AVR-Guide

Ich habe hier zusammengeschrieben was ihr braucht um mit dem Atmel ATMega8L los zulegen. Auch erwähne ich noch was für Hard-/Software noch hilfreich sein kann. Falls ihr noch Fragen habt, meldet euch, dann erweitere ich die Doku entsprechend.

Beachtet, wir nutzen den ATMega8L zur Zeit um die Software zu entwickeln, das finale Board wird aller Voraussicht ein SMD Board sein mit einem ATMega 168 und einer komplett neuen Spannungsversorgung drauf, die noch sparsamer sein wird.

Welche Hardware brauche ich?

- EIB Spannungsversorgung (bekommt man auf Ebay mit 640mA für ca. 100EUR), Achtung 28-30V mit integrierter Drossel kaufen! Man kann auch ein Labornetzteil mit einer EIB Drossel nutzen, aber bedenkt, ihr braucht sowieso früher oder später ein Netzteil für eure Installation. Solange ihr nur ein paar Geräte auf einem kurzen Bus dranhängen wollt, geht auch die Drossel auf unserer Seite, die dann mit einem Labornetzteil kombiniert werden kann.
- Einen EIB Aktor (hab mir da einen Gira 2-fach Taster auf Ebay für 39 EUR gekauft), wird gebraucht um Telegramme auf dem Bus zu versenden.
- Für die Programmierung der EIB Geräte entweder einen Busankoppler mit einem RS232 Pegelwandler (welchen man sich selber bauen muss, ein MAX202 und ein paar Elkos) oder man besorgt sich eine EIB Datenschnittstelle. Achtung ihr braucht einen echten seriellen Anschluss, die USB-RS232 Adapter klappen nicht! (zumindest nicht die ich versucht hatte: FTDI).
- Falls man alte Geräte erwischt (Spannungsversorgung), braucht man noch eine EIB Datenschiene. Die neuen EIB Geräte brauchen das nicht mehr, sind aber nur ein paar Euro für die Schiene.
- Bei einer alten Spannungsversorgung braucht man noch einen Busverbinder oder man muss halt die Drähte direkt an die Datenschiene löten (der Bus wird an die beiden inneren Bahnen angeschlossen, hier auf jeden Fall dann messen das die Polung passt).
- Ein großes Steckbrett um die Schaltungen aufzubauen (Reichelt: Artikel-Nr.: STECKBOARD 4K7V)
- Draht um das Steckbrett zu verkabeln (ich hab mir hier Reichelt: Artikel-Nr.: H05VK 0,75-10BL gekauft, zugeschnitten und die Enden verzinnt). Ich empfehle mehrere Farben zu verwenden. Hab hier rot, blau, schwarz und grau.
- Programmieradapter für die Atmel Mikrokontroller AVR MK II USB(Reichelt: Artikel-Nr.: AT AVR ISP)
- Bauteile für die Schaltung:
  http://www.reichelt.de/?ACTION=20;AWKID=68984;PROVID=2084
  Ich empfehle hier die Anzahl der Bauteile noch zu erhöhen, vor allem bei den Widerständen, Kondensatoren, Dioden und Transistoren, da diese bei einer höheren Stückzahl günstiger werden und man die sowieso immer wieder mal brauchen kann. Vielleicht will man das noch mitbestellen um seine Bauteile entsprechend aufbewahren zu können:
  Artikel-Nr.: RAACO PSB 4-32

Die ganze EIB Hardware die hier gelistet ist, kann man dann auch gleich im Haus nutzen, also wird nicht umsonst gekauft. Evtl. ist es ratsam, wenn man EIB schon im Einsatz hat erst mal eine Testumgebung aufzubauen. Die aktuelle Software und Schaltungen klappt zwar wunderbar, aber wer will den versehentlich durch einen Kurzschluss etc. gleich was am Lifesystem kaputt machen oder durch eine fehlerhafte Firmware den ganzen Bus lahm legen, weil man bei einer neuen Application die man gerade schreibt noch ein Fehler drin ist.

Falls man an den Schaltungen oder an der Software entwickelt, sind ein Oszi und ein Logic Analyser noch enorm hilfreich, aber nicht unbedingt notwendig!

Hab das folgende im Einsatz und bin damit sehr zufrieden:

Oszi: http://www.conrad.de/goto.php?artikel=121887

LogicAnalyser (LA): http://www.pctestinstruments.com/

Welche Software brauche ich?

Um die Atmel Kontroller zu programmieren kann ich für Windows das AVR Studio 4.0 empfehlen, die Software ist kostenlos, man muss nur seine Adresse angeben (Version 4.14 build 589) und WinAVR in der Version 20071221:

http://atmel.com/dyn/products/tools.asp?family_id=607#798

http://sourceforge.net/project/showfiles.php?group_id=68108

Ich persönlich schreibe meinen Code mit emacs, aber das ist einfach Geschmackssache :)

Das kompilieren, simulieren und downloaden auf den Mikrokontroller mache ich dann mit dem AVR Studio.

Wir dokumentieren unseren Quellcode mit doxygen, welches gebraucht wird um sich die HTML Dokumentation zu erstellen, wenn braucht auch noch GraphViz:

http://www.stack.nl/~dimitri/doxygen/download.html#latestsrc

http://www.graphviz.org/Download_windows.php

Für die EIB Programmierung brauchen wir noch die Software ETS. Wir arbeiten zur Zeit an einer eigenen Software, die hoffentlich bald die ETS ersetzen kann.

http://www.knx.org/de/downloads-support/downloads/

Um an den Schaltplänen zu arbeiten wird Eagle benötigt, das gibt es kostenlos unter:

http://cadsoft.de/download.htm

Nach dem Start als Freeware verwenden auswählen.

Ich hab keine Ahnung, wo fange ich am besten an?

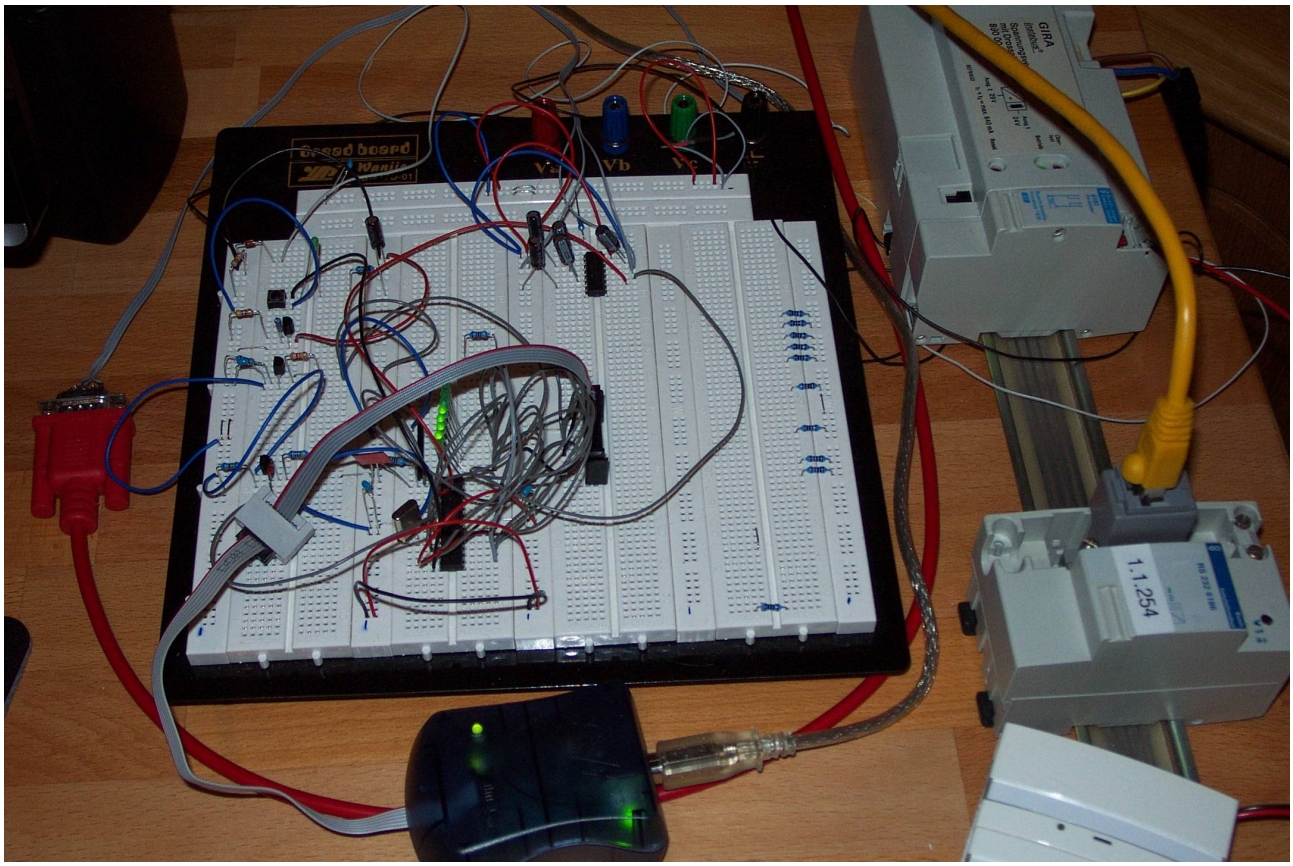
Um in die Atmel Programmierung einzusteigen, sei auf die folgende Seite verwiesen:

http://www.mikrocontroller.net/articles/AVR-GCC-Tutorial

Am besten man arbeitet sich hier langsam durch das Tutorial durch bevor man unsere Software anpackt, denn unsere Software ist mittlerweile recht komplex geworden und ohne Grundkenntnisse am AVR wird das sehr schwer zu verstehen.

Wir helfen euch gerne weiter, schaut einfach im IRC vorbei.

Hier zum Schluss noch ein Bild vom Steckbrett Aufbau mit der entsprechend Hardware außen herum:



Hier sieht man rechts oben das Netzteil mit integrierter Drossel auf der Datenschiene, direkt am Netzteil die Anschlussklemme und ein bisschen weiter unten die Datenschnittstelle mit RS232 Anschluss. Rechts unten im Eck der Gira EIB Schalter.

Unten in der Mitte der Atmel Programmer. Auf dem Steckbrett sind zwei Schaltungen aufgebaut, oben in der Mitte ein RS232 Pegelwandler damit man auf der seriellen Konsole Debugging Meldungen sehen kann und unsere Grundschaltung basierend auf dem ATMega8L.

Viel Spass beim einsteigen,

Matthias Fechner (Idefix)