Title: **Documentation for AVR J1850 VPW Interface**
Issue: F
Author:Michael Wolf
Printed: 22/10/2005

## Overview

AVR J1850 VPW interface to connect a Chrysler or GM car bus to a PC for On Board Diagnostic (OBD) monitoring. My interface is build around Atmel AVR mega8 controller, my favourite “workhorse”. The controller is available also in DIP package for all who not want to built with SMD components. With 8k flash memory we have plenty of space for all kind of features. The basic source code which supports all functions uses 3k code space.

Some features:

- SAE J1850 VPW to RS232 interface
- simple AT command control
- different Baud rates, from 9600 to 115200 Baud
- 4 different bus monitor functions
- support for 1 and 3 byte header messages

The source code for this interface is released under GNU GENERAL PUBLIC LICENSE. See license.txt for details.

## Hardware description

The circuit is designed for a single layer PCB.

| | |
|---|---|
| K1 | VPW bus connector<br>Pin 1 = Ground<br>Pin 7 = VPW bus In/Out<br>Pin 9 = 12V supply |
| K2 | RS232 serial connector<br>Pin 2 = Serial out<br>Pin 3 = Serial in<br>Pin 5 = Ground |
| K3 | In-Circuit-Programming connector<br>Compatible pin-out for Atmel STK500 development board<br>Pin 1 = MISO<br>Pin 2 = 5V supply<br>Pin 3 = SCK<br>Pin 4 = MOSI<br>Pin 5 = Reset<br>Pin 6 = Ground |
| IC1 | 5V fixed voltage converter to supply the uC and RS232 converter |
| D1 | Power on indication |
| D2, T1 | VPW bus traffic indication, incoming and outgoing |
| T2 | VPW input buffer, signal inverter and voltage converter |
| D3, D4, T3, T4 | VPW output buffer, signal inverter and voltage converter |
| IC2 | main controller |
| IC3 | RS232 level converter |

Note:
Firmware version 1.04 and higher requires a crystal with 7.3728MHz for proper function.

**Important note:**

**The voltage regulation circuit R8/D4 is designed for at least 315 Ohm bus load!**
**Do not keep the J1850 line in high level without load resistor connected to K1-7 and K1-1 otherwise D4 will be destroyed immediately.**

## Programming the AVR

The circuit provides an ISP connector to program the firmware. Use your favourite programming hardware to flash the AVR.
After flashing the firmware you must change the fuse settings on new devices. Remember that a “clean” ATmega8 is delivered with internal RC oscillator enabled, running at 1MHz.
Program (checked) the fuses in the following order:
In AVR-Studio:

- Boot flash section size=128 words Boot start address=$0F80; BOOTSZ=11
- Brown-out detection level at VCC=4.0V; BODLEVEL=0
- Brown-out detection enabled; BODEN=0
- CKOPT fuse; CKOPT=0
- Ext. Crystal/resonator High Freq.; Start-up time: 1k CK + 0ms; CKSEL=1110 SUT=10

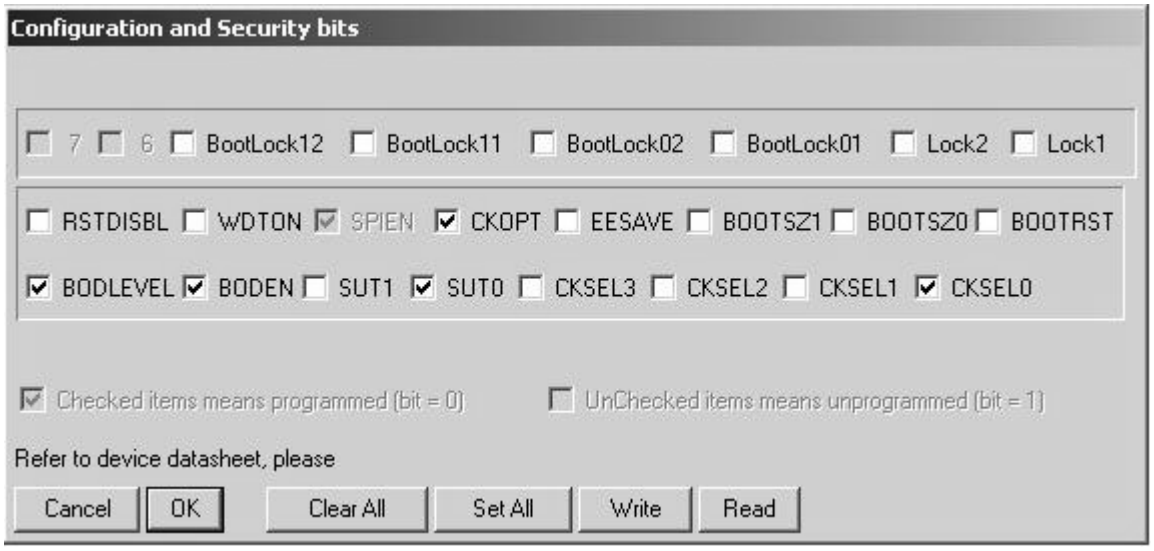
All other fuses are not used.

In PonyProg:

**Checked** fuses:
SUT1, CKSEL0

**Unchecked** Fuses:
BOOTSZ1, BOOTSZ0

**Leave the rest untouched!**



## Communicating with the interface

Communication is done via serial RS232 connection.
Default connection settings are:
9600 (default) or 19200 Baud
8N1 - 8 data bits, no parity bit, 1 stop bit, no handshake

All responses from the interface are terminated with a single carriage return character and, by default, a line feed character as well.
After correct connection and power up, the interface will display the initial message:

AVR-J1850 VPW v1.04
May 10 2005 / 18:33:53

>

Showing code description and version, date and time when the code was compiled and the command prompt character '>'.
The interface is now ready and can receiver user commands. All commands used to control the interface are separated from vehicle commands with the command tag "AT". Vehicle commands can contain only with ASCII characters for hexadecimal digits (0-9 and A-F).
All commands send to the interface, internal AT or hex string for VPW, must be terminated with a single carriage return character (0x0D) before they will be accepted, they are not case sensitive.
A single "?" will indicate that the message was misunderstood by the interface (syntax error). This includes wrong or incomplete AT commands and wrong hex strings. This is not a sign for invalid VPW message because the interface not validates VPW message.

## AT command description

An interface control command must begin with the "AT" command tag followed by a command code and, if required, one or more parameters.

**AR** [Automatic Receive address] Default: ON

Set the receive address is equal to the third header byte for physical addressing or second header byte +1 for functional addressing.
The receive address changes whenever the header bytes changes.

**Bxx** [set Baud rates] Default: 115200 Baud

Set desired Baud rate. The interface will change the Baud rate immediately after executing the command. Status message is returned with new Baud rate set.
Possible values for xx:

| | |
|---|---|
| 0 = 9600 Baud | 1 = 14400 Baud |
| 2 = 19200 Baud | 3 = 28800 Baud |
| 4 = 38400 Baud | 5 = 57600 Baud |
| Any other value = 115200 Baud | |

Note: Default Baud rate is configured in file main.h

**D** [set Defaults]

Set all interface values to default.

**E0** or **E1** [Echo on/off] Default: ON

With echo on all characters received via RS232 are echoed back to user terminal.

**FD** [Formatted Data] Default: ON

All responses are returned as standard ASCII characters readable on any terminal program. Hex values are returned as 2 byte ASCII values, separated by space and terminated by carriage return and optional line feed.

**H0** or **H1** [Header on/off] Default: OFF

Controls whether or not the header and CRC bytes are shown for received VPW messages.

**I** [Ident]

Returns interface identify string.

**L0** or **L1** [Line feed on/off] Default: ON

Controls whether or not a line feed character is returned with each carriage return character.

**MA** [Monitor All]

Set the interface in monitor mode. All message received via VPW bus are returned, optional with header and CRC.
Any serial input will stop this mode and the interface returns to idle state waiting for user inputs.
Note:
You will see garbage output if MA mode is enabled, Header output is disabled and VPW frames are transmitted with one byte header only. Use command AT O1 first to enable one byte header modus for correct output!

**MI** xx [Monitor ID]

Set the interface in monitor mode for a specific frame ID provided as command parameter. The frame ID is the first byte in J1850 messages in single byte header format, i.e. used in Chrysler vehicles.
All message received via VPW bus matching the given frame ID are returned, optional with header and CRC.
Any serial input will stop this mode and the interface returns to idle state waiting for user inputs.
See SAE J2178-Part 3 for details of frame ID and single byte header format.

**MR** xx [Monitor Receiver]

Set the interface in monitor mode for a specific receiver address provided as command parameter. All message received via VPW bus matching the given receiver address are returned, optional with header and CRC.
Any serial input will stop this mode and the interface returns to idle state waiting for user inputs.

**MT** xx [Monitor Transmitter]

Set the interface in monitor mode for a specific transmitter address provided as command parameter. All message received via VPW bus matching the given transmitter address are returned, optional with header and CRC.
Any serial input will stop this mode and the interface returns to idle state waiting for user inputs.

**O0** or **O1** [One Byte Header on/off] Default: OFF

Use this command to enable or disable transmitting and monitoring of one byte header messages. With this option enabled the interface will send only one byte as header instead of three bytes. The header byte must be set using command AT SH xxyyzz, where xx is the header byte which is will be used. Byte 2 and 3 will be ignored, but must be send to make a valid AT SH command.
In MA mode this command ensures the correct data output if headers are disabled.

**PD** [Packed Data] Default: OFF

All messages received via VPW bus are returned as hex string starting with a single length byte followed by complete message content, without terminating character (carriage return or line feed).
The length byte will indicate only the number of data bytes following, excluding itself.
If a message error was detected (CRC error) the MSB bit 7 of the length byte is set to indicate and error.
If no data was received within the time out a single length byte with value ‘0’ is returned.

**R0** or **R1** [Responses on/off] Default: ON

With responses turned off, the interface will not wait for any response from the vehicle. All given VPW messages are passed to the bus and the interface returns immediately to idle state, waiting for user inputs.
With responses turned on, the interface will wait at least 100ms for node response and an additional timeout which is set via ST command.

**SH** xx yy zz [Set Header]

This command set the three byte header which is send before all user VPW commands.
For details of header bytes see standard SAE J2178 part 1 and SAE J1979.
Default header is 0x68 0x6A 0xF1 for external diagnostic equipment.

**SR** xx [Set Receive Address]

Users can set a specific receive address to which the interface will respond. This command will turn off AR mode and the interface will accept only responses from address xx.

**ST** xx [Set Timeout] Default: 50

This command set the additional timeout before the interface returns an “NO DATA”.
Default value is 50 which give an additional timeout of 4ms x 25 = 100ms.
The command parameter must be send as hex value range 0x00 to 0xFF. Values less than 0x08 will be forced to 0x08 to set a minimum timeout of 32ms.
This timeout recommended to the 100ms minimum node response timeout specified in SAE J1850.

**Z** [Reset]

The interface will perform a complete reset, all values will set to default.

**Note:**
For commands with 0/1 disable/enable option, only a ‘0’ will disable the function, other values than ‘0’ will enable the function!

## ODB or VPW commands

I will not descript VPW or ODB commands here, because you can find a lot of ODB information using you favourite internet search engine and it's not the purpose of this document.

Just a few words how to send VPW messages using the interface:
All VPW messages must be send as hex string given with 2 byte ASCII values. String length must be an even number of ASCII bytes with a maximum number of 8 VPW data bytes. No header byte (set via SH command) and NO CRC.
CRC will be calculated independent for each given hex string will be included automatically to the message send to the bus.
So the maximum message length send to the VPW bus is 12 byte as specified in SAE J1850.

The interface will calculate a CRC for all messages received via VPW bus. This CRC will be compared with the received CRC. In case of CRC error, the interface will mark this message faulty.

## Error messages

The interface will return various error messages; here is a description of each message:

**BUSBUSY**
The interface tried to send a message but bus was busy.

**BUSERROR**
The interface has detected a bus error. It was not possible to change bus levels, or an bit level collision was detected.

**<DATAERROR**
Calculated and received CRC byte not equal, message faulty.

**NO DATA**
No data received within timeout.

**?**
Given command wrong or incomplete or hex string wrong or incomplete.

**OK**
Not an error message, but indicates execution of given command successful.

You will find the interface schematic, PCB layout, part placement layout and bill of material in the appendix of this document.
PCB layout file is available on request. Format will be Target 3001 v11.
(http://www.ibfriedrich.de/)

Any comments, suggestions or critic are welcome.
Feel free to mail me at: michael AT mictronics.de or use my forum at
http://www.mictronics.de

**Disclaimer:**

By using this circuit or description or other material presented, then you (meaning the reader of this material being yourself, or any other person that you may subsequently pass any material to) explicitly accept the following.
I will accept no liability for loss or damages that may be imposed upon you by any Court of Law, statutory or other body. It is entirely the reader's responsibility to determine the suitability of any design for the intended purpose. The project is presented "as is" and are believed to be without error, however this cannot be guaranteed, and it is reasonable to assume that mistakes or other errors will occur from time to time. If errors are found, please contact me, Michael and describe the error (and its consequences) so that corrections may be made.

**License for firmware source code**

AVR J1850 VPW Interface, this firmware and hardware will realize an
interface between a cars J1850 VPW bus and RS232.
Copyright (C) 2004 Michael Wolf

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

---

AVR J1850 VPW Interface, diese Firmware und Hardware realisiert ein
Interface zwischen einem J1850 VPW Fahrzeug Bus und RS232.
Copyright (C) 2004 Michael Wolf

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der
GNU General Public License, wie von der Free Software Foundation
veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß
Version 2 der Lizenz oder jeder späteren Version.

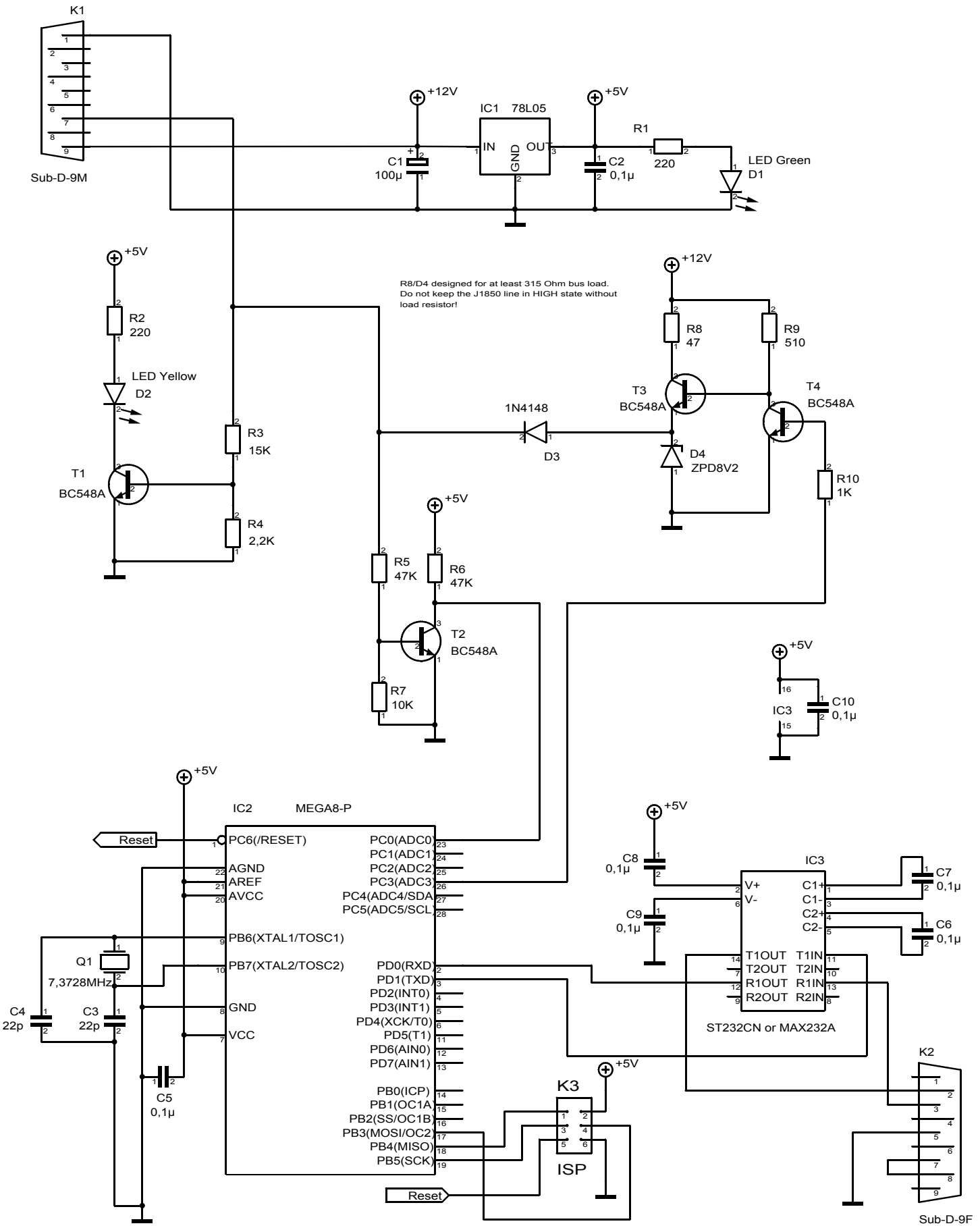
Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, daß es Ihnen
von Nutzen sein wird, aber OHNE IRGENDEINE GARANTIE, sogar ohne die implizite
Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN
ZWECK.
Details finden Sie in der GNU General Public License.

Sie sollten eine Kopie der GNU General Public License zusammen mit diesem
Programm erhalten haben. Falls nicht, schreiben Sie an die Free Software Foundation,
Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

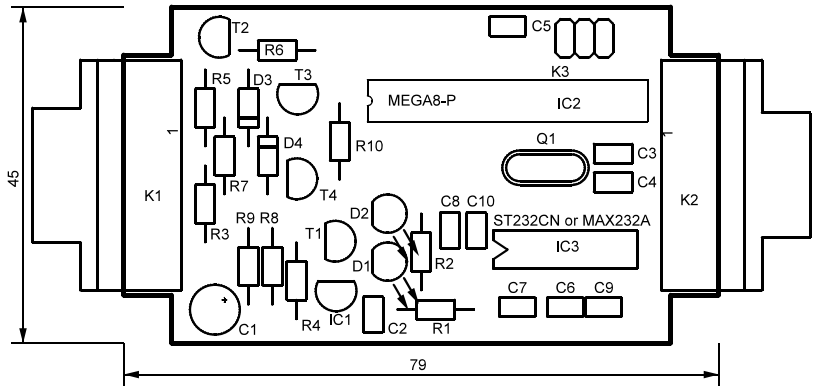
You can contact me, Michael, by:
WWW: http://www.mictronics.de
E-mail: michael@mictronics.de

# AVR J1850 VPW to RS-232 interface schematic

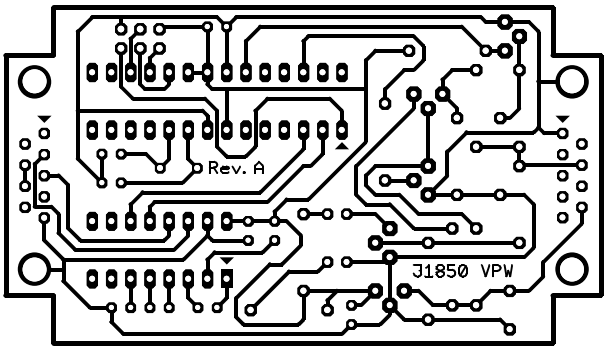
http://www.mictronics.de




| | | | |
|---|---|---|---|
| Maßstab 107,49% | Datei AVR-J1850 VPW Interface.T3000 | Zeichner Rev. B | Blatt 1/1 |
| Änderung 13.12.04 | 20:17 | Titel | |
| Ausgabe 14.05.05 | 12:39 | AVR-J1850 VPW Interface | |
| Firma www.mictronics.de | | Projekt | |


T2
R6
C5
K3
R5
D3
T3
MEGA8-P
IC2
1
D4
R10
Q1
C3
K1
R7
T4
C4
K2
R3
R9
R8
D2
C8
C10
ST232CN or MAX232A
T1
IC3
D1
R2
C7
C6
C9
C1
R4
IC1
C2
R1
45
79

Rev. A
VPM 31850


Rev. A
J1850 VPW

BOM AVR-J1850 VPW Interface
14.November.2004 16:39

| Pos | Name | Value | Package |
|---|---|---|---|
| 1 | C7 | 0,1µ | MKS2.5X4.6 |
| 2 | C6 | 0,1µ | MKS2.5X4.6 |
| 3 | C5 | 0,1µ | MKS2.5X4.6 |
| 4 | C10 | 0,1µ | MKS2.5X4.6 |
| 5 | C9 | 0,1µ | MKS2.5X4.6 |
| 6 | C8 | 0,1µ | MKS2.5X4.6 |
| 7 | C2 | 0,1µ | MKS2.5X4.6 |
| 8 | R10 | 1K | |
| 9 | D3 | 1N4148 | |
| 10 | R4 | 2,2K | |
| 11 | Q1 | 7.3728MHZ | HC-18U |
| 12 | R7 | 10K | |
| 13 | R3 | 15K | |
| 14 | C4 | 22p | |
| 15 | C3 | 22p | |
| 16 | R8 | 47 | |
| 17 | R5 | 47K | |
| 18 | R6 | 47K | |
| 19 | IC1 | 78L05 | TO-92G |
| 20 | C1 | 100µ | ELKO5-6.5 |
| 21 | R2 | 220 | |
| 22 | R1 | 220 | |
| 23 | R9 | 510 | |
| 24 | T2 | BC548A | TO-92G |
| 25 | T1 | BC548A | TO-92G |
| 26 | T4 | BC548A | TO-92G |
| 27 | T3 | BC548A | TO-92G |
| 28 | D1 | LED Green | LED 3mm |
| 29 | D2 | LED Yellow | LED 3mm |
| 30 | IC2 | MEGA8-P | DIL28-3 |
| 31 | IC3 | ST232CN or MAX232A | DIL16 |
| 32 | K2 | Sub-D-9F | SUB-D-9-F-90 |
| 33 | K1 | Sub-D-9M | SUB-D-9-M-90 |
| 34 | D4 | ZPD9V1 | |