AS/400 :: Robots :: Microcontroller :: Electronic :: Downloads :: Links :: Site History

feedback :: home

Microcontroller

Nitron LCD Terminal

MON08 Programming and Debugging circuits

Nitron Oszillator trimming

AB32 Board

HC08 Flash programming

HCS08 Controller

HCS12 Controller

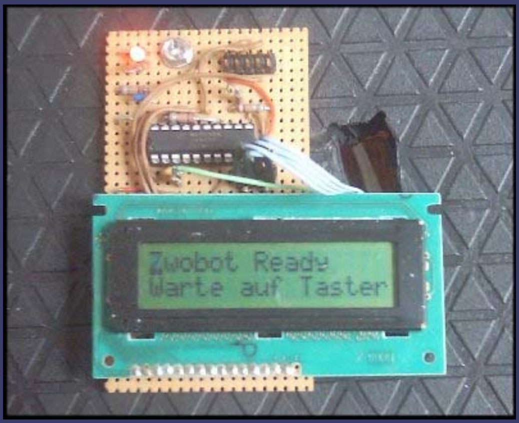
HC12 Welcome Kit

TBDML HCS12 BDM Tool

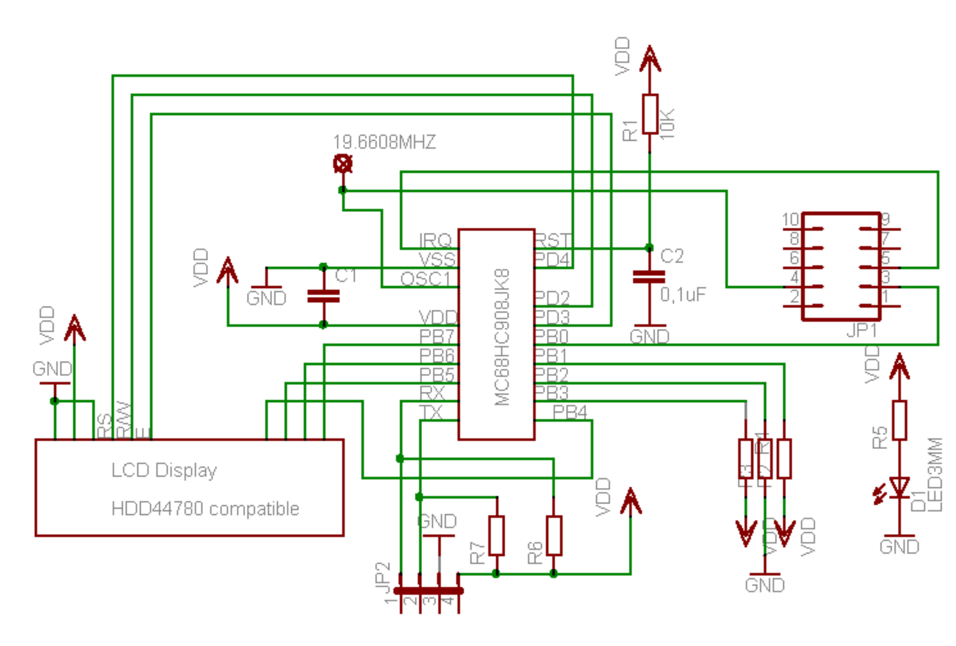
Zwobots Display

Zwobots LCD Controller

The idea behind this is the same as it has been with the **Nitron LCD Terminal** a LC Display with a HDD44780 compatible controller, that is connected via a serial interface that takes care the formatting of the output. This time a MC68HC908JK8 is used as Controller. The software is written in C and the display is used in 4 Bit mode. The Busy Flag is checked to minimize delay times. The JK8 controller has a hardware SCI and is availible in a 20 pin DIP package, so it s easy to solder. This time I dn´t care about standards like vt100 for display commands. All command have a % as prefix. I implemented commands for numerical output of a Byte, displaying a Byte as Bits and displaying a 2 Byte integer. The other comands are cursorpositioning and clearing the display. It s fits on a small piece of prototype board.



The scheatic is quite simple. I connected the data bus D4 - D7 to portt B PB4 to PB7 and the control lines to PD2 to PD4. The other pins on port B are used for monitor mode. The ins on PB2 to PB4 are tied with resistors to the appropriate levels. The clock is derived from Zwobots contrllerboard. The oscillator can drive both controllers without problems. A 4 pin header is used to connect the serial interface an the power supply



The software is written using Codewarrion 5.0. I sed the device initialisation wizard to initialize the SCI. The wizard generates code that initializes the SCI and creates the bodys for the interrupt service routines. I just had to fill in my code into the fnction. Transmittng is used only for synchonisation with Zwobot, so I dont use interruts here. Recieiving is interrupt driven. The ISR puts the recieved byte in a buffer which is processed in the main function. If the clock frequency ore the cntroller is changed, the functions in wait.c have to be adapted. The adress of the ROM routine DELNUS has to be changed ore you must write your own wait function. The rest of the software should run n other controllers without problems, if the ports are adapted. If you dont want to read the busyflag, the wait commands that are comments now can be used. Here you can find the whole project : **ZwobLCD.zip**. If you have troubles to import the project, just create a new porject an add the sources to it.

^

eckhard@eckhard-gosch.de