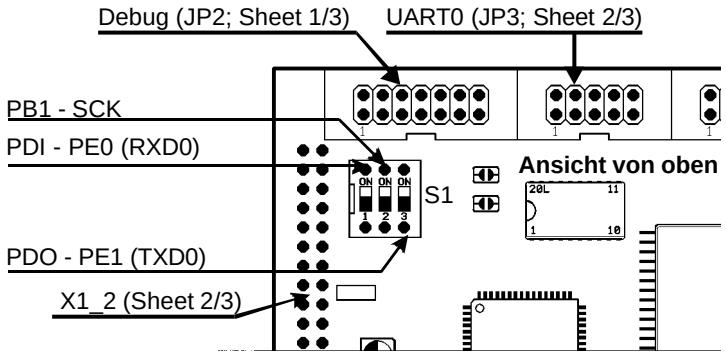


figure 11: *Debug-Interface and ISP-Jumper*

name	description		default-setting
S1.1	***PDI - PE0***		ON
	OFF	• disconnect PDI-Pin (PortE.0) of ATmega128 from connection X1_2.17 of eTW-Interface; • Multiplexers (74HC4053) of Prog-Adapter switches these pin to connection X1_2.17;	
	ON	• connect PDI-Pin (PortE.0) of ATmega128 and connection X1_2.17 of eTW-Interface directly	

name	description	default-setting
S1.2	***SCK - PB1***	ON
	OFF • disconnect SCK-Pin (PortB.1) of ATmega128 from connection X1_2.10 of eTW-Interface; • Multiplexers (74HC4053) of Prog-Adapter switches these pin to connection X1_2.10;	
	ON • connect SCK-Pin (PortB.1) of ATmega128 and connection X1_2.10 of eTW-Interface directly	
S1.3	***PDO - PE1***	ON
	OFF • disconnect PDO-Pin (PortE.1) of ATmega128 from connection X1_2.18 of eTW-Interface; • Multiplexers (74HC4053) of Prog-Adapter switches these pin to connection X1_2.18;	
	ON • connect PDO-Pin (PortE.1) of ATmega128 and connection X1_2.18 of eTW-Interface directly	

Table 7: *Settings of S1-DIP-switch*

solder straps of easyToWeb-AVR

The easyToWeb-AVR module allows the adjustment of different additional functions and options through the activation or disactivation of solder- straps. Following adjust-possibilities are so controlled:

- selections of SRAM typ either 32kByte or 64kByte,
- activation of RS232-Driver,
- use if internal Power adapter,
- activation PortG.4 and PortG.3 (32768kHz crystal),
- activation of 1Wire-Bus with PortD.7,
- modification of reset function,
- expansion of external clock of ATmega128,
- connection between AGND- and DGND signal,
- connection of Ethernet-Interface to external board.

Table 8 shows the adjustments of the solder straps and the figures 13 and 14 describe the position of solder straps.

name	description		default-setting
SJ1	***0R0_/RESET***	***top (Sheet 1/3)***	open
	• connection between ATmega128 (Pin 20; /RESET) with connector X1_2-Pin 30;		
SJ2	***0R0_XTAL_RC***	***bottom (Sheet 1/3)***	open
	• resistor-wiring for external RC oscillator;		
SJ3	***0R0_PRG_CLK***	***bottom (Sheet 1/3)***	open
	• connection between programm-clock with ATmega128 (Pin 24; XTAL1);		
SJ4	***0R0_TOSC1***	***top (Sheet 1/3)***	open
	• connection between Port PG4 (TOSC1-Input) and connector X1_2-Pin 2;		
SJ5	***0R0_TOSC2***	***top (Sheet 1/3)***	open
	• connection betweenPort PG3 (TOSC2-Input) and connector X1_2-Pin 1;		
SJ6	***0R0_1WIRE***	***top (Sheet 1/3)***	close
	• connection between Port PD7 and internal 1WIRE signal (DS2430A)		
SJ7	***0R0_A15***	***bottom (Sheet 1/3)***	close
	• connection between signal A15 (Pin 42; PortC.7) with external SRAM (Pin 2; /CS1); • option to use 32kByte SRAM; **attention: close SJ7 and SJ8 not together**		
SJ8	***0R0_A15_GND***	***bottom (Sheet 1/3)***	open
	• connect chip select of external SRAM to GND (Pin 2; /CS1); • option to use 64kByte SRAM; (use whole external address range of ATmega128); **attention: close SJ7 and SJ8 not together**		
SJ9	***0R0_RXD0***	***bottom (Sheet 2/3)***	close
	• connection between Receive-Output2 of IC8 and Port PE0 (UART0 Receive) of ATmega128;		
SJ10	***0R0_TXD0***	***bottom (Sheet 2/3)***	close
	1. connection between Transmit-Input2 von IC8 and Port PE1 (UART0 Transmit) of ATmega128;		
SJ11	***0R0_RDX1***	***top (Sheet 2/3)***	close

name	description		default-setting
	• connection between Receive-Output1 of IC8 and Port PD2 (UART1 Receive) of ATmega128;		
SJ12	***0R0_TXD1***	***top (Sheet 2/3)***	close
	• connection between Transmit-Input1 of IC8 and Port PD3 (UART1 Transmit) of ATmega128;		
SJ13	***0R0_VCC_FLASH***	***top (Sheet 2/3)***	open
	• connection between operating voltage (+5VD) and +3.3V (necessary for a 3.3V version);		
SJ14	***0R0_5V_EXT***	***top (Sheet 2/3)***	close
	• connection between internal power adapter (+5V_E) and internal voltage +5VD; • leave open if power supply from connector X1_3-Pin 1 and X1_3-Pin 2;		
SJ15	***0R0_AGND***	***top (Sheet 2/3)***	close
	• connection between DGND and AGND;		
SJ16	***0R0_5V_EXT_AVR***	***top (Sheet 2/3)***	close
	• connection between internal power adapter (+5V_E) and internal voltage +5VD_AVR; • leave open if power supply from connector X1_3-Pin 1 and X1_3-Pin 2;		
SJ17	***0R0_TXD+***	***top (Sheet 3/3)***	open
	• connection between Ethernet Controller (CS8900; Pin 87) and connector X1_3-Pin 17;		
SJ18	***0R0_TXD-***	***top (Sheet 3/3)***	open
	• connection between Ethernet Controller (CS8900; Pin 88) and connector X3-Pin 18;		
SJ19	***0R0_RXD+***	***top (Sheet 3/3)***	open
	• connection between Ethernet Controller (CS8900; Pin 91) and connector X3-Pin 19;		
SJ20	***0R0_RXD-***	***top (Sheet 3/3)***	open
	• connection between Ethernet Controller (CS8900; Pin 92) and Connector X3-Pin 20;		

Table 8: *list of solder straps of the "easyToWeb-AVR"-module*

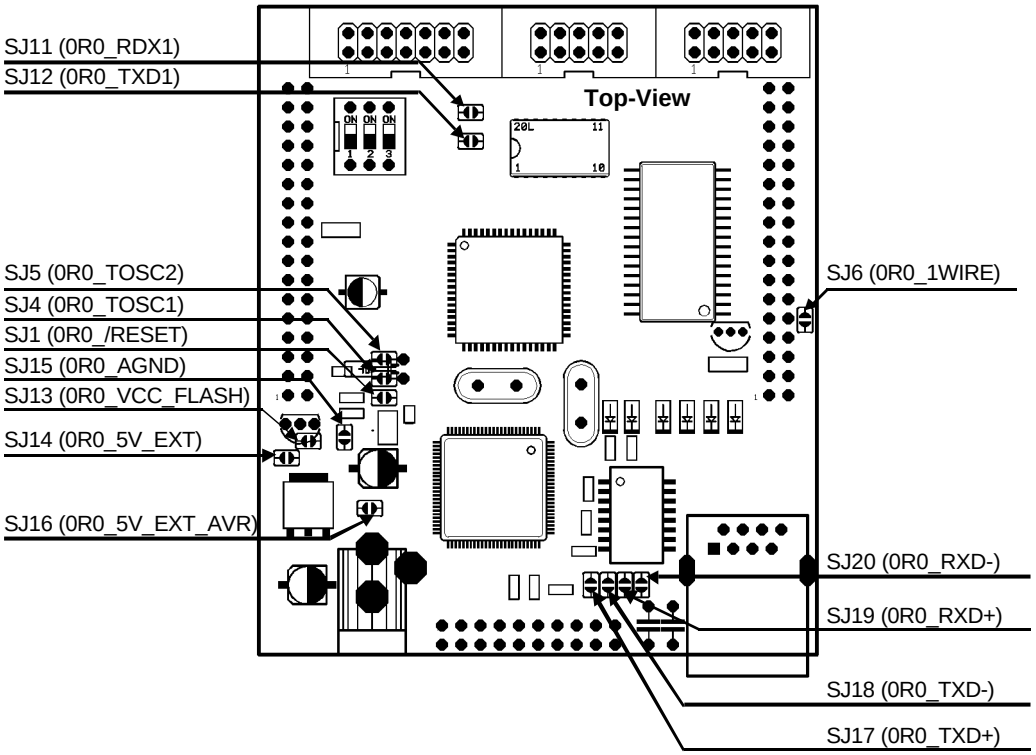


figure 13: *solder straps - Top-View*

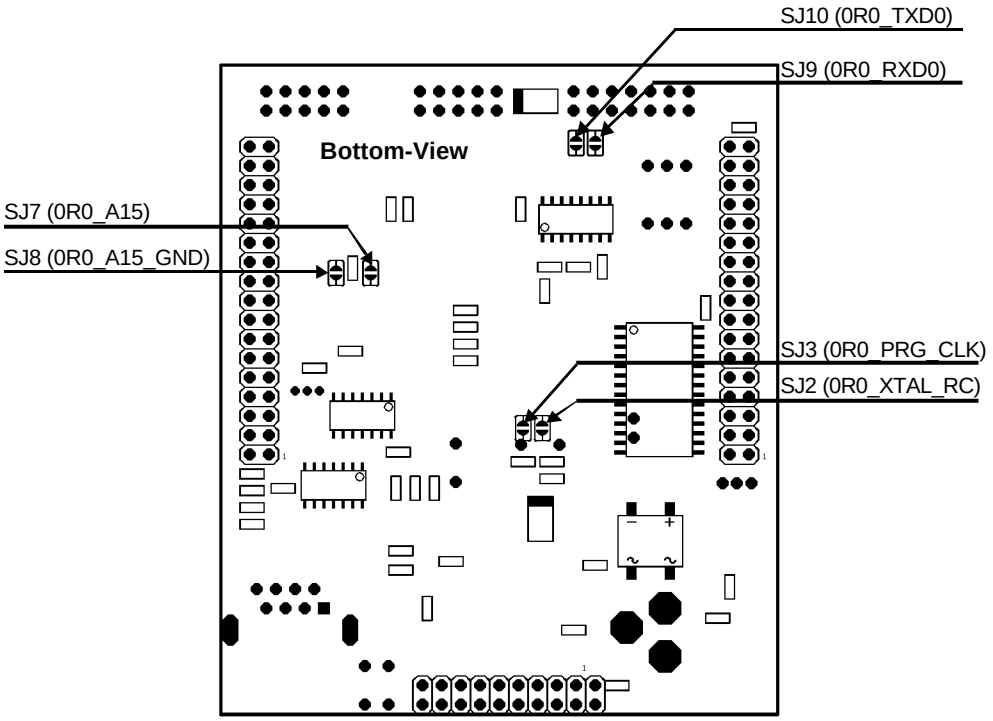


figure 14: *solder straps - Bottom-View*

mechanical dimensions

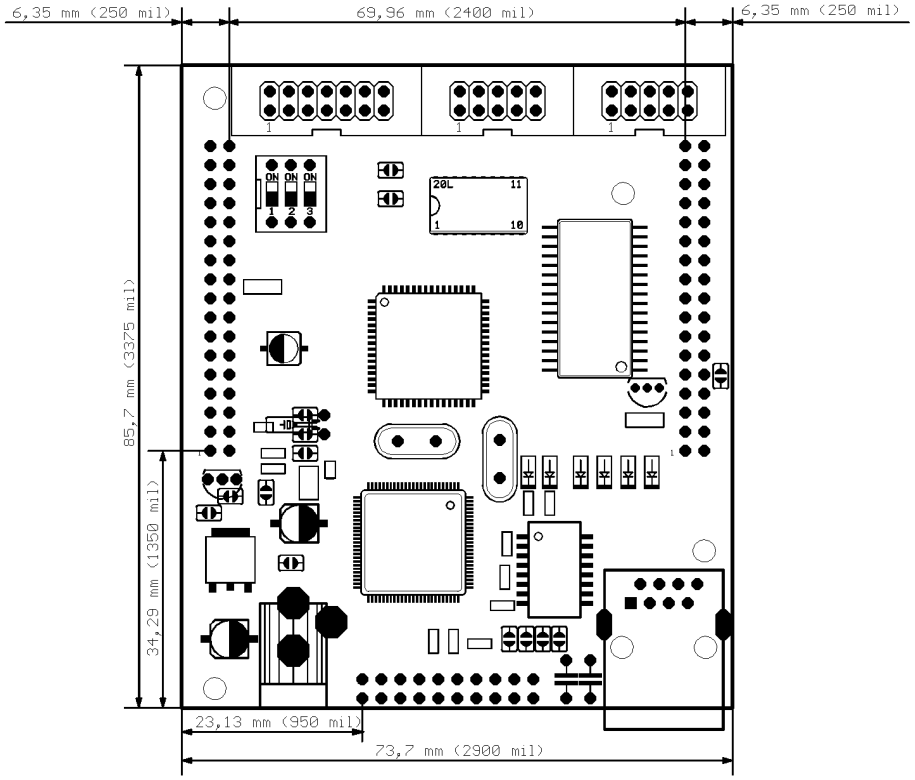


figure 15: *mechanical dimensions*

fastening drilling

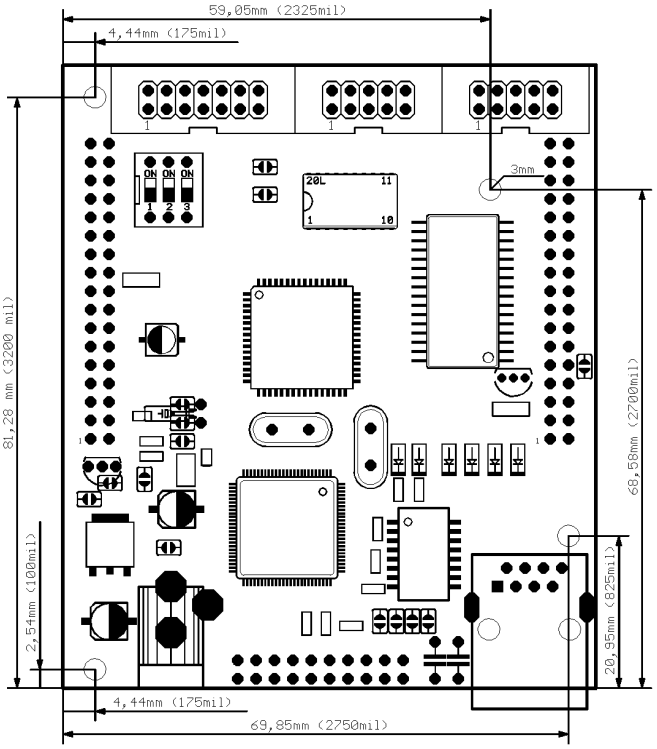
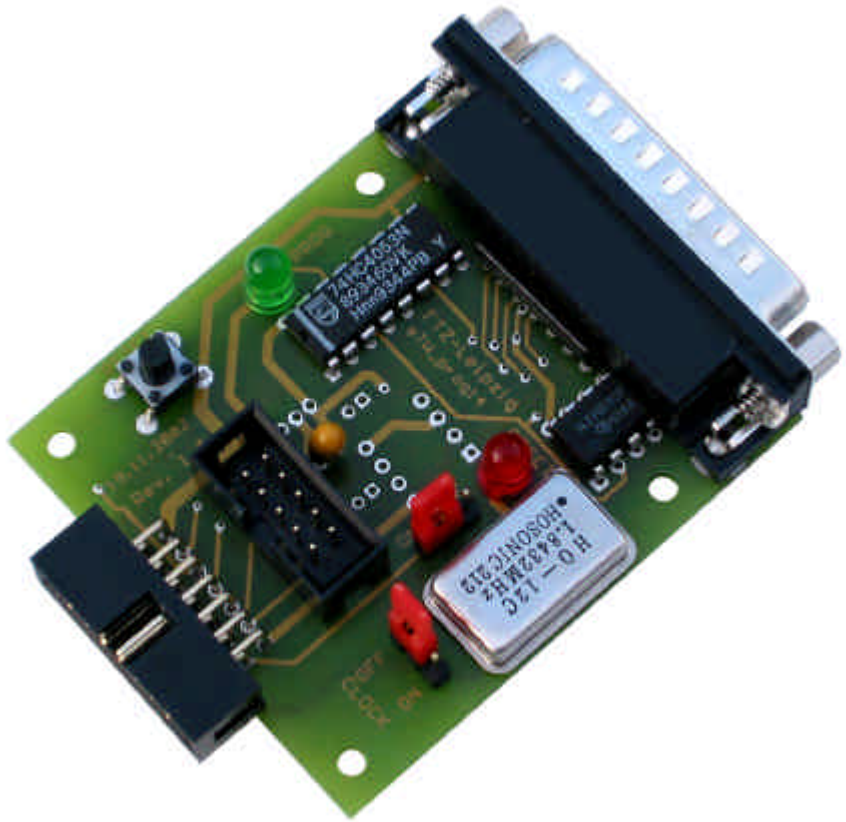


figure 17: *positions of fastening drilling*

Prog-Adapter (Hardware-Version 1.3)



The prog-adapter is a simple hardware-Interface for programming the "easyToWeb-AVR"-module. Followind functions are realised:

- Coupling of the ISP-Interface of the ATmega128 to the parallel port of the PC (functionally identic with the STK200 or STK300 of ATMEL);
- Isolation of the ISP-signals of the ATmega128 (SCK, PDI, PDO) of the parallel port of the PC;
- Reset-Beschaltung made of manual release of the RESET-signals and reset-controller;
- JTAG-Adapter on standard allocation of the JTAG ICE of ATMEL;

Multiplexer of the ISP-Interface

For a surer and good functioning of the ISP-Interface of the ATmega128, ATMEL advises the usage of a multiplexer. This one isolates the external signals at the ISP (PB1, PE0, PE1) from the programm Interface (here parallel port of the PC). The IC5 realises this activation, figure 18.
ATTENTION:
The electrical properties of PB1, PE0, PE1 are setted at the usage of the prog-adapter of the multiplexer IC3 (74HC4053). It concerns especially the voltage stability and the signal speed. For this aim the data sheet ("74HC4053_schs122c.pdf") on the CD-ROM should be read.

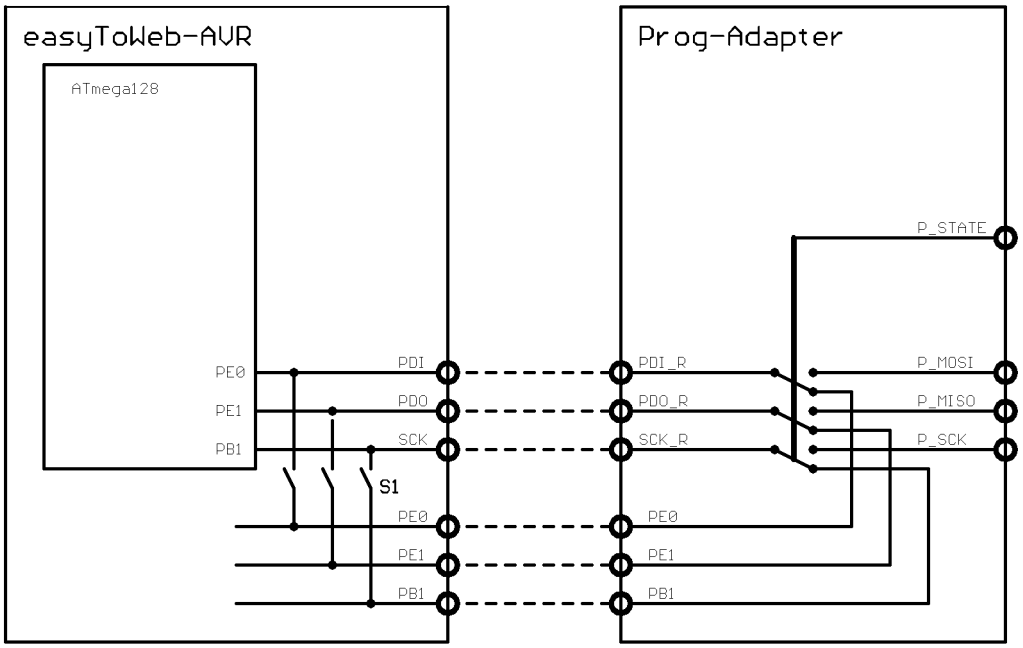


figure 18: *schematic of multiplexing of the programm lines*

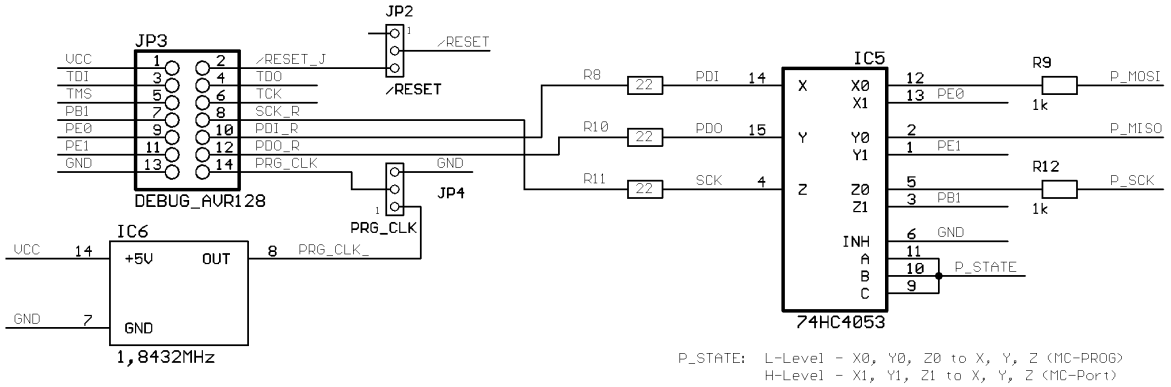


Bild 19: *schematic of multiplexers 74HC4053*

PRG_CLK of the Prog-Adapter

The experiences with the ATmega128 have shown that the programming of the internal FUSE-bits of the ATmega128 could disable the ISP interface. When the FUSE-bits CKSEL 3..0 of the ATmega128 are in the mode "external Clock with RC-combination", the main clock works and with lacking RC-Beschaltung no more. The ATmega128 is then no more programmable over the ISP-interface.
A 1,8432Mhz clock is then available on the programm adaptor.The jumper JP3 should additionally be connected to the prog-adapter and the soulder-bridge JP3 to the "easyToWeb-AVR"-Module.

ATTENTION - PRG_CLK==1,8432Mhz:
For a good funcrtioning with activated PRG_CLK, the functionality of the software should be verified (baudrates, timer etc.).

connections of the Prog-Adapter

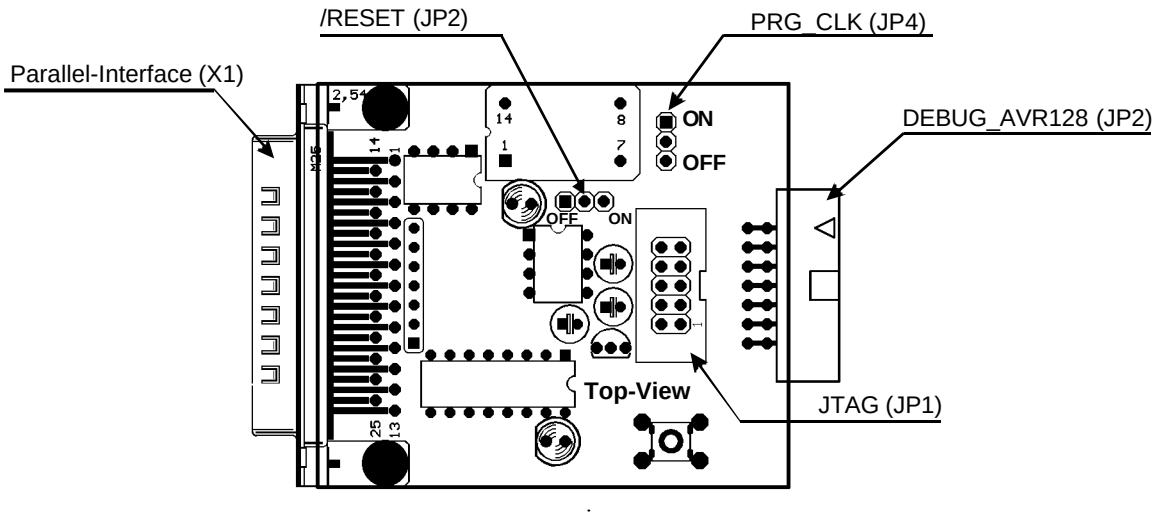


figure 20: *Prog-Adapter external interfaces - top view*

Parallel-Interface X1

Debug-Interface	Signalname	I/O	Beschreibung
X1.1	NC		not connected
X1.2	B1	Input/Output	stud to connection X1.12
X1.3	B2	Input/Output	stud to connection X1.11
X1.4	DATA2	Input	Enable-signal for driver IC4B (/RES_MR and P_MISO)
X1.5	DATA3	Input	Enable-signal for driver IC4A (P_MOSI, P_STATE and P_SCK)
X1.6	DATA4	Input	P_SCK Signal over IC4A.6 and IC4A.8
X1.7	DATA5	Input	P_MOSI Signal over IC4A.2
X1.8	DATA6	Input	P_STATE Signal over IC4A.4
X1.9	DATA7	Input	/RES_MR Signal over IC4B.13, IC4B.15 und IC4B.17
X1.10	ACK	Output	P_MISO Signal from IC4B.9
X1.11	B2	Input/Output	stud to connection X1.3
X1.12	B1	Input/Output	stud to connection X1.2
X1.13 .. X1.17	NC		not connected
X1.18 .. X1.25	GND		ground

Table 10: *connections of PC-Parallel interface*

JTAG-Interface JP1

JTAG-Interface	Signalname	I/O	Beschreibung
JP1.1	TCK	Input	Test-Clock
JP1.2	GND		ground
JP1.3	TDO	Output	Test-Data-Output
JP1.4	+5V		Betriebsspannung
JP1.5	TMS	Input	Test-Mode-Select
JP1.6	/RES_MR	Input	RESET-Signal
JP1.7, JP1.8	NC		not connected
JP1.9	TDI	Input	Test-Data-Input
JP1.10	GND		ground

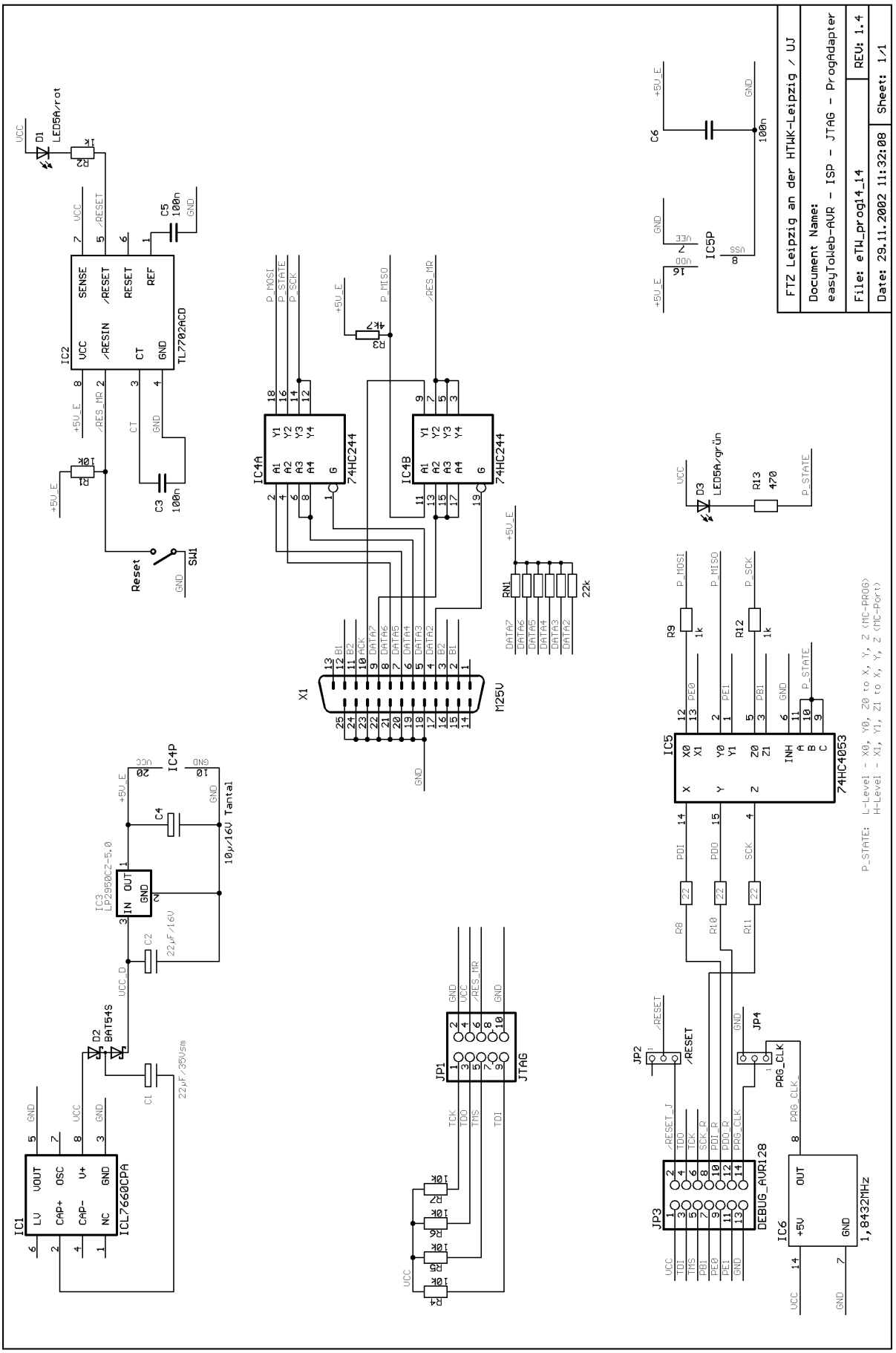
Table 11: *connections of JTAG interface*

Debug-Interface JP2

Debug-Interface	Signalname	I/O	Beschreibung
JP2.1	+5VD		+5 power supply
JP2.2	/RESET	Output	/Reset to ATmega128
JP2.3	TDI	Output	Test-Data-Input PortF.7 (ADC7/TDI)
JP2.4	TDO	Input	Test-Data-Output PortF.6 (ADC6/TDO)
JP2.5	TMS	Output	Test-Mode-Select PortF.5 (ADC5/TMS)
JP2.6	TCK	Output	Test-Clock PortF.4 (ADC4/TCK)
JP2.7	PB1	I/O	X2.10 eTW-Interface
JP2.8	SCK	Output	SCK (PortB.1)
JP2.9	PE0	I/O	X2.17 eTW-Interface
JP2.10	PDI	Output	RXD0 (PDI/PortE.0)
JP2.11	PE1	I/O	X2.18 eTW-Interface
JP2.12	PDO	Input	TXD0 (PDO/PortE.1)
JP2.13	DGND		ground
JP2.14	**PRG_CLK**	**Output**	**1,8432Mhz Clock for ATmega128**

Tabelle 12: *connections of debug interface*

Prog-Adapter schematic circuit diagram sheet 1/1



literature

[1] Sturm, M. (Hrsg.): Embedded Internet. Sonderheft Design&Elektronik, Poing: WEKA Verlag, September 2001

[2] ATMEL: 8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash. Rev. 2468F-AVR-05/02; ATMEL Corporation 2002

[3] Sturm, M. (Hrsg.): Embedded Computing. Design&Elektronik, Poing: WEKA Verlag, August 2002