

embedded-projects.net

JOURNAL

OPEN SOURCE SOFT-AND HARDWARE PROJECTS

[EDITORIAL]

DOPING FÜR ENTWICKLER

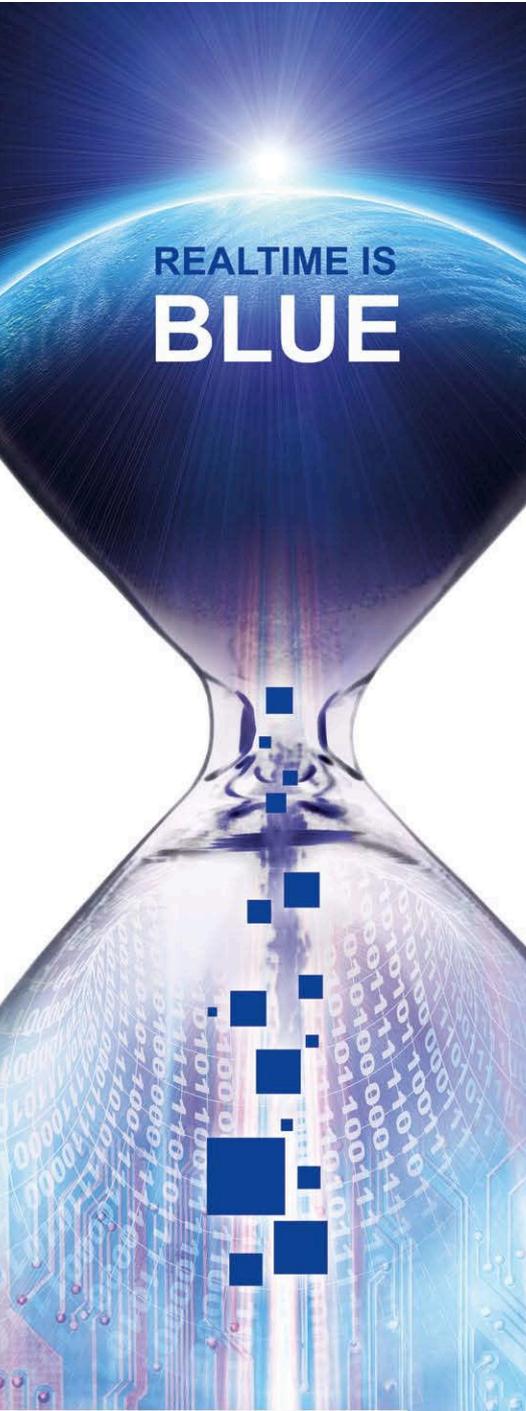
Das Journal zur Vorstellung Deiner Open-Source Projekte. Zum nachbauen und weiterentwickeln oder als Ideengeber zum Start neuer Innovationen.



[PROJECTS]

- OpenStreetMap Navigationsgerät ohne Routenplanung
- Ethersex Hausbus-Netzwerk (Ethernet + IPv6)
- Elektronischer Zauberwürfel

- USB-Stick am Mikrocontroller
- LED Flitzer
- ACF Mikrocontrollerkurs Status



WELCOME

Ausgabe 4 von Embedded-Projects Journal

Wie der Zufall die Freiheit will, braucht die Freiheit den Zufall

Zufall hin oder her, ist der Würfel erst einmal gefallen kann man nur warten bis ein Ergebnis herbei geführt ist.

In Wikipedia, der freien Enzyklopädie findet man Folgendes über den Zufall:

*Beim **Zufall** handelt es sich um den Übergang aus einer Ausgangssituation, die mehrere Endsituationen ermöglicht, in genau eine dieser Endsituationen, wobei zum einen keine erkennbare Ursache für das Zustandekommen dieser einen Endsituation vorliegt und zum anderen bei wiederholtem Vorliegen derselben Ausgangssituation auch die anderen Endsituationen eintreten können.*

Stellen wir uns einmal einen Mann im Alter von 22 Jahren vor. Der Computer ist die Leidenschaft und das Unbekannte eine Herausforderung. Diese Eigenschaft ist jetzt nichts so besonderes. Betrachten wir die Menschen von denen es ca. 6,75 Milliarden gibt, korreliert man diese mit der Alterspyramide basierend auf einer Verteilungsfunktion, die in etwa die Altersstruktur der Weltbevölkerung entspricht, fallen dann in diese Gruppe Menschen im Alter von 22 Jahren etwa 1/60tel. Jene welche dann noch den Computer und die Leidenschaft dafür machen vielleicht ein 1/100tel aus.

Summa Sumarum haben wir ca. 1125000 Kandidaten nach dieser Abschätzung. Linus Torvalds war ebenso einer dieser potentiellen Menschen. Er setzte sich hin um seinen neuen 386er Prozessor kennen zu lernen. Ein gemütlicher Hack ist besser als ein schlechter Film am Abend, rein aus Eigeninteresse. Nachdem ein paar Routinen geschrieben worden sind war klar, es muss ein einfaches Programm zum Lesen der E-Mails auf dem Uni-Server her. Betrachten wir jetzt wiederum wie viele Studenten 1991 E-Mails schon nutzten oder gar von zu Hause aus lesen wollten oder konnten wird es schwer das mathematische Modell weiterzuführen. Naja nehmen wir die maximale Anzahl der Internetanschlüsse 1991 laut [\[www.isc.org\]](http://www.isc.org): Das waren 617000.

Linus merkte schnell, dass sein E-Mail Einwahl-Leseprogramm eine Möglichkeit brauchte, Daten auf der Festplatte abzulegen. So entstand ein einfacher Festplatten- und Dateisystemtreiber. Als Dateiablage diente eine Minixpartition. Betrachtet man jetzt die Verkaufszahlen von Minix bleibt wiederum eine weitere immer kleiner werdende Restmenge von vielleicht einigen 10.000 potentiellen Torvalds übrig.

Linus hatte den Drang nach neuem Wissen und mathematischen Strategien. So wurde die Software nicht nur einfach ein kleines einfaches System, das nach gut dünken hinuntergeschrieben worden ist, sondern Linus beschloss sein System ähnlich einem Unix System nachzubauen. Voilà, Linux 0.01 war entstanden. Für Linus war das zu diesem Zeitpunkt immer noch ein reines Spaßprojekt zur Einwahl in die Uni. Eines Tages war dann der große und glückliche (für Linus im ersten Moment grausame) Zufall gekommen – er startete Linux, tippte auf der Konsole seinen alt bewährten Befehl für die Einwahl über sein Modem in das Internet und gab jedoch als Parameter für das Modem aus Versehen die Festplatte an. Erfolgreich war so die Mini1xpartition durch das eigene Programm demoliert und Minix konnte nicht mehr booten. Erst dieses Ereignis öffnete Linus die Augen und er merkte, er brauchte Minix gar nicht mehr. Wie hoch ist die Wahrscheinlichkeit, solch einen Fehler zu begehen ?

Lotto spielen hat eine Wahrscheinlichkeit von 1 zu 13983816. Was für ein Glück muss es gewesen sein, dass wir heute Linux nutzen können? Wer hätte gedacht, dass wir dank diesem Zufall jetzt solch eine Freiheit genießen können.

In diesem Sinne freue ich mich Ausgabe 4 des Open-Source Projektes Zeitschrift präsentieren zu können.

Benedikt Sauter

sauter@embedded-projects.net

Software- und Echtzeitsysteme

All-In-One-Service für Embedded Projekte

- Softwareentwicklung (Treiber, BSPs, Applikationen)
- Analyse und Design
- Beratung und Schulung
- Hardwareentwicklung
- Vertrieb



Realtime is BLUE

IBV - ECHTZEIT- UND EMBEDDED GMBH & CO. KG

Keltenstraße 2 D-86343 Königsbrunn
Fon +49 (0) 82 31.95 86-041
Fax +49 (0) 82 31.95 86-049

www.ibv-augsburg.net

Anzeige



ARM Boards

ARM USB Debugger (ARM-USB-Tiny)

ARM7/ARM9/Cortex-M3 und XScale Debuginterface für OpenOCD und CrossWorks



ARM USB Debugger
44,90 €
embedded-projects SHOP

AT91SAM7X256 + TFT + Ethernet (SAM7-EX256)

Bestückt mit einem 32 Bit ARM-Mikrocontroller mit 256 kB Flash, 64 kB RAM, 55 MHz, Ethernet 10/100, USB 2.0, RS232, CAN, MMC/SD-Card Slot und TFT-Display



AT91 SAM7X256 TFT+ Ethernet
119,90 €
embedded-projects SHOP

MSP430 Boards

MSP430F1611 Adapterplatine (MSP430-H1611)

MSP430F1611 mit 48K Bytes Programm Flash, 256 Bytes Daten Flash, 10K Bytes RAM



MSP430F1611 Adapterplatine
24,90 €
embedded-projects SHOP

MSP430 USB JTAG Adapter (MSP430-JTAG-TINY)

Programmierung/ Debugging für alle MSP430Fxxx Flash-Microcontroller



MSP430 USB JTAG Adapter
64,90 €
embedded-projects SHOP

AVR Boards

Atmel AVRISP mkII (USB)

Original AVRISP mkII In-System Programmer von Atmel.



Atmel AVRISP mkII
39,90 €
embedded-projects SHOP

AVR Starterkit (inkl. USBprog, Netzteil und ATMega8)

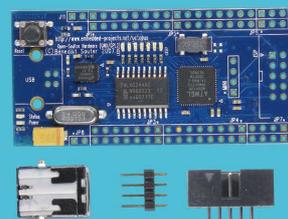
AVR-Starterkit bestehend aus USBprog, AVR-Starterplatine, ATMega8 und Steckernetzteil.



SET AVR Starterkit
61,90 €
embedded-projects SHOP

Octopus USB

Octopus bietet viele bekannte Schnittstellen aus der Mikrocontrollerwelt über ein einfaches USB Gerät an.



Octopus USB
39,00 €
embedded-projects SHOP

PIC Boards

PIC Entwicklungsplatine (PIC-P28-USB)

PIC Mikrocontroller Entwicklungsboard für 28-polige ICs + USB RS232.



PIC Entwicklungsplatine (PIC-P28-USB)
24,90 €
embedded-projects SHOP

AVR32 Boards

ATNGW100 Network Gateway Kit



ATNGW100 Network Gateway Kit
89,90 €
embedded-projects SHOP

Grasshopper AVR32-Board *Open-Source-Version (ohne CD+Kabel)

freie Plattform für die AVR32 Entwicklung



Grasshopper AVR32-Board*
ab 85,00 €
embedded-projects SHOP



Jetzt schneller zum Shop:
www.eproo.net

- großes Sortiment an Evaluations- und Testboards
- bekannte Open Source Projekte
- übernommener Artikelbestand von www.mikrocontroller.net
- faire Preise
- Produkte direkt vom Hersteller
- bequeme Zahlungsabwicklung und schneller Versand



Jetzt Neu:
Versand weltweit

3,95
Euro

OpenStreetMap for Primer 2

Contest project

www.amatronica.de

Einleitung

Einen GPS-Receiver an einen Mikrocontroller anzubinden ist eine weniger schwierige Aufgabe. Der Receiver sendet einen seriellen Datenstrom, in dem die Positionsdaten meist als ASCII-Text enthalten sind. (NMEA-Format) Mit einigen Stringroutinen können der Längen- und der Breitengrad aus dem Datenstrom extrahiert werden. Zwecks späterer Verarbeitung lassen sich die Daten auf einem Speichermedium sichern. (Tracking-Modus)

Möchte man jedoch die aktuelle Position auf einer Karte visualisieren stößt man gleich auf mehrere Probleme:

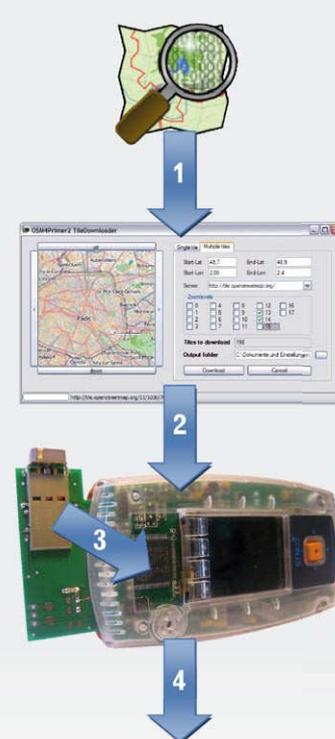
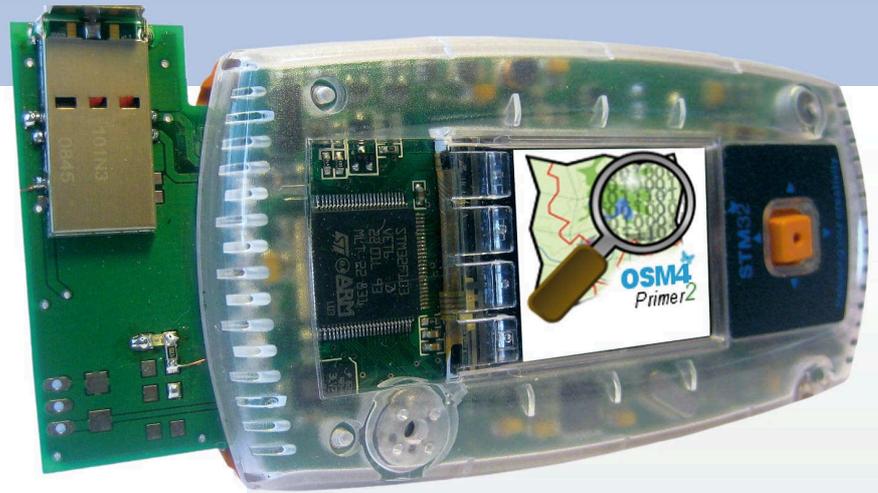
- Geografische Karten sind meist sehr speicherintensiv. Mehrere hundert Megabyte sind, je nach Kartengröße, keine Seltenheit. Hier ist ein Massenspeichermedium gefragt.
- Die von dem GPS-Receiver gelieferten Positionsdaten müssen mit den Kartendaten korreliert werden. Die Berechnungen bedienen sich ausgiebig der Winkelfunktionen und erfordern Rechenpower.
- Die Kartendaten, die im Internet erhältlich sind, unterliegen überwiegend einer restriktiven Lizenz. Das bedeutet, dass die Benutzung kostenlos ist, aber das Kopieren ist nicht ohne weiteres möglich. (Bsp Google-Maps)

OpenStreetMap4Primer2

Das Projekt OpenStreetMap4Primer2 überwindet diese Probleme:

- Die vorhandene Infrastruktur des Primer2 (LCD, Joystick, Mikro-SD-Karten Steckplatz und USB) wird verwendet um die Kartendaten auf das Gerät zu laden, das Gerät komfortabel zu bedienen und die aktuelle Position auf der Karte anzuzeigen.
- Der STM32 Mikrocontroller bietet genügend Rechenpower um die Korrelation GPS-Position zu Kartendaten ausreichend schnell auszuführen.
- Die Kartendaten werden von dem Projekt OpenStreetMap bezogen, aufbereitet und auf die Mikro-SD-Karte des Primer2 überspielt. Die Kartendaten unterliegen der Creative Commons License und sind somit frei zu verwenden. [7]

OSM4Primer berechnet den Kartenausschnitt anhand der vom GPS-Receiver empfangenen Daten „on the fly“ und zeigt diesen auf dem Display an. [**Abbildung 1**]



Funktionsprinzip OpenStreetMap.

1. Kartendaten downloaden und konvertieren.
2. Über USB auf der Mikro-SDKarte im Primer2 speichern.
3. GPS-Receiver liefert Positionsdaten.
4. Kartendaten und Positionsdaten korrelieren und anzeigen.



Abb. 1

Hardware

Primer2

Das Projekt OpenStreetMap4Primer2 setzt auf dem Entwicklungskit Primer2 der Firma Raissonance auf. (siehe Link)
Rund um einen STM32 Mikrocontroller mit Cortex-M3 Kern sind folgende Komponenten angebracht:

- **STM32F103E** (512 Kbytes Flash)
- **Li-Ion-Batterie** mit Ladecontroller
- **Touchscreen** TFT display mit 128x160 Pixel Auflösung
- **Micro SD card connector**
- **Joystick** mit Pushbutton
- 4 zusätzliche **Pushbuttons als Softkeys** nutzbar
- **IrDA** transceiver
- **Kopfhöreranschluß** und integriertes **Mikrofon**
- **MEMs Beschleunigungssensor** zum Messen von 3D-Positionen
- **Mini USB** Schnittstelle
- 20-pin **Extensionboard-connector** mit Zugriff auf **SPI, I2C, USART, CAN** und analog/digital I/Os

Basierend auf dieser Hardware ist ein Betriebssystem CircleOS für den Primer2 erhältlich, welches die Hardware des Primer2 unterstützt. CircleOS ist frei erhältlich.

GPS Receiver

Ausgewählt wurde der GPS-Receiver von Aarlogic [8], da hier die Antenne senkrecht zur Leiterplattenoberfläche angebracht ist. So zeigt die Antenne bei Montage des Gerätes in einer Saugnapfhalterung an der Autofrontscheibe in den Himmel. Dadurch wird die beste Empfangsposition gewährleistet.

Technische Daten:

Chipsatz: Sirfstar III
Startup-time: Hot<2s, Cold<35s (typ.)
Update-rate: 1 s
Channels: 20
Stromaufn.: ca. 30-50mA
Abmaße: 11 x 11 x 30 mm

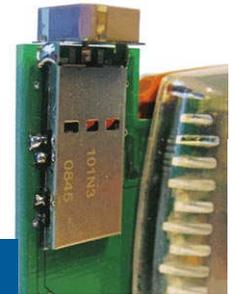
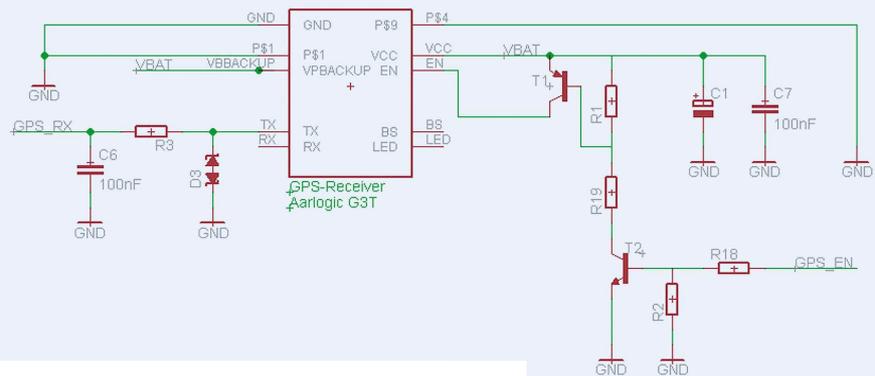


Abb. 2

Abb. 3:

Die Schaltung
Sie ist auf das Nötigste beschränkt. Es gibt nur eine Handvoll Bauelemente rund um den GPS-Receiver.



OpenStreetMap Projekt

Das OpenStreetMap hat sich zum Ziel gesetzt eine freie Weltkarte zu erschaffen. Die User sammeln weltweit Daten über Strassen, Eisenbahnen, Flüsse, Wälder, Häuser und alles andere, was gemeinhin auf Karten zu sehen ist. Weil die Daten von den Usern selbst erhoben werden, haben die User selbst auch alle Rechte daran. Die OpenStreetMap-Daten darf jeder lizenzkostenfrei einsetzen und beliebig weiterverarbeiten. [8]

Organisation der geografischen Daten

Die grundlegenden geografischen Daten sind in einer OSM Datenbank organisiert. Diese besteht aus vielen Dateien im XML-Format, die die geografischen Punkte beschreiben.

Abb. 4: Bestücktes Extensionboard imPrimer2.

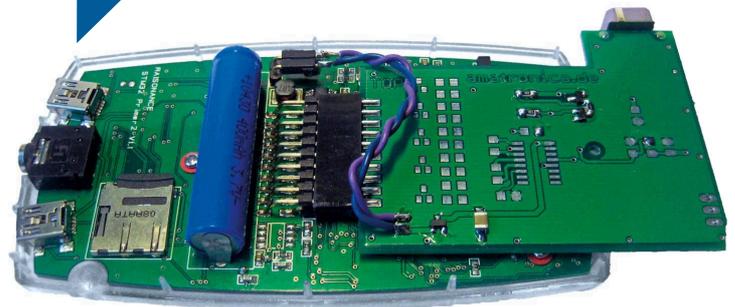


Abb. 5: Auszug aus einer OSM-Datei von Berlin (Brandenburger Tor)

```
<node id="29040842" timestamp="2009-03-05T19:32:18Z" user="anbr"
lat="52.5162835" lon="13.3777436">
  <tag k="name" v="Brandenburger Tor"/>
  <tag k="created_by" v="Potlatch 0.10f"/>
  <tag k="tourism" v="attraction"/>
  <tag k="historic" v="monument"/>
</node>
```

Um diese Daten als Karte auf einem Bildschirm anzeigen zu können müssen sie in ein Bild umgewandelt werden. Dies bewerkstelligt ein Renderer. Ein Renderer erzeugt aus den XML-Daten sogenannte Tiles (Kacheln). Jedes Tile ist ein Bild im PNG-Format mit einer Abmessung von 256x256 Pixeln. Tiles werden in verschiedenen Zoomstufen gerendert. Der Bereich geht von Zoomstufe 1, die die gesamte Weltkarte zeigt, bis zur Zoomstufe 18 mit einer Auflösung von wenigen Metern. Alle Tiles der jeweiligen Zoomstufe bilden zusammen die Weltkarte. (link)
Die Tiles können von verschiedenen Servern heruntergeladen werden. Jeder Server unterscheidet sich in der Art der Darstellung der Tiles und der angezeigten Daten. (Bsp. Mapnik, OSM-Renderer)

Aufbereitung der geografischen Daten

Damit die Tiles auf dem LCD des Primer2 dargestellt werden können, werden sie in das PRM-Bildformat gewandelt. Die PC-Applikation OSM4Primer2 Tiledownloader vereinfacht den Vorgang des Herunterladens der Tiles und das Konvertieren in das PRM-Format enorm.

Vorgehensweise beim Speichern von Kartendaten auf dem Primer2:

Möchte man einen Kartenausschnitt z.B. von Paris für den Primer2 aufbereiten, muss man zunächst die Koordinaten der Ränder des Kartenausschnittes herausfinden. InformationFreeWay oder GoogleEarth zeigen die Koordinaten an. [Abb. 6]
Das Programm OSM4Primer_TileDownloader starten und die gewünschten Kartendaten eintragen "Download Tiles" startet den Download. OSM4Primer_TileDownloader lädt die Tiles herunter und konvertiert diese dann in das PRM-Format. Abhängig von der Downloadbandbreite und dem gewählten Zoomlevel kann dies einige Zeit in Anspruch nehmen
Nach dem Download befinden sich die Kartendaten in Verzeichnissen organisiert. (entsprechend OpenStreetMap tilenames)

Die Verzeichnisstruktur auf der Mikro-SD-Karte sieht folgendermaßen aus und darf nicht verändert werden: [Abb. 7]

Wie konvertiert man Trackdaten in OpenStreetMap Tracks

Wenn man Trackdaten aufgezeichnet hat wurden diese Daten in eine Datei mit dem Dateinamen xxxxx.trk (xxxxx besteht aus einer Pseudo-Zufallszahl). Diese Datei muss noch in eine GPX-Datei umgewandelt werden. Eine GPX-Datei kann dann zu OpenStreetMap hochgeladen

Abbildung 6

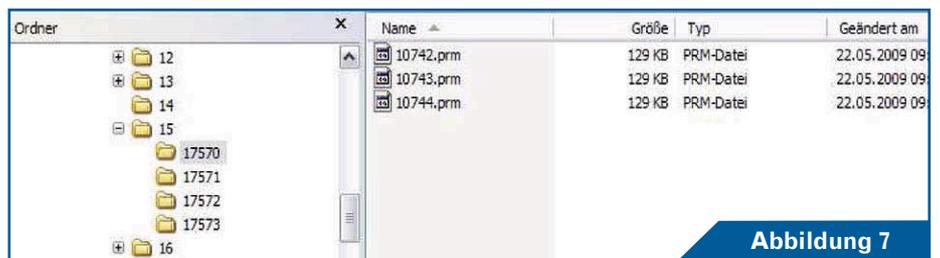
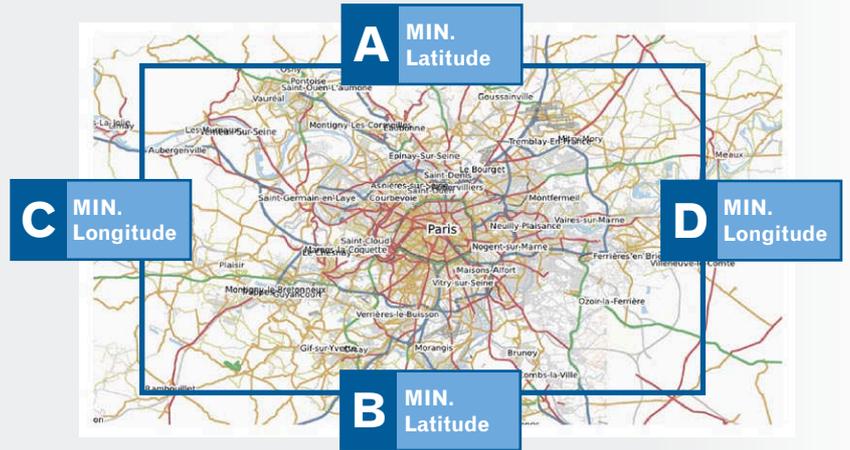


Abbildung 7

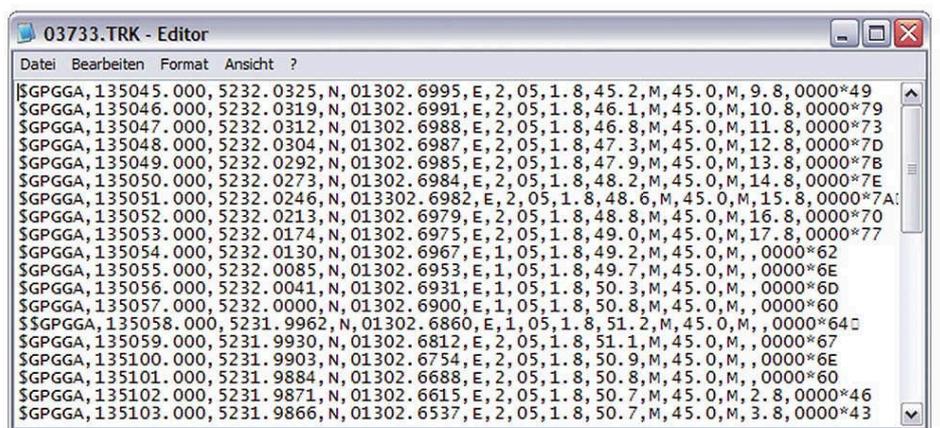


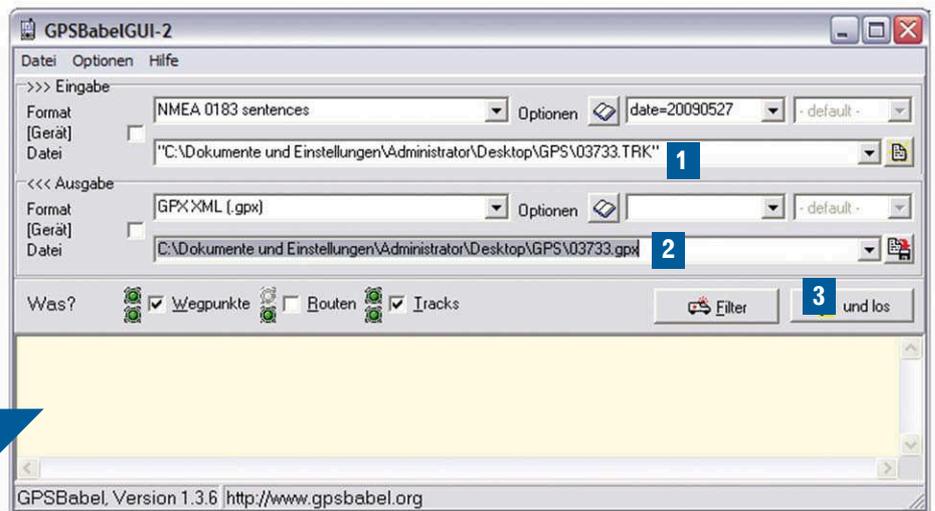
Abb. 8: Eine typische xxxxx.trk Datei

werden und man kann dann Karten online editieren. Die Daten können mit dem Tool GPSBabel [3] leicht in verschiedene Formate gewandelt werden. [Abb. 8]

GPSBabelGUI.exe starten, Trk und gpx Datei auswählen und Konvertierung starten. [Abb. 9]

Abbildung 9:

1. Trk Datei
2. gpx Datei
3. Start



Berechnung der Karten und Darstellung auf dem Primer2

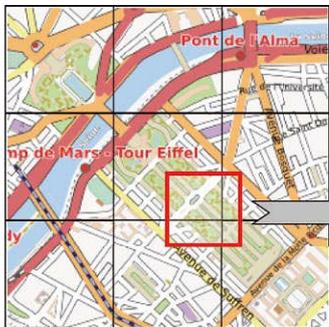


Abb. 10: Virtual map as seen on lcd

OSM4Primer2 berechnet die Längen- und Breitengrade aus dem Datenstream vom GPS-Receiver. In einem NMEA Datensatz sind der Längen- und Breitengrad in Grad angegeben. OSM4Primer2 errechnet die OSM-Tiles, die zu den Positionsdaten korrelieren. Um die aktuelle Position in der Mitte des Bildschirms zu halten ist es mitunter notwendig die darzustellende Karte aus bis zu vier Tiles zusammenzustellen.

Die rote Linie im linken Bild der **Abbildung 10** zeigt den Kartenausschnitt, der auf dem Display angezeigt wird.

Anzeige

GEDACHT. GETAN.
Lösungen für Ihre gute Idee.

quategra

Entwicklung – Embedded Systeme – Training

Quategra GmbH - Karl-Heine-Str. 99 - 04229 Leipzig - Telefon +49 341 49 12 335

www.quategra.de/trainings

Software

CircleOS application

OSM4Primer2 basiert auf einer Statemachine. Ihre Hauptaufgabe ist es GPS-Daten von der Usart (GPS-Receiver) entgegenzunehmen, die GPS-Position zu extrahieren und diese mit den OpenStreetMap Tiles zu korrelieren. Teile der OpenStreetMap Tiles werden zu einer virtuellen Karte zusammengestellt und auf dem Display angezeigt.

Cursor

Die aktuelle Position wird mit einem Fadenkreuz angezeigt. The actual position is shown by a crosshair-cursor. Ausser der Anzeige der aktuellen Position hat der Cursor noch eine zweite Funktion.

 Schlechte Empfangsqualität. Wird beim Systemstart oder innerhalb eines Gebäudes gezeigt. Der angezeigten Position kann nicht vertraut werden.

 Guter Empfang.

 Exzellenter Empfang.

 OSM4Primer2 speichert Trackingdaten. Wird nur im Tracking Menu angezeigt.

 Zeigt die Position deines Buddy. (noch nicht implementiert)

Inifile

Beim Programmstart liest OSM4Primer2 die Datei "osm4prm2.ini". Dies eine Textdatei, die folgende Parameter enthalten kann:

- Der Modus von OSM4Primer2
- Längen- und Breitengrad im Demo-Modus
- Zoomlevel beim Programmstart

Beispiel:

```
mode=demo
lat=52.4567
lon=13.7654
zoom=15
```

Abb. 11: OSM4Primer2 Menus

Mainmenu		
	Zoom map in	Zoomt den Kartenausschnitt zum nächsten Zoomlevel.
	Zoom map out	siehe oben
	Next menu	Wechsel in das Systemmenu.
	Exit application	Beendet OSM4Primer2.
Systemmenu		
	Infoscreen	Infoscreen mit GPS-Daten, zeigt detaillierte Infos über die eintreffenden GPS-Daten. Anzahl der Satelliten, Qualität der Daten, Längen- und Breitengrad, Zoomlevel und die Koordinaten der Tiles. Im Debugmode werden die Rohdaten (GPGGA sentence) angezeigt.
	Track recording	Speichert die aktuelle Position, GPS-Daten (Tracks), in einer Datei auf der Mikro-SD- Karte. Es wird immer nur der GPGGA Datensatz gespeichert.
	Buddyfinder	Sende/empfang GPS-Position von deinem Buddy, tauscht die aktuelle Position. (Diese Funktion ist noch nicht implementiert)
	Back	Zurück zum Mainmenu.
Trackrecording menu		
	Start recording	„Aufnahme starten“ speichert den aktuellen GPGGA Datensatz in einer Textfile(*.trk) Es wird eine Datei mit einer Pseudozufallszahl erzeugt. Der Cursor ändert seine Farbe in Rot.
	Stop recording	Die Aufnahme wird beenden.
	Set recording intervall	Einstellen des Aufnahmeintervall. Es kann zwischen manuellem Auslösen und einem 1 Sekunden Intervall gewählt werden.
	Back	Zurück zum Mainmenu.

Abb. 12: OSM4Primer2 unterstützt derzeit folgende Parameter:

mode	„normal“ normale Funktionalität „debug“ Debugging Informationen werden angezeigt. z.B. Rohdaten im Infowindow „demo“ Demomodus. Zeigt den im Inifile angegebenen Längen- und Breitengrad auf der Karte an
lat	Kommazahl. (Bitte Punkt als Separator verwenden)
lon	Kommazahl. (Bitte Punkt als Separator verwenden)
zoom	Integer zwischen 12 und 17.
baud	Stellt die Baudrate des GPS-Receiver ein.. Default: 9600 baud. Dwenn unbekannt nicht ändern.

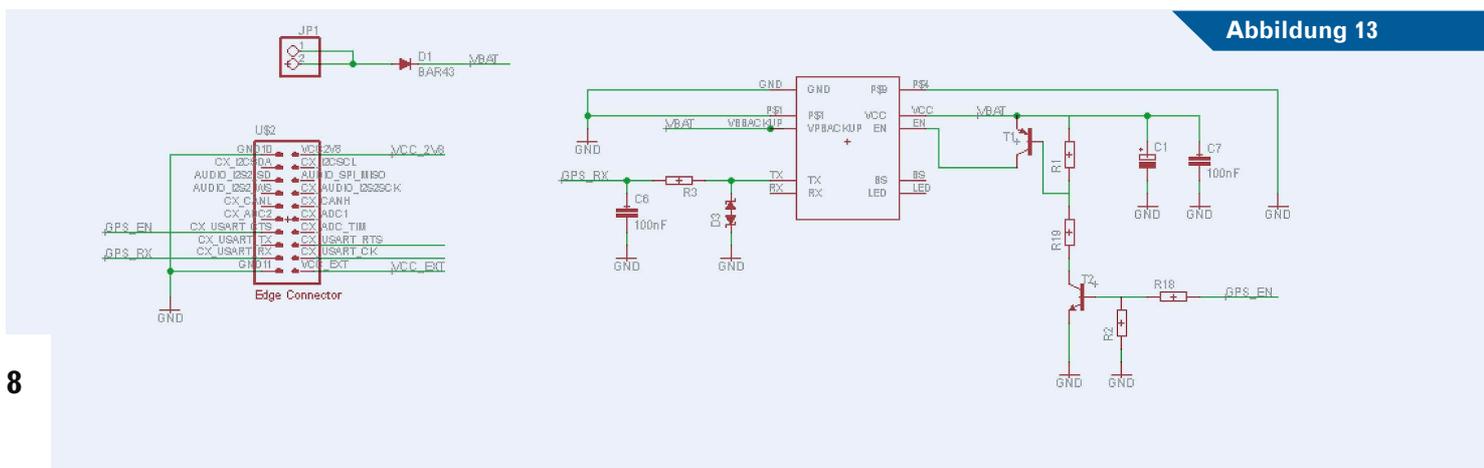


Abbildung 13

Abbildung 14

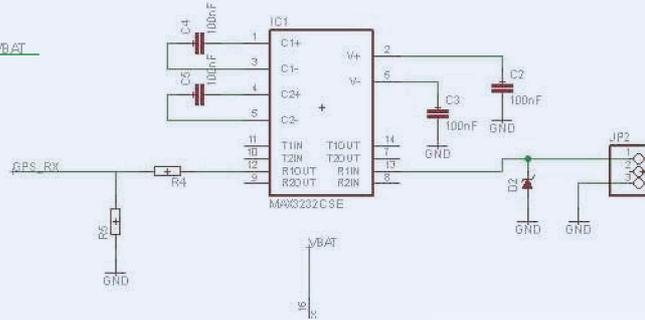
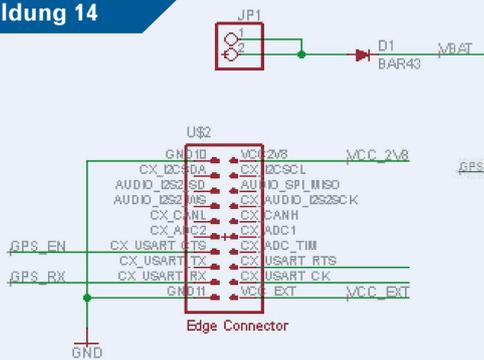


Abb. 15: Verbindungskabel



ACHTUNG:

Sollte der externe GPS-Receiver falsch angeschlossen werden. (Falsche Polarität oder falsche Signalspannungen) kann der Primer2 oder der GPS-Receiver oder beide zerstört werden! Der Autor zeigt sich nicht verantwortlich für Schäden, die aus einem unsachgemäßen Betrieb der Hardware entstehen!

ExtensionBoard

Das Extensionboard verbindet den GPS-Receiver mit dem Primer2 Es gibt zwei Betriebsarten des Extensionboards:

- Es wird der GPS-Receiver von Aarlogic bestückt. [Abb. 13] JP1 muss über ein kurzes Verbindungskabel mit JP3 vom Primer2 verbunden werden. [Abb. 15]
- Es wird ein eigener GPS-Receiver eingesetzt. In diesem Fall kann auf dem Extensionboard der Signalwandler bestückt werden. [Abb. 14]

Fazit

OSM4Primer2 kann kein Navigationsgerät ersetzen, da (noch) keine Routingfunktionalität vorhanden ist. Es wird nur die aktuelle Position auf einer Karte dargestellt. Trotzdem ist die Tracking-Funktionalität eine große Hilfe für OSM-Mappers. Sollte in der Zukunft eine Ethernet-Hardware und ein Ethernet-Stack entstehen, könnte das Gerät die Kartendaten online aus dem Internet beziehen. Ein Austausch der Positionsdaten (Buddy-Funktion) wäre dann ebenfalls leicht zu bewerkstelligen.

ToDo's

1. Einen Algorithmus entwickeln, der PNG-Dateien direkt anzeigen kann. Dadurch entfällt die Konvertierung PNG -> PRM.
2. Eigene Position zu dem Buddy senden. Setzt aber eine geeignete Hardware voraus, mit der Daten ausgetauscht werden können. (Über Internet oder SMS?)
3. Routing Funktionalität implementieren.

Diesen Artikel und weitergehende Informationen und Schaltbilder und Layoutdaten findet man unter der Projekt-Webseite www.amatronica.de

Links

- [1] OpenStreetMap Projekt: <http://wiki.openstreetmap.org/wiki/About>
<http://wiki.openstreetmap.org/wiki/FAQ>
- [2] Tool zum Konvertieren der Tracks: <http://www.gpsbabel.org>
- [3] Tool zum Downloaden von OpenStreetMap-Karten: <http://wiki.openstreetmap.org/index.php/JTileDownloader>
- [4] Tool zum Konvertieren von Längen- und Breitengraden: <http://www.csgnetwork.com/gpscoordconv.html>
- [5] Die Theorie zu NMEA: http://en.wikipedia.org/wiki/NMEA_0183

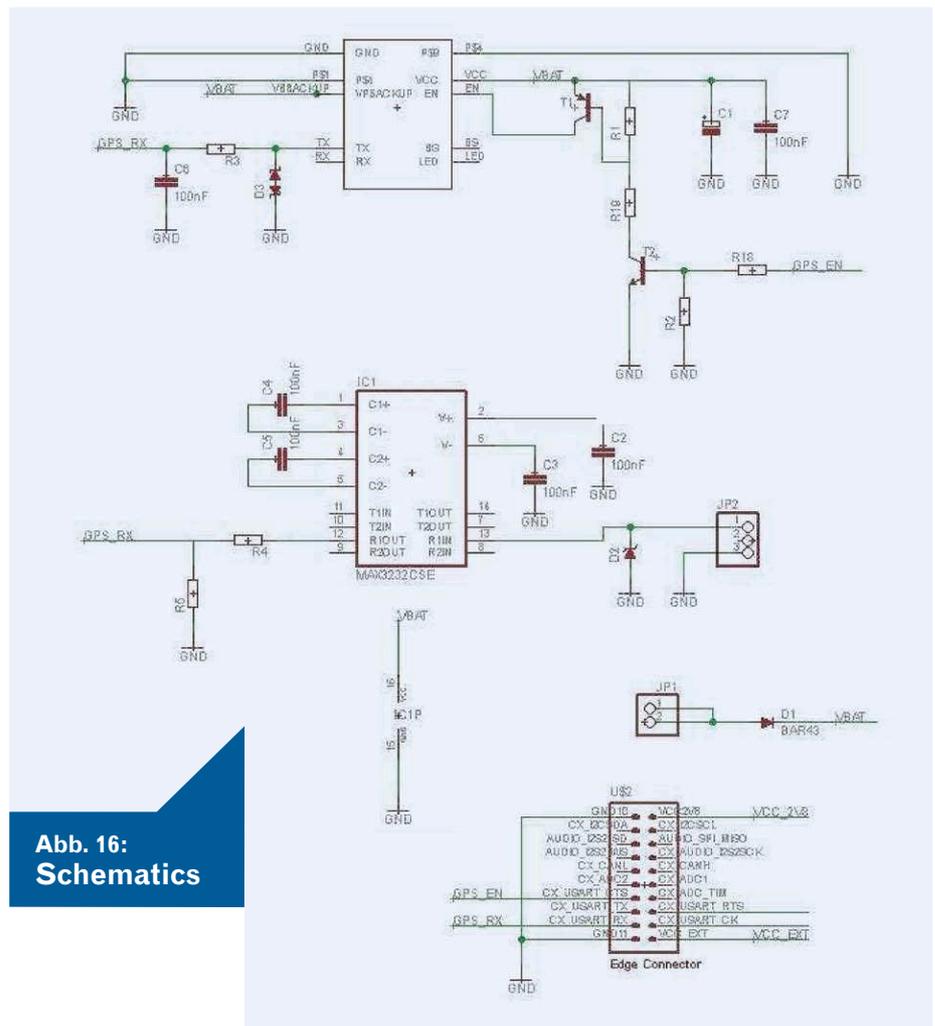


Abb. 16: Schematics

- [6] Bugtracking: <http://www.amatronica.de/mantis>
- [7] Creative Common Attribution-Share Alike license: <http://creativecommons.org/licenses/by-sa/3.0/>
- [8] GPS module: <http://www.roundsolutions.com/gps/gps-module.htm>
- [9] Projekt Webseite: <http://www.amatronica.de>

Ethersex

Softwareplattform für AVR-Firmware-Erstellung

www.zerties.org

Einleitung

Ethersex beansprucht für sich eine der universellsten Softwareplattformen für AVR-Firmware-Erstellung zu sein. Wenn es darum geht das Steuern diverser Aufgaben im Haus, vom Treppenlicht über die Kaffemaschine bis hin zur Heizung zu übernehmen, kommt Ethersex ins Spiel. Die Heizungssteuerung ist eine relativ beliebte Beschäftigung unter Hardwarebastlern. Auch für solche Aufgaben haben sich die AVR-Mikrocontroller der Firma Atmel zum De-facto-Standard entwickelt. Schuld daran ist nicht zu-

letzt die Programmierbarkeit in der Hochsprache C mithilfe der freien Software-Werkzeuge AVR-Libc und GCC, welche für alle Computer-Plattformen verfügbar sind.

Oft ist es wünschenswert Daten über einen Zeitraum aufzuzeichnen, zu visualisieren oder die Hardware von überall steuern zu können. Zum Beispiel ist das Fernschalten und Überwachen der heimischen Warmwasserbereitung/Heizung, von der Arbeit, vom internetfähigen Handy oder Netbook aus, eine sehr bequeme Sache. Aber nicht nur das. Praktisch kann eine komplette Hausautomatisierung mittels Ethersex-basierter Mikrokontrolle realisiert werden.

Für derart anspruchsvolle Anwendungen wird der Mikrocontroller mit einem Ethernet-Anschluss und/oder einem Funk-Modul verheiratet. So kann dieser Minirechner zu einem universellen (Inter-) Netzgerät ausgebaut werden. Mit Ethersex spricht der 8-Bit AVR-Mikrocontroller das Internetprotokoll (IP) via TCP, UDP und sogar über diverse serielle Verbindungen. Ethersex ist also die AVR-basierte Lösung, die Netzwerk und Low-Level Hardware zusammenbringt. Apropos Zusammenbringen: Der Name „Ethersex“ entstand aus der Verbindung von Etherape und IPv6 = Ethersex. Denn Ethersex startete als alternative Firmware für fd0's Etherrape und unterstützt neben IPv4 das Internetprotokoll Version sechs.

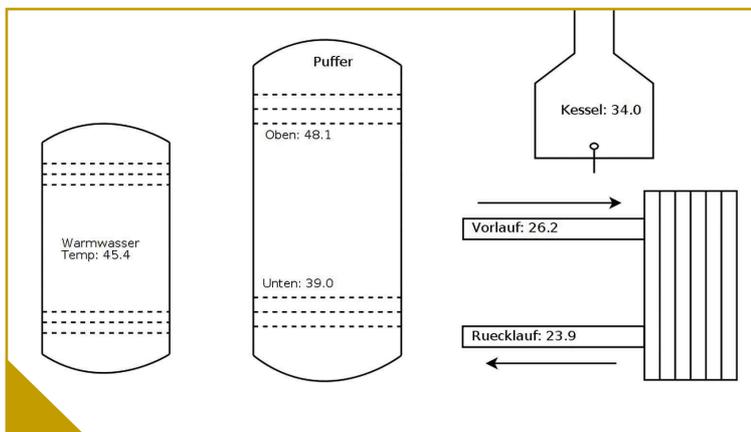


Abbildung 1: Visualisierung von Messwerten

Kerntechnik - Der IP Router

Als erstes gab es die Etherrape [<http://www.lochraster.org/etherrape/>] Hard- und Software von Alexander Neumann, welche den Grundstein für Ethersex darstellt. Der dort unterstützte Enc28j60 bietet ein 10 Mbit Ethernetinterface und wird per SPI an den Mikrocontroller angeschlossen. Der Chip ist eine quasi All-in-One Lösung für die Kommunikation per Ethernet. Man muss nur noch einen Impulsübertrager und eine RJ-45 Buchse verbauen. Um die Integrationsdichte zu erhöhen gibt es auch noch Buchsen mit integriertem Impulsübertrager. Der Kern von Ethersex ist ein deutlich aufgebohrter uIP [http://www.sics.se/~adam/uiip/index.php/Main_Page] -Stack, der neben TCP, UDP und IPv4 auch die neuere Version des Internetprotokolls, IPv6, unterstützt, welches 2^{128} Adressen bietet.

Dies gibt uns die Möglichkeit jeden Controller mit einer weltweit eindeutigen Adresse anzusprechen, ohne sich mit NAT-Routern und ähnlichem zu verrenken. Es steht einem also frei sich von überall in der Welt über die Temperatur seines Warmwassers zu Hause zu informieren und gegebenenfalls die Heizung anzuwerfen.

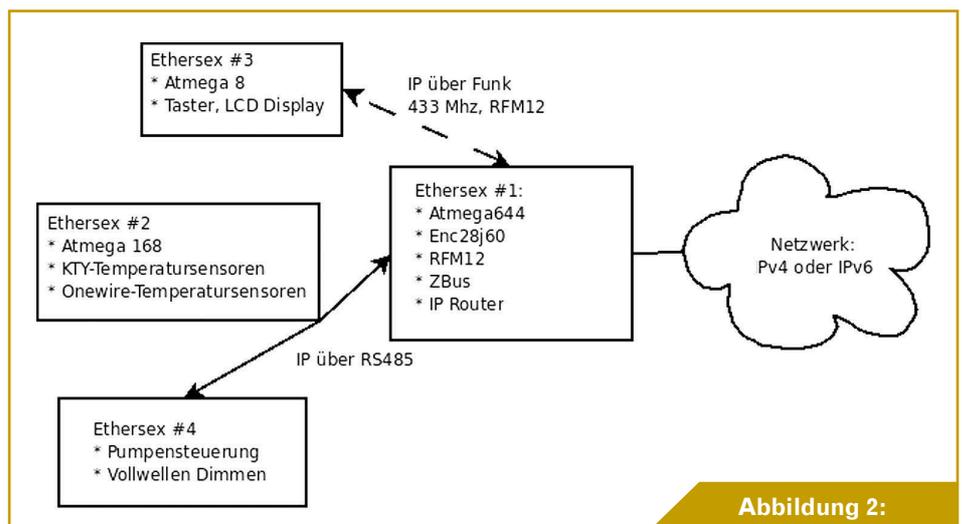


Abbildung 2:

Wie oben erwähnt, unterstützt Ethersex IP nicht nur über Ethernet als Übertragungsmedium, sondern ebenfalls über serielle Leitungen (RS485, ZBus), Funk in den ISM-Bändern (Transceiver-Modul Typ RFM12) und über USB (mit eigenem TUN Device Treiber). Um IP-Pakete über diese Wege zu transportieren, werden die

Pakete in das jeweilige Protokoll eingebettet und als Ganzes übertragen. Es findet also kein Umschreiben der IP-Pakete statt. Dies mag anfänglich als riesiger Ballast erscheinen, allerdings hat man so alle gut erprobten Netzwerktools, vom Sniffer bis zum universellen Netcat zur Hand und braucht nicht seine eigene Werkzeugkiste zu basteln. Dies spart Zeit, vereinfacht die Suche nach Fehlern und reicht für das Steuern und Messdaten sammeln locker aus. Um nun eine Brücke zwischen dem herkömmlichen Ethernet IP-Netzwerk und RFM12 oder ZBus zu schlagen, besitzt Ethersex die Möglichkeit Pakete zwischen diesen Schnittstellen hin und her zu routen. So kann man mit IPv6 selbst einem ATmega8 (8k Flash,

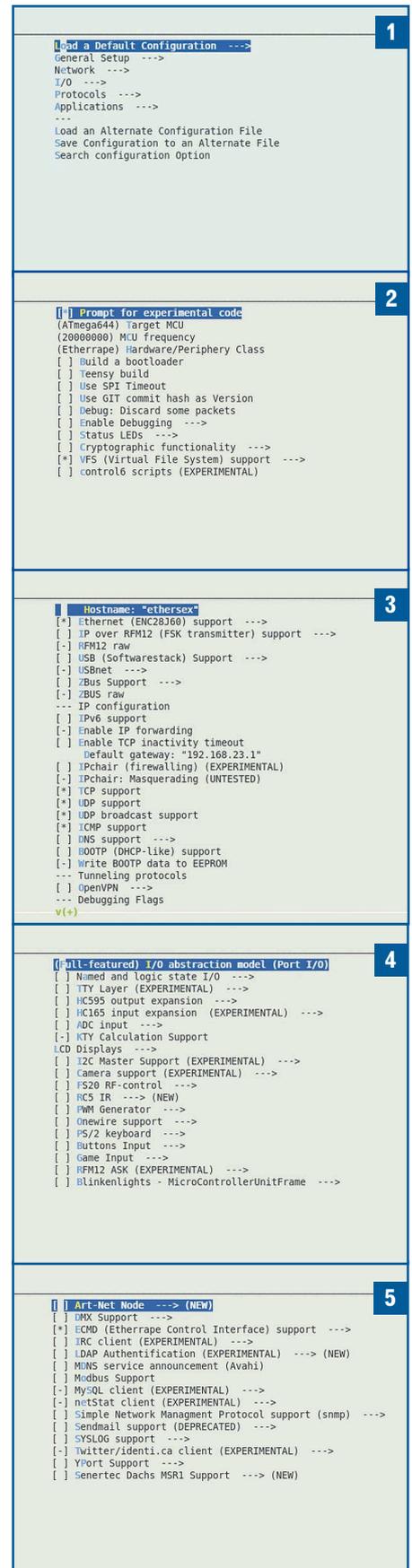
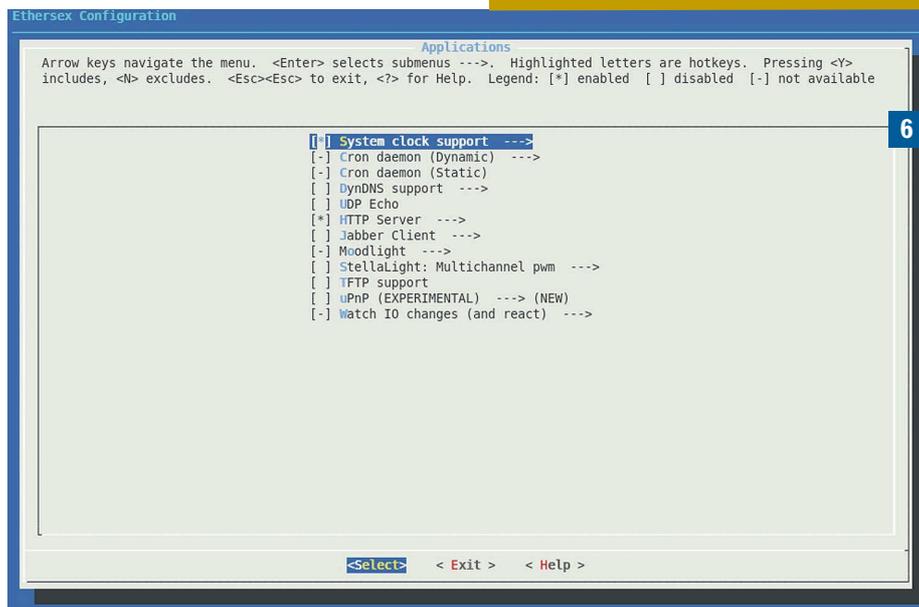
1k RAM) eine weltweit eindeutige IP-Adresse geben. Man verwendet also ein Ethersex, das 2 oder mehr Schnittstellen hat (zum Beispiel Ethernet und RFM12) als Router. Legt man nun die Route für das RFM12 Netz auf das Router-Ethersex, so werden alle Pakete automatisch auf RFM12 und die Antwortpakete zurück geroutet. Möchte man die Verbindung zum Ethersex noch absichern, kann man sich per OpenVPN zum Mikrocontroller verbinden und hat mit ipchair ebenfalls einen Paketfilter zur Hand, der es einem erlaubt zur Kompilierzeit Paketfilterregeln anzulegen und so eine (kleine) Firewall zu realisieren.

282 Optionen - Extrem konfigurierbar

Es gibt in der Konfiguration von Ethersex 282 verschiedene Optionen die man an- und abwählen kann. Daraus und aus der Beschränktheit der Plattform folgt, dass man das Meiste vor dem Kompilieren festlegen und konfigurieren muss. Man aktiviert einfach die Features, die man im aktuellen Fall braucht und lässt den Rest weg. Das kleinste konfigurierbare Ethersex benötigt 400 Bytes an Flashspeicher und besteht nur aus einem Timer, dem sogar jede Netzwerkfähigkeit fehlt. Das größte konfigurierbare Ethersex passt noch nicht einmal in einen ATmega128. Die Vielzahl an Optionen, die sich teilweise gegenseitig ausschließen, werden über eine ncurses Oberfläche, dem sogenannten menuconfig (ja das haben wir vom Linux-Kernel „geklaunt“) konfiguriert. Die meisten der eingebauten Features sind über das ECMD Interface erreichbar. ECMD ist ein text-ba-

siertes Kommandointerface, das über verschiedene Frontends angesprochen werden kann (TCP, UDP, HTTP, Jaber, IRC, I2C, USART). Dadurch, dass ECMD auch für HTTP zur Verfügung steht, hat man Dank JavaScript die Möglichkeit schnell und einfach dynamische HTML-Seiten zu erstellen, die dann den aktuellen Status visualisieren. Die HTML-Seiten werden direkt in die Firmware eingebettet, automatisch. Eine Reihe dieser dynamischen HTML-Seiten steht bereits zur Verfügung, sodass die wichtigen Funktionen von Ethersex auch im Browser angesteuert werden können. Auch können mit Transgraph einfach interaktive SVG-Grafiken erstellt werden, die Javascript enthalten und so automatisch die benötigten Daten nachladen und aktualisieren.

Abb. 4:



Die Möglichkeiten in Ethersex erstrecken sich von seriellen Kommunikationsprotokollen, wie Modbus, über die Ansteuerung bestimmter Webcams (DC3840 und MCA25) bis hin zu Netzwerkprotokollen wie HTTP, SNMP oder MYSQL. Durch die Unterstützung von DCF77 und NTP, sowie der Implementierung eines Cron Daemons lässt sich einfach eine Zeitschaltuhr implementieren, die sogar ohne Uhrenquarz genau genug für Steuerungsaufgaben arbeitet. Eine vollständigere Liste von Features findet sich auf der Ethersex Projektseite [http://ethersex.de/index.php/Feature_Liste].

Abb. 5:

Betriebsstunden HKA	41726 Std.	Wartungsanzeige	erforderlich
Betriebszustand	Abschaltung>4 Minuten	elektr. Leistung	5.294 KW
Vorlauf	55 C	Vorlauf (soll)	31 C
Rücklauf	28 C	Rücklauf (soll)	31 C
Abgas	34 C	Einschalt.soll	25 C
Außentemperatur	23 C	Kühlwasser (Generator)	35 C
Fühler 1	55 C	Fühler 2	114 C
Kühlwasser (Motor)	34 C	Drehzahl	0
KWh elektr.	209694	KWh therm.	398625
aktuelle elek. Leistung	0.000 KW	Starts	0

Metaschichten

Bei der Programmierung von Mikrocontrollern streiten sich im Programmierer immer 2 Seiten: Auf der einen Seite möchte man alles schön abstrahieren um nicht am Ende vor einem riesigen Haufen unwartbarem Quellcode zu stehen, bei dem das Anpassen an neue Features mehr Zeit braucht als die eigentliche Programmierung des Features. Eine Abstraktion bringt auch eine bessere Konfigurierbarkeit mit sich, da nicht alles so starr gelöst ist.

Auf der anderen Seite bringen einem die meisten Abstraktionen einen größeren Verbrauch an Flashspeicher und/oder RAM ein, was man auf so beschränkter Hardware auf keinen Fall haben möchte. So gibt es zwar für den AVR einen C++ Compiler, allerdings verliert man damit wohl zu schnell den Bezug zur Hardware und wundert sich, dass der Flashspeicher schon voll ist.

Ethersex versucht in den meisten Fällen einen Mittelweg zwischen beidem zu finden. Die Dynamik der Abstraktion findet zur Kompilierzeit statt. Es wird also durch den C Präprozessor oder M4 (eine sehr mächtige Makrosprache die einmal den C Präprozessor ersetzen sollte) C-Code erstellt, der anschließend kompiliert wird. So kann man in der Pinning Metaschicht mit *pin (DECKENLICHT, PD4, OUTPUT)* einen Pin definieren, der als Ausgang konfiguriert wird und dann mit *PIN_SET(DECKENLICHT)* angeschaltet werden kann. Will man jetzt sein Deckenlicht auf einen anderen Pin legen, so muss man nur eine Zeile im Pinning ändern, die Firmware neu übersetzen und schon ist alles angepasst. Und das ohne in 10 Dateien nach allen Aufrufen suchen zu müssen!

Mehr Power mit Control6

Da jede Aufgabe andere Anforderungen hat, gibt es mit Control6 eine Art Metasprache, mit der schnell und einfach C-Code entsteht, der neue Funktionen bietet. Der große Charme von Control6 ist die multi-threading Unterstützung und der hohe Abstraktionsgrad. Durch das Multithreading können mehrere Prozesse parallel abgearbeitet werden. Control6 ist genau genommen „nur“ ein M4 Makropaket, aber dennoch sehr flexibel und leistungsfähig. Es ersetzt keine andere Sprache, sondern erweitert eine bestehende: C.

Control6 ist auch sehr gut geeignet, um schnell aus bestehenden Features Neues zusammenzufügen. So ist es ein leichtes die Tastendrucke einer Infrarot RC5 Fernbedienung mit den Hardware-Ausgängen des Mikrocontrollers zu verbinden oder Temperaturwerte von Onewire-Sensoren zum Beispiel auf einem 2x16 Zeichen Text-LC-Display anzuzeigen.

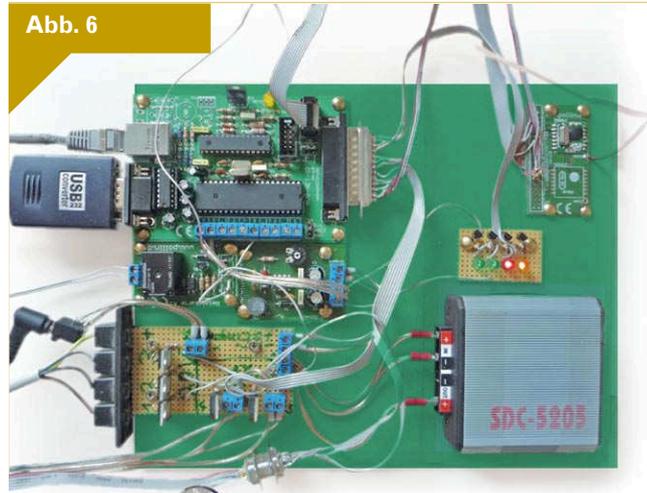


Abb. 6

Verschiedenste Hardware

Wie bereits erwähnt, lässt sich Ethersex für verschiedene AVR Controller kompilieren (vom kleinen ATmega8 bis zum Flaggschiff ATmega128). Der Funktionsumfang ist lediglich von dem im jeweiligen Controller vorhandenen (Flash-)Speicher abhängig. Zudem ist es durch die Pinning-Metaschicht leicht auf neue Ethernet-Boards übertragbar. So werden neben dem Etherape auch verschiedene Bausätze (u.a. AVR Net-IO und Radig AVR-Webmodul) und eigene Basteleien unterstützt. Eine Selbstbaulösung liegt bei ungefähr 20-25 Euro und ist damit nur eine geringe

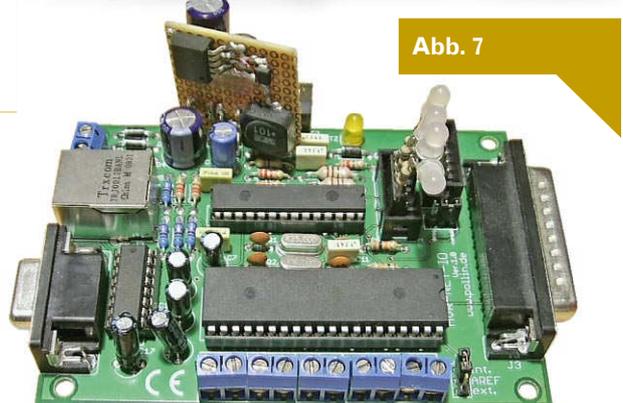


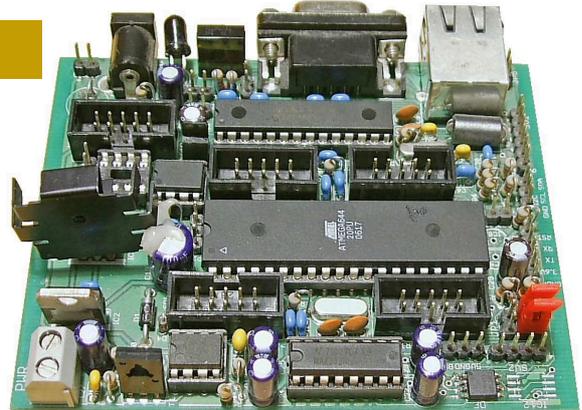
Abb. 7

Einstiegshürde. Mit dem Firmware-BUILDER [<http://www.ethersex.de/firmware-builder/list.cgi>] gibt es die Möglichkeit Ethersex auch ohne großen Aufwand - auf Fertighardware - auszuprobieren. Die Selbstbauvariante kommt mit 3-4 größeren Bauteilen und einer Hand voll Hühnerfutter aus. Das Ethersex-Projekt selbst

bietet keine Hardware oder Schaltungen an, jedoch sehen die meisten Schaltungen mit dem Enc28j60 gleich aus (es wird der SPI-Bus belegt) und können leicht nachgebaut werden (Zu allen Fertiglösungen erhält man die Schaltpläne frei online).

Entstandene Projekte

Abbildung 8



Mit Ethersex als Basis sind bereits einige Selbstbauprojekte entstanden. So wurde eine Heizungssteuerung, welche auch die Pumpen regelt, die drahtlose Steuerung einer Güllepumpe in einer Biogasanlage und die Ansteuerung diverser DMX-Fähiger Leuchtmittel realisiert. Das Auslesen und Visualisieren der Messwerte eines verbreiteten Blockheizkraftwerks gehört ebenso zu den Fähigkeiten wie das direkte Steuern von Funksteckdosen (mit dem RFM12 Modul). Und es ist aktuell eine Infrarotfernbedienung für eine B-77 Bandmaschine in Arbeit.

Mitmachen

Jeder ist herzlich eingeladen bei Ethersex mitzumachen. Die Einstiegshürden finanzieller Art sind aktuell ziemlich geschrumpft. Eine Entwicklungsumgebung ist dank freier Software-Werkzeuge problemlos und schnell eingerichtet. Dafür eignet sich auch eine Ubuntu Live CD, die man beispielsweise in Virtualbox laufen lässt. Das Ethersex-Projekt hat ein öffentliches Git-Repository auf github.com, im Wiki ist vieles gut beschrieben und es wird so gut wie jedes Feature aufgenommen. Vielleicht gelingt es uns ja die

AVR Plattform endlich ganz auszureizen . Und sollte doch mal etwas unklar sein, so sind die Mailingliste, der Bugtracker, der IRC Channel #ethersex auf irc.freenode.net oder der Silc Channel (silc.brokenpipe.de, #cccrbg) gute Anlaufstellen für Mitmacher und Interessierte.

<http://ethersex.de/index.php/Ethersex-Artikel>

Anzeige

Wir suchen dich!

Not.

Studenten der Informatik und E-Technik gesucht.

Net of Trust beschäftigt sich mit Forschung und Entwicklung von Software und Hardware mit dem Schwerpunkt Innovative Technologien für Sicherheit, Forensik und Messtechnik an der Universität der Bundeswehr München.

Werkstudenten, Praxissemester und Diplomarbeiten gesucht!

Wir suchen für unser Team in Augsburg Verstärkung:

Embedded Systeme (AVR, ARM9, AVR32)

GNU/Linux Programmierung (Treiber, Anwendungen)

Schaltungsentwurf / Platinenlayout

Internet Programmierung (PHP, Python, Perl, SQL)

Kryptografie bzw. Mathematik

Bewerbungen und Fragen an:

Net of Trust Solution GmbH
An der Universität der Bundeswehr München
Zweigniederlassung Augsburg

Holzbachstraße 4
86152 Augsburg

Telefon: 0821/27959902
E-Mail: bewerbung@netoftrust.net



der Bundeswehr
Universität  **München**

Elektronischer Zauberwürfel

Projektwettbewerb Mikrocontroller.net

André <and_ref@canathome.de>



Der „Elektronische Zauberwürfel“ ist die elektronische Variante des aus den 80er-Jahren bekannten Rubik's Magic Zauberwürfels. Es entsteht zunächst eine Studie/Prototyp, die zeigen soll, ob die Spielidee des mechanischen Zauberwürfels auf eine elektronische Variante übertragbar ist und durch zusätzliche Features sinnvoll ergänzt werden kann.

Idee und Motivation

Die gängigen Rubiks Zauberwürfel haben 3x3x3 Steine. Neben dem Rubiks Revenge (4x4x4) und dem Rubiks Professor (5x5x5) gibt es keine käuflichen größeren Zauberwürfel - vermutlich aus mechanischen Gründen. Die virtuellen Würfel (PC-Software) lassen sich nicht in die Hand nehmen und auf natürliche Weise begreifen und lösen. Daher dachte ich zunächst einen 8x8x8 Würfel zu bauen, der ähnlich wie ein mechanischer Zauberwürfel lösbar sein sollte (In die Hand nehmen, drehen und die Farben suchen...). Nach kurzer Überlegung, wie viele Lampen und Taster ich dazu brauchen würde, habe ich die Idee gaaanz schnell aufgegeben und mich auf einen 3x3x3er konzentriert. (8x8x8: 512Felder zu je 6LEDs → 3000LEDs)

Nachtrag: 8x8x6(Seiten) zu je 6 LEDs sind „nur“ 2304LEDs!

Der Reiz den mechanischen Zauberwürfel überhaupt lösen zu können, verfliegt nach ein paar Tagen und es stellt sich die Frage: „Wie schnell kann ich den Zauberwürfel lösen?“. Aus dem anfänglichen neugierigen Spiel wird Elektronischer Zauberwürfel eine technische Sportart, die auf folgende Punkte Wert legt:

- zeiteffizientes und/oder zugeffizientes Lösungsverfahren (Lösen mit möglichst wenigen Drehungen)
- definiertes Verdrehen vor dem eigentlichen Lösungsvorgang
- Zeitmessung des Lösungsvorgangs
- Protokollierung des Lernfortschritts (Lösungszeiten aufzeichnen und auswerten)

Der Elektronische Zauberwürfel soll diese Dinge vereinfachen.

Mechanik

Beim elektronischen Zauberwürfel gibt es keine beweglichen Teile. Jede der sechs Seiten des Würfels besteht aus einem feststehenden 3x3-Feld aus LEDs. An allen Kanten sind Taster, die die gedachten beweglichen Teile in der gewünschten Drehrichtung weiterdrehen, d.h. die angezeigten Felder ändern ihre Farben. Übertragen auf einen mechanischen Würfel, würde sich diese Ebene bei einem Tastendruck dann um 90° in die gedrückte Richtung drehen.

Die folgenden Bilder 2-4 zeigen den bereits gefrästen, aber noch nicht verleimten Holzrahmen.

In der Mitte jeder Seite ist das 3x3-LED-Feld. Jedes Feld besteht aus 6 farbigen EinzelLEDs, von denen immer nur eine leuchtet. Neben jeder Spalte und Zeile sitzen Taster. Pro Seite werden somit 12 Taster verbaut.



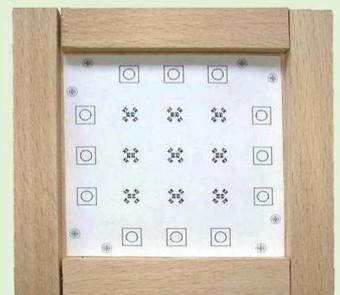
Features

Folgende Funktionen sollen umgesetzt werden:

- **Auto-Scramble** (Verdrehen des Zauberwürfels auf Knopfdruck)
- **Automatische Zeitmessung** (Competition Mode)
 - Die LEDs des verdrehten Würfels werden erst nach Tastendruck freigegeben. Damit beginnt die Einprägungsphase.
 - Nach Ablauf der Einprägungsphase (typisch 15s) beginnt die eigentliche Lösungszeit zu laufen. Dies kann vorgezogen werden, indem schon vor Ablauf der Einprägungsphase eine Taste gedrückt wird.
- Abspeichern der erzielten Zeiten zur **Auswertung am PC**
- mechanisches Drehen entfällt. Dadurch soll sich die Lösungszeit verringern lassen, da keine „Verklemmer“ die Zeit beeinflussen. (Muss noch ausprobiert werden, ob nicht gerade das dann letztendlich den Reiz ausmacht)
- **Tutorialfunktion** zum Erlernen von Zugfolgen, schnelles Rücksetzen auf abgespeicherte Ausgangspositionen (Rückdrehen oder langwieriges Wiederherstellen der Position entfällt)
- **Lösungsvorschläge:** Würfel macht Vorschlag, wie er aus aktueller Situation gelöst werden kann (das dürfte die anspruchvollste Aufgabe sein)

Abb. 2: geplante Würfeloberseite

Damit die Außenkante des Würfels höher ist als die verwendeten Taster (Schurter LSH SMD-Taster; Reichelt: TASTER 9315; Bauhöhe 5mm; Taster sollen nicht überstehen), müssen die Platinen 8mm tief liegen. > Außenprofiltiefe der Kante: 8mm.



Die Platinen sind mit einem umlaufenden Rand von 6mm entworfen. Damit ergibt sich die gewählte Profilform der Kantenteile.

Abb. 3: Profilschnittbogen

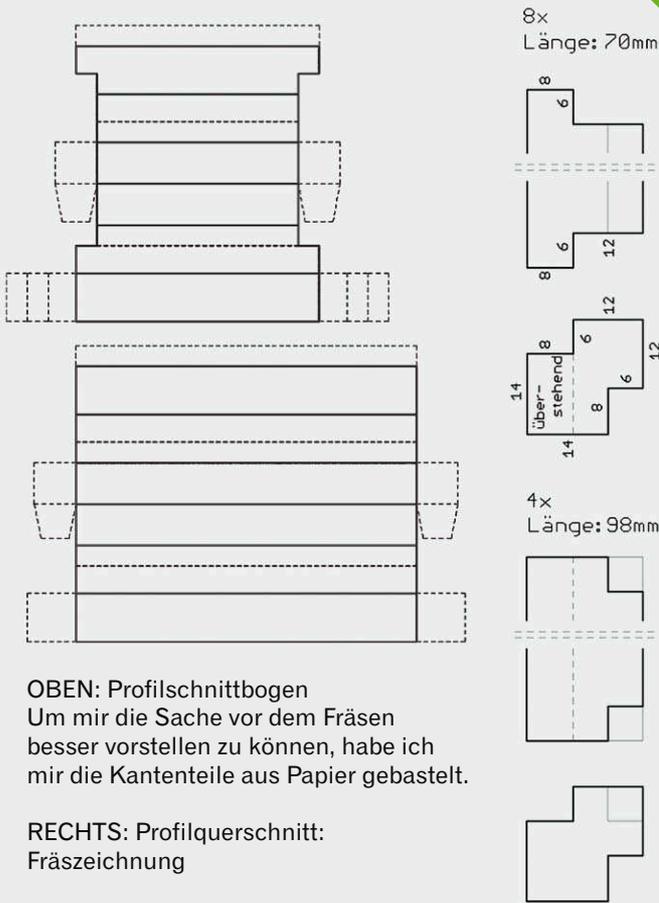
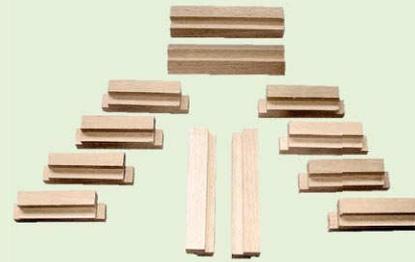
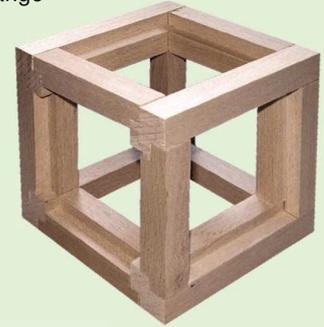


Abb. 4: Holzrahmen

An einem Holzrahmen werden die Seitenplatten befestigt. Kantenlänge des Holzwürfels: 10cm (möglicherweise etwas zu groß?!).

Auf dem rechten Foto fehlen noch die Ausfräsungen der langen Kantenteile, damit die Platine (hier: links vorne) plan aufliegt und keine Aussparungen braucht)



Der Holzrahmen besteht aus zwei verschiedenen Profilteilen:

- 4x Kantenteil -lang- (98mm)
- 8x Kantenteil -kurz- (70mm)

Die Profile der gefrästen Kantenteile haben die Kantenmaße: 20x20mm.



Elektronik

Konzept mit dimmbaren RGB-LEDs (verworfen)

Zunächst habe ich versucht die Anzahl der LEDs gering zu halten, und für jedes Feld eine RGB-TrueColor-LED vorgesehen. Der Vorteil dabei ist, es ist nur eine LED zu bestücken und es müssen nur 3 Ausgangskanäle (statt 6) verwendet werden - aber diese müssen dann auch dimmbar sein. Und da fing das Problem an...

Kurze Überschlagsrechnung für 162 dimmbare Kanäle (= 6 Seiten x 9 Felder x 3 LEDs) über 8Bit-SPI-Schieberegister (andere Ansteuermöglichkeit sehe ich nicht):

- **162 Kanäle** verteilt auf 6 Platinen (nur „ganze“ Schieberegister pro Platine) → 27 Kanäle pro Würfelseite/Platine → 4 Schieberegister pro Seite → Summe: **24 Schieberegister** für den gesamten Würfel (wären theoretisch dann sogar 192 dimmbare Kanäle)

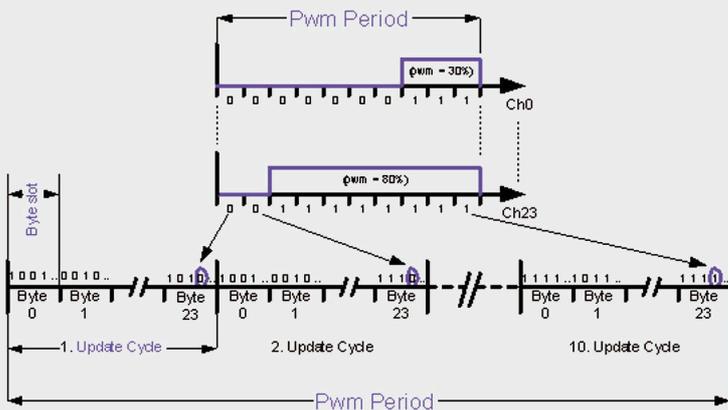
- damit die LEDs nicht flimmern sollten **200Hz Wiederholrate** gewählt werden (= 5ms Periodendauer).

- um vernünftige Farben mischen zu können, müssen diese mit **10% Helligkeitsauflösung** (lineare, zeitliche Auflösung der Periode; Wert 10% experimentell ermittelt) ansteuerbar sein.

- somit muss jede LED alle 500µs aktualisiert werden (5ms/10Steps=500µs). Bei 24 in Reihe geschalteten Schieberegistern, bleibt somit 20.8µs pro Schieberegister Zeit. (das entspricht also auch der Zeit pro rauszuschickendem Byte).

- → SPI-Baudrate: 500µs/24Byte → 2.6µs/Bit
→ **384kBit/s** (das wäre noch problemlos machbar)

Abb. 5: PWM-Konzept zur Ermittlung des zu versendenden Bytes aus der Tabelle der Helligkeitswerte



Ergebnis: D.h. es bleibt dem System genau 20µs, um zu berechnen, welches Byte als nächstes per SPI zu verschicken ist. In dieser Zeit müssen die Helligkeitswerte von 8LEDs in eine binäre Information umgesetzt werden (z.B. LED0: 80%; LED1: 100%...). Die hierfür entwickelte Testroutine (Lookuptable, zur Erhöhung

der Ausgabegeschwindigkeit...) benötigte dafür 45µs auf dem Zielsystem (HCS12C32@50MHz) und ist damit um den Faktor 2.25 zu langsam. Nebenbei sollte ja noch andere Dinge erledigt werden, wie z.B. Züge berechnen usw. Damit ist das Konzept erst mal auf Eis gelegt.

Konzept mit sechs Einzel-LEDs pro Feld

Nachdem das DimmKonzept aus Timinggründen nicht umsetzbar war, plante ich die Umsetzung von 6 Einzel-LEDs pro Feld. Jede Seite hat 9x6 EinzelLEDs -> pro Würfel sind das dann 324 verbauende LEDs, die über Schieberegister angesteuert werden und statisch leuchten sollen.

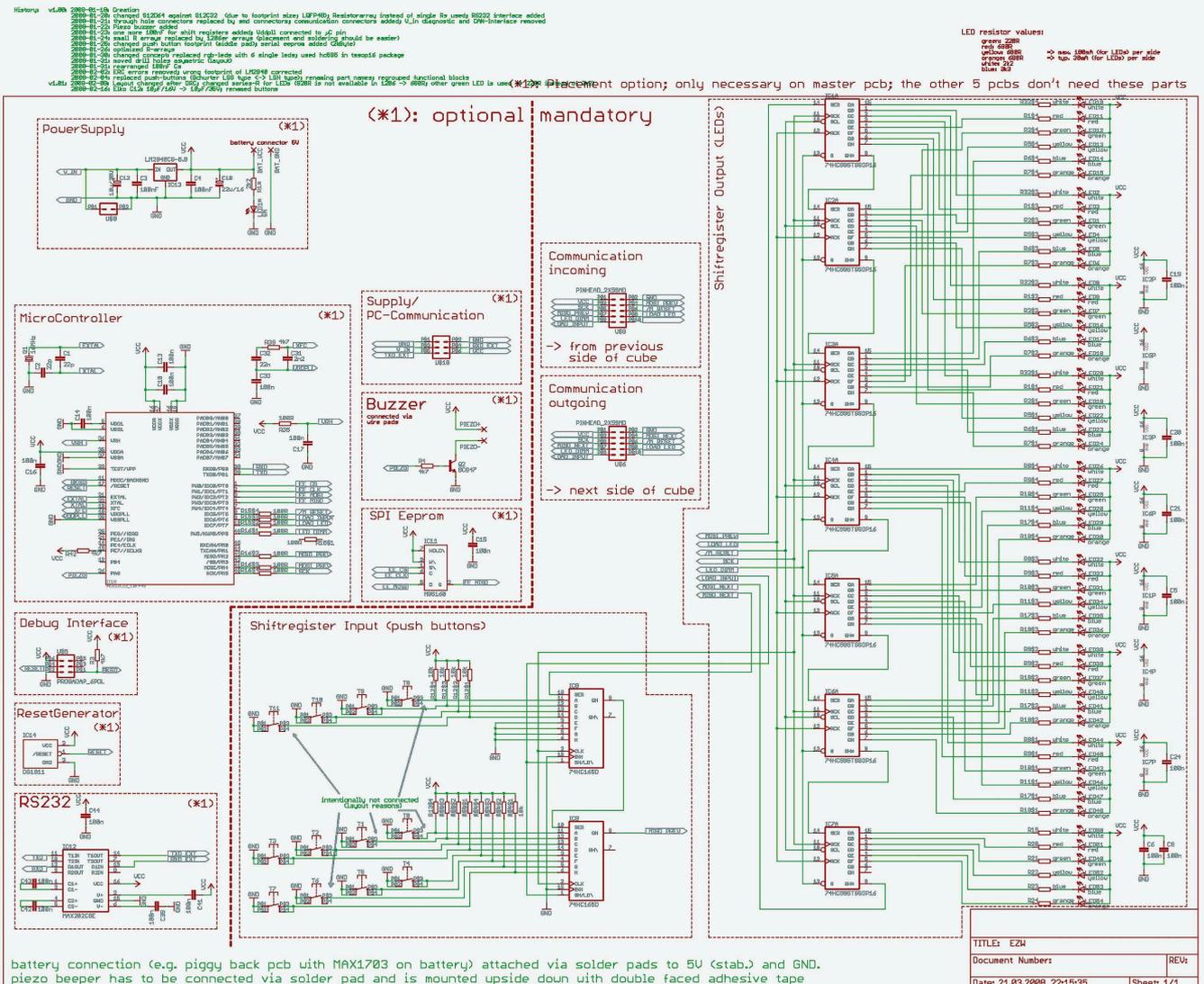
Elektrische Features:

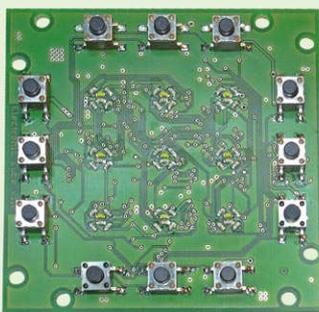
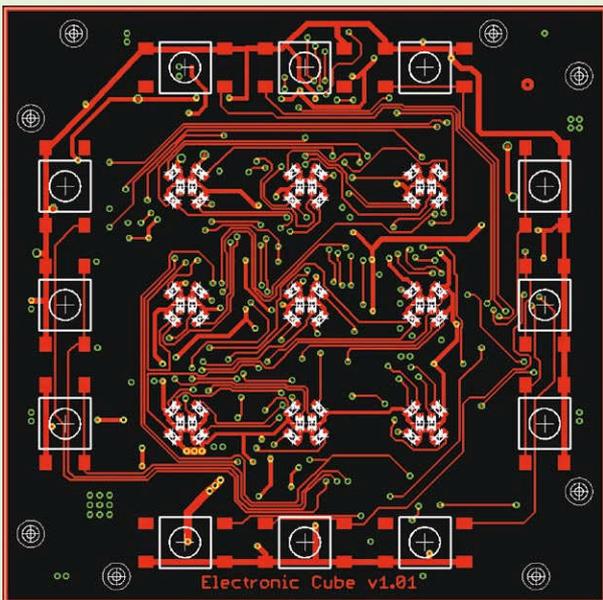
- statische Ansteuerung der LEDs (kein Dimmen); LED Typ: Osram SmartLED 0603, da ausreichende Helligkeit bei kleinem Strom (1..5mA)
- Verwendung von 7 handelsüblichen 8Bit-Schieberegistern HC595 pro Seite (keine separate Treiberstufen oder spezielle LED-Treiberbausteine notwendig); der HC595 erlaubt an seinem EnableEingang durch eine PWM-Ansteuerung auch das Dimmen der angeschlossenen LEDs - allerdings alle auf einmal, so dass hiermit die Gesamthelligkeit des Würfels verringert werden kann.

- die 72 Taster werden ebenfalls über SPI-Schieberegister eingelesen (zeitgleich mit der Ausgabe der LED-Daten); HC165
- Piezopiepser zum Erzeugen von Hinweistönen
- es soll nur 1 Layout für alle 6 Seiten geben
- Kabelverbindung zwischen den 6 Seiten über Verbindung zur „Vorgängerseite“ und „Nachfolgerseite“. Damit ergibt sich eine Kette über die 6 Seiten.
- Stromversorgung über Kabel; Anbindung von Akkus (über Step-up/-down-Konverter) bereits über Löt pads vorgesehen.
- Datenanbindung an PC über RS232-Schnittstelle (Zeitmessung; Darstellung auf PC) Eeprom zum Abspeichern der erzielten Zeiten ausgesuchter Prozessor: HCS12C32 @50MHz, da Gehäuse mit LQFP48relativ klein ist (verglichen mit anderen HCS12-Derivaten) und hierfür bereits eine Entwicklungsumgebung und ein Referenzdesign vorlag; Die Realisierung ist natürlich ebenso mit anderen Controllern möglich.

Schaltplan/Layout & fertige Hardware

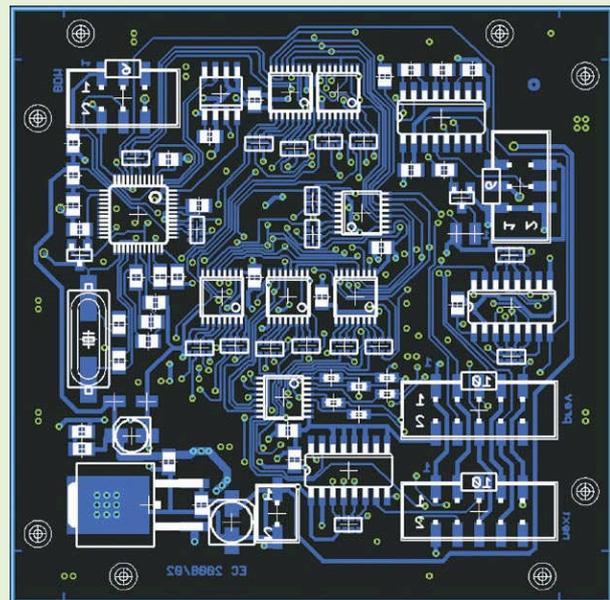
Abb. 6: Schaltplan





Layout - top
Platine: **LED-Seite**

Jede Seite des Würfels besteht aus einer dieser Platinen. Die Außenseite ist mit Tastern und LEDs bestückt...



Layout - bottom
Platine: **Prozessor-Seite**

... auf der Innenseite sitzt die Elektronik. Nur eine der sechs Platinen ist mit einem eigenen Prozessor ausgestattet - alle anderen tragen nur die Schieberegister.

Gesamtansicht
So sieht der Würfel aus, wenn er zusammengebaut ist.



Technische Daten

- **Prozessor:** Freescale HCS12C32 (50MHz); 5V; 32k Flash(12k benutzt); 2k RAM (600Bytes benutzt)
- **Schieberegister:** 42x 74HC595 zur LED-Ansteuerung; 12x74HC165 zum Einlesen der Taster
- **LEDs:** Gesamt: 324 Stück (54Stück von jeder Farbe: weiß, blau, rot, orange, grün, gelb); Gehäuse: 0603
- **Taster:** 72 Stück (12 auf jeder Seite)
- **Gesamtstromaufnahme:** 50mA (LEDs auf 1% gedimmt) 320mA (LEDs auf 100% gedimmt) bei 5V

Ausblick/Verbesserungsideen

Statt der Verwendung von RGB-LEDs könnten auch 6 Einzel-LEDs verbaut werden, die allerdings nicht gedimmt werden, sondern mit verschiedenen zuschaltbaren Vorwiderständen angesteuert werden. Vorteil: LED in einem Gehäuse; kein „Versatz“ bei unterschiedlichen Farben; zu prüfen, ob RGB-LED auch ohne

Streuscheibe verwendet werden kann; sind Farben noch als solche erkennbar oder nur die 3 LED-Grundfarben?

Als Tasteralternative könnten eventuell auch Sensortasten in Frage kommen?! QT1106 (Datenblatt) (3,55€ bei Farnell); evtl. zwischen den LEDs angeordnet...

Mikrocontroller-Software/Firmware

Die Software besteht aus den Modulen/Teilen für:

- RS232-Kommunikation (zum PC)
- SPI-Kommunikation (zu Tastern und LEDs)
- Menü- und Ausgabefunktionen des Würfels (Set/Restore waypoint; History; Mischen; Prüfen ob Würfel gelöst ist; Animation beim Drehen; ShowCube)

- Funktionen zum Drehen des Würfels (L, R, U, D, F, B); „L“ bedeutet z.B., dass die linke Seite des Würfels im Uhrzeigersinn gedreht werden soll;
- Welche Steine werden durch die Drehbewegung an welche Position gebracht? Um den Überblick zu behalten, helfen die unten aufgeführten Einteilungen/Nummerierungen der Steine/Seiten/Stickers.

Zusammengehalten wird die Software von einem Scheduler, der einen 10ms-Task und einen 100ms-Task zyklisch bedient. Der 100ms-Task wird als 500ms-Task benutzt (nur jeden 5. Durchgang ausführen), um folgende Aufgaben abwechselnd auszuführen:

- aktuelle Konstellation (wie ist der Würfel aktuell angeordnet; alle Stickerfarben) zum PC senden
- Zustand aller Taster zum PC senden
- Im CompetitionMode die aktuell laufende Lösungszeit an PC senden (Zeitmessung findet ja im Würfel selbst statt).

Der 10ms-Task erledigt folgende Aufgaben:

- Einlesen und Entprellen der Taster
- „Umordnen“ des Würfels in Folge der erkannten Tastendrucke (das erfordert relativ viel Aufwand, um die ganzen Sticker auf die neue Position zu „kleben“; kann durch Verwendung von Const-Arrays und generischen Funktionen sicher noch deutlich optimiert werden)
- Menübehandlung (Brightness +/-; StartCompetition; Scramble; AnimationMode on/off; Undo/Redo; Set/Restore Waypoint; ResetCube)
- Auswerten der vom PC empfangenen Kommandos (Grundhelligkeit; vom PC ausgelöste Drehbewegung...)
- Animation der Drehbewegung/Drehrichtungsanzeige (abschaltbar; LEDs nacheinander einschalten, um dem Benutzer optisch die Drehrichtung anzuzeigen); hierzu ist hinterlegt, wie und wann jede LED für jede mögliche Bewegung geschaltet werden muss.
- Ausgabe der SPI-Daten zum Schalten der LEDs

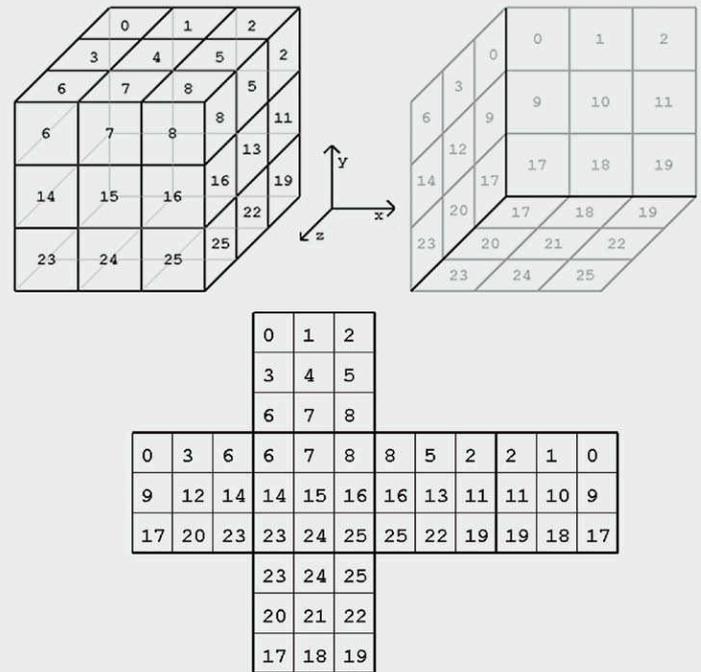
Download der Sourcen [1]

PC-Software

Als zusätzliches Feature war geplant eine Tutoriumsfunktion in die Hardware einzubauen. Nachdem ich mich etwas intensiver mit Konzept/Implementierung beschäftigt hatte, wurde mir schnell klar, dass das aus Ressourcengründen nicht autark in der Würfelektronik abgehandelt werden kann (Laufzeit und/oder ROM/RAM-Bedarf). Darum beschäftige ich mich derzeit mit der Implementierung eines Autosolving-Algorithmus im PC, der aus jeder Situation heraus, den Zug vorschlägt, der einen Schritt in Richtung gelöster Würfel führt.

Es gibt Lösungsalgorithmen, die versuchen mit möglichst wenig Zügen den Würfel zu lösen. Derzeit geht man davon aus, dass jede Würfelpermutation mit max. 20 Zügen gelöst werden kann (bewiesen ist das Maximum mit 23 Zügen). Das ist aber alles sehr theoretisch und die vorgeschlagenen Züge sind vom Menschen unmöglich nachzuvollziehen. (Zunächst kommen einem die ersten 12..15 Züge vollkommen willkürlich vor, da-

Abb. 7: Anordnung der Felder (oben, vorne, rechts); (unten, hinten, links)

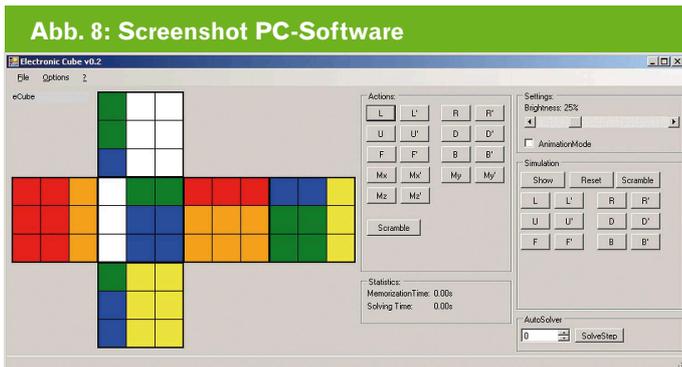


nach „fallen“ dann alle Steine innerhalb weniger Züge in ihre richtige Lage).

Um auch einen praktischen Nutzen aus dem Autosolver zu ziehen, habe ich mich entschieden die „Spiegelmethode“ (erstmals im Spiegel 1981 veröffentlicht) als Tutorium zu implementieren. Beschreibung der Methoden auf Mathematische Basteleien [2] und Kantenkreuz [3]. Oder für diejenigen, die sich das lieber als Youtube-Video [4] reinziehen.

Der Vorteil dieser Methode ist, dass es viele Zwischenstationen gibt, die der Reihe nach zu erreichen sind, und die nur ein paar wenige Züge auseinander liegen. Dabei sind intuitive Züge (die jeder mehr oder weniger intuitiv durchführen kann, ohne genau zu wissen, was er tut) mit „vorgefertigten“ Zugfolgen zu kombinieren. Die vorgefertigten Zugfolgen sollen ja hiermit eingeübt werden. Für einen Autosolver stellen die, für Menschen intuitiven Züge, natürlich ein Problem dar. Irgendwie müssen diese in einen Algorithmus gepackt werden. Der einfachste Algorithmus hierfür ist „God’s algorithm“ [5], der stur alle Möglichkeiten durchprobiert. Bei 10^{19} verschiedenen Permutationen (beim 3x3x3 Würfel) ist dies natürlich in endlicher Rechenzeit nicht möglich. Aber wenn nur 4..6 Züge in die Zukunft gerechnet werden soll, so kann das in <1s errechnet werden (Werte meiner Implementierung). Und das reicht dann auch meist aus, um zur nächsten Zwischenstation zu kommen und hierfür Zugvorschläge zu machen. Um die Zugfolgen dann mechanisch/motorisch einzuüben, wird ein Wegpunkt gesetzt, von dem aus dann die Folge geübt werden kann. Auf Knopfdruck springt dann der Würfel wieder in die abgespeicherte Stellung zurück. Mit dem mechanischen Zauberwürfel ist das „geringfügig“ aufwendiger.

Abb. 8: Screenshot PC-Software



Ausblick/Verbesserungsideen

Zusätzlich zur „Spiegel-Methode“ könnte man auch ein Blind-Folding-Tutorium einbauen. Blind-folding kann ich nämlich selbst (noch?) nicht. Beim Blind-Folding wird der Würfel verdreht, anschliessend darf der Würfel eine beliebig lange Zeit betrachtet werden und muss dann blind gelöst werden. Während des Lösungsvorganges darf der Würfel nicht mehr betrachtet werden (Augenbinde). Die Lösezeit ist die Summe aus Einpräphase und anschliessender Drehzeit.

Probleme bei Planung und Durchführung

Folgende Probleme sind bei der Planung/Erstellung aufgetreten:

- **mechanische Trägerkonstruktion** für Würfel; wie halten die Platinen zusammen?; kann auf ein Gehäuse evtl. verzichtet werden? Wie wird vermieden, dass die überstehenden Taster beim Hinlegen des Würfels alle gedrückt werden? Wie kann der Würfel angefasst werden, ohne dass die Taster versehentlich betätigt werden?
- Liegen die Kanten- und Eck-LEDs **nahe genug beisammen**, so dass man diese auch als solche erkennt?
- **Platzbedarf für Bauteile** ist sehr hoch:
 - Bauteile dürfen nur auf der Unterseite platziert werden
 - Platine muss umlaufenden Rand von 6mm zur Befestigung haben, in dem keine Bauteile (Unterseite) platziert werden dürfen.
- Stromversorgungs- und Kommunikationskabel **stören beim Lösen...** muss in der nächsten Musterphase durch Akkus und Funk (Zigbee/XBee ersetzt werden)
- das räumliche Vorstellungsvermögen wird gut geschult... besonders bei der Umrechnung der PC-internen Datendarstellung (Würfel ist beschrieben durch Position und Orientierung der Steine) in die Aufkleberdarstellung (Farben der einzelnen LEDs).

Fazit

Der größte Nachteil an diesem Prototypen ist die schwierige Erkennung wo die Farben liegen, da diese zu weit im „Innern“ entfernt von den Kanten liegen und somit nicht (ohne längere Übungszeit) auf einen Blick erkannt werden können. Beim mechanischen Zauberwürfel grenzen die farbigen Aufkleber direkt aneinander. Hier ist ein Abstand von 35mm bis zur Würfelkante.

Beim „normalen“ Speedcubing werden Zugfolgen trainiert, die dann nur noch mechanisch/motorisch abgerufen werden, ohne bewusst darüber nachzudenken. Gleiches konnte ich nach einiger Zeit auch in Bezug auf das Drücken der Tastenfolgen beobachten - das geschieht ebenso „natürlich“ wie beim mechanischen Würfel - hätte ich so nicht unbedingt erwartet.

Die Idee mit dem elektronischen Zauberwürfel habe ich zuvor im Internet noch nicht gesehen... naja, also wohl gute Idee gehabt, dachte ich mir. Bis ich zwei Monate nach Fertigstellung des Projekts, dieses Video fand [6] - aber das spielt auch in einer ganz anderen Liga.

Der elektronische Zauberwürfel in Aktion als Youtube-Video [7]

Interessante Details/Links:

Anzahl der verschiedenen Stellungen beim 3x3x3-Würfel: 4.3×10^{19} ; beim 4x4x4-Würfel: 10^{43} ; beim 5x5x5-Würfel: 10^{87} . Der Weltrekord im Lösen des 3x3 Würfels liegt bei knapp unter 9s! [8] Optimal AutoSolver [9]. Virtueller Würfel zum Selbsta ausprobieren (Win-Software) [10].

- [1] http://www.mikrocontroller.net/articles/Elektronischer_Zauberwuerfel
 [2] <http://www.mathematische-basteleien.de/zauberwuerfel.htm>
 [3] <http://www.kantenkreuz.de/>
 [4] <http://www.youtube.com/watch?v=mfvgeOarPBo>
 [5] <http://www.jaapsch.net/puzzles/theory.htm>
 [6] http://www.youtube.com/watch?v=V4A_wfaScy4
 [7] <http://www.youtube.com/watch?v=G6SVoK5OG90>
 [8] <http://www.youtube.com/watch?v=h6GnxKGicyg>
 [9] <http://www.kociemba.org/cube.htm>
 [10] <http://gabbasoft.com/?404=Y>



Persönlich & Online!

► Flexible Leiterplatten ONLINE!

Polyimid 0,06mm bis 0,25mm
 * 1- und 2-Lagen durchkontaktiert * chem. NiAu * Versteifungen * 4mil * Abdeckfolie, Lötstopplack oder Kombination * 3M-Klebefolie * Nutzenfertigung ...

► Starre Leiterplatten bis 8 Lagen online!

FR4 0,50mm bis 2,40mm * 1-bis 8-Lagen Multilayer * chem. Zinn, HAL bleifrei oder chem. NiAu * 35µm oder 70µm Cu * 4mil Strukturen * 0,20mm Bohrungen * Sonderfarben * Viafülldruck * Abziehlack * UL-Kennzeichnung * Sonderkonturen & DK-Slitze inklusive * Nutzensetzung für Sie nach Zeichnung ...

► Mehr Leistung & Spezialfertigung

Bis 18 Lagen * Rogers-HF und Isola Hoch-Tg Material ab Lager * Stecker gold * blind- & buried vias * DK-Z-Achse-Fräsen * Alukern oder -Träger * bis 400µm Dickkupfer * Sonderlacke * Sonderaufbauten * SMD-Schablonen * Großserien über LeitOn Hongkong Ltd. ...

► Ihr Service - Ihre Qualität

ISO 9001:2000 zertifiziert * zuverlässig & termintreu * kompetent und erfahren * persönlicher Service & netter Kontakt * schnellste Bearbeitung Ihrer Anliegen * professioneller CAM Design Rule Check (DRC) ...

www.leiton.de
 kontakt@leiton.de
 +49-(0)30-701 73 49 0



LeitOn GmbH, Gottlieb-Dunkel-Str. 47/48,
 12099 Berlin, www.leiton.de,
 kontakt@leiton.de,
 +49 (0)30 701 73 49 0

USB-Stick am Mikrocontroller

Grundlagen zur Realisierung in eigenen Anwendungen.

Matthias Kahnt <Matthias.Kahnt@eltra-tec.de>

Allgemeine Funktionsbeschreibung

USB-Sticks sind zur Zeit günstig zu bekommen. Daher ist es naheliegend diese auch in eigenen Projekten einzusetzen. Das Projekt demonstriert an einem Hard- und Softwarebeispiel die Verwendung von USB-Sticks an Mikrocontroller-Systemen. Es ist nicht als fertiges Projekt zu betrachten, sondern soll eine Grundlage zur Realisierung eigener Anwendungen sein.

Spezielle Kenntnisse des USB-Busses sind nicht erforderlich. Wer sich dennoch näher damit beschäftigen möchte, dem wird der Artikel „USB-Grundkurs“, der ersten Ausgabe dieses Journals [1] empfohlen.

Ebenfalls unnötig ist die tiefgreifende Einarbeitung zu den Funktionen des FAT-Dateisystems, was einen Vorteil gegenüber der Verwendung von SD-Karten darstellt. Ein

paar Grundkenntnisse dazu dürfte aber jeder PC-Benutzer schon haben.

Zur Realisierung ist ein USB-Hostcontroller erforderlich. Die meisten Mikrocontroller mit integrierten USB-Port, bzw. Lösungen mit FT232-Bausteinen, bieten keine Host-Funktionalität und sind demnach nicht in der Lage, einen USB-Stick anzusprechen. Es wird der Typ VNC1L von FTDI (Vinculum) eingesetzt.

Dieser IC hat folgende Eigenschaften:

- USB 2.0 Host- und Slave-Funktionalität
- 2 USB-Ports mit Low- und Full-Speed
- UART-, SPI- und Parallelinterface
- zusätzliche, frei verwendbare I/O-Ports
- integrierter 8Bit Mikrocontroller und 32Bit Arithmetik-Prozessor
- interner DMA-Controller, 64kByte Flash, 4kByte SRAM
- Firmware für FAT16 und FAT32 Support kostenfrei verfügbar
- 3.3V (25mA) Versorgungsspannung, I/O-Pins sind 5V tolerant
- wenige externe Bauteile erforderlich

Leider ist der IC nur in SMD (LQFP-48 Gehäuse) erhältlich.

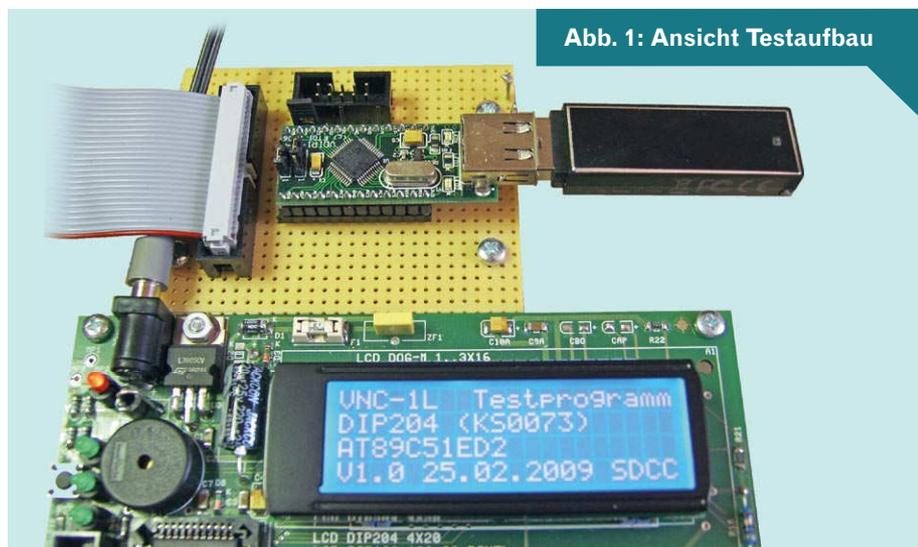


Abb. 1: Ansicht Testaufbau

Hardware

Zum Einsatz kommt ein fertiges Modul VDIP1. Dieses enthält auf einer kleinen Platine einen VNC1L mit allen erforderlichen Zusatzbauteilen. Werksmäßig ist auch die passende Firmware (VDAP) bereits vorprogrammiert. Durch den mechanischen Aufbau, passend für Lochrasterplatinen, ist dieses Modul besonderes für Versuchsaufbauten geeignet.

VDIP1-Modul:

- VNC1L im LQFP-48 Gehäuse
- Quarz 12MHz (per interne PLL auf 48MHz vervielfacht)
- 3.3V Spannungsregler
- 2 Jumper für die Interfaceeinstellung
- USB-A Buchse
- komplette Schutzbeschaltung
- 2 Status-LEDs
- 2x 12 Stifte im DIP24 IC-Format

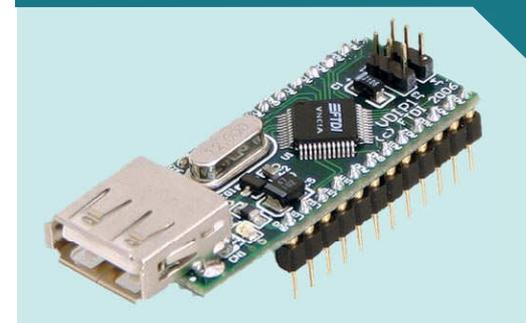
- VDAP Firmware und Bootloader bereits vorinstalliert
- Updatefähig über USB-Stick

Die genaue Schaltung ist unter [2] zu finden. Falls doch der VNC1L IC auf der eigenen Platine zum Einsatz kommt, sollte man sich an dieser Schaltung orientieren. Das Datenblatt dazu ist unter [3] downloadbar.

Das VDIP1-Modul ist für Experimente gut geeignet, zumal es die VDAP-Firmware bereits enthält. Das flashen eines leeren VNC1L ist dagegen nicht gerade einfach! Bezug ist über diverse Elektronikversender möglich. Preis: VDIP1 ca. 25 Euro, VNC1L als einzelner IC ab ca. 10 Euro.

Im Beispiel wurde als Zielsystem eine universelle Mikrocontroller-Platine [4] mit 8051-Mikrocontroller (Atmel

Abb. 2: Ansicht VDIP1-Modul



AT89C51ED2 [5]) verwendet. Es kann natürlich jeder andere Mikrocontroller eingesetzt werden. Besondere Leistung ist nicht erforderlich. Die Taktfrequenz beträgt im Bsp. 11.0592MHz und ist unkritisch. Bei testweiser Erhöhung der Taktfrequenz auf 24MHz bzw. Nutzung des X2-Modes, funktionierte die Software ohne Änderung auch.

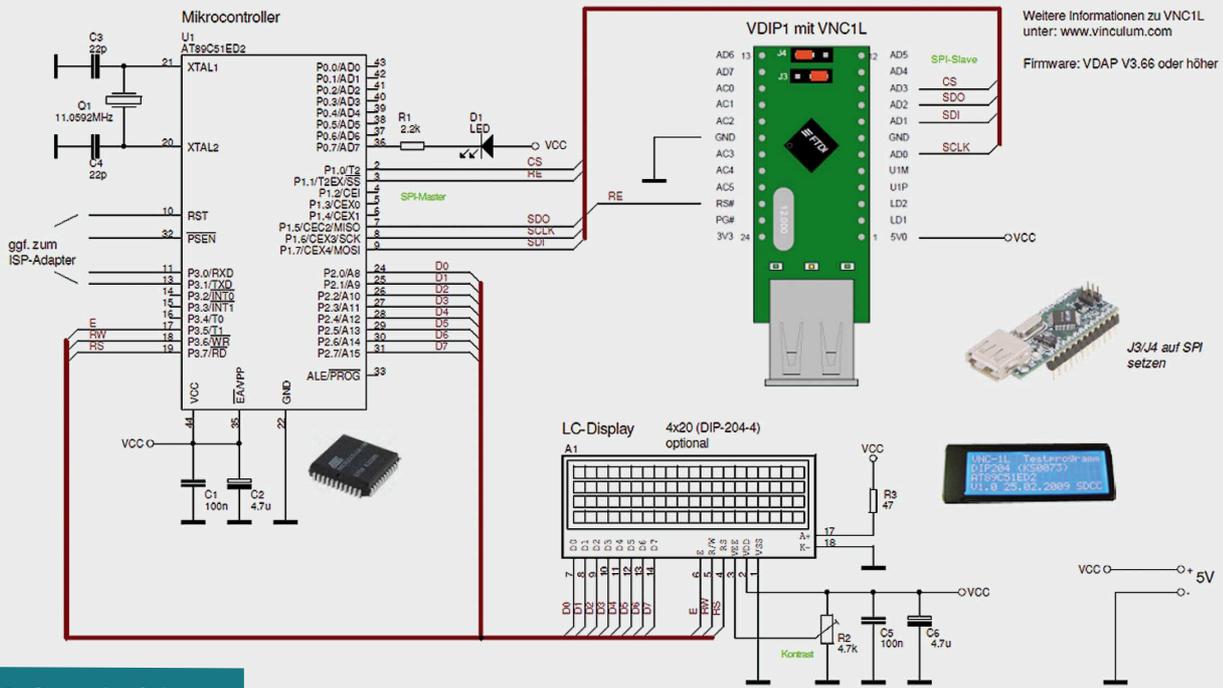


Abb. 3: Stromlaufplan

Zur Übertragung wird die SPI-Schnittstelle verwendet. Das SPI-Handling ist komplexer als beispielsweise die UART-Schnittstelle, hat aber den Vorteil, dass die meist nur einmal vorhandene, wertvolle Mikrocontroller-UART für andere Anwendungen erhalten bleibt.

Nicht vergessen, die Jumper J3 und J4 des VDIP1-Moduls auf SPI zu stecken!

Das VNC1L SPI-Protokoll weicht erheblich von dem üblichen SPI ab. Es ist eher eine Mischung aus SPI-, I2C- und UART-Eigenschaften und bedarf einiger Aufmerksamkeit. Wenn es einmal läuft, dann jedoch sehr stabil und bis 12MHz Busfrequenz schnell. Delays sind kaum notwendig. Da immer 13 Bit übertragen werden, ist eine Hardware-SPI auf

Mikrocontrollerseite praktisch nicht nutzbar. Das Timing wurde über normale Port-Pins per Software nachgebildet.

Der Mikrocontroller ist Master, der VNC1L Slave. Zum senden eines Zeichens an den VNC1L ist dies ideal, schwieriger erscheint das Empfangen von Zeichen vom VNC1L, da dieser keinen Interrupt-Pin hat, welcher das Vorhandensein neuer Zeichen signalisiert. Der VNC1L muss dennoch nicht ständig vom Mikrocontroller auf neue Zeichen gepollt werden, da er offenbar einen ziemlich großen FIFO hat. Ein „Tx-Buffer voll“ konnte in Tests nie erreicht werden.

Jedes übertragene Zeichen wird vom VNC1L im Status-Bit sofort auf Akzeptanz bestätigt, ähnlich dem Acknowledgment-Bit des I2C-Buses. Ein nicht bestätigtes Zeichen schaltet

LED D1 zur Signalisierung und Fehlersuche dauerhaft ein.

Auf den Anschluss des RS# (RE)-Pins als RESET-Funktion sollte man nicht verzichten. Damit hält man sich für den Fehlerfall einen Neustart des System offen.

Das Testsystem benutzt zur Anzeige ein alphanumerischen LC-Display mit 4x20 Zeichen, konkret ein DIP204-4 von EA, mit KS0073 Controller. Die Anschaltung und der Betrieb entspricht dem Standard und ist nicht Thema dieses Beitrages. Alle an den VNC1L gesendeten und von ihm empfangenen Daten können bei Bedarf auf dem Display angezeigt werden. Dies ist hilfreich bei der Inbetriebnahme und zum verstehen der Arbeitsweise, im späteren regulärem Betrieb natürlich nicht erforderlich.

Software

Es wird vorausgesetzt, das auf dem VNC1L die VDAC Software bereits installiert ist. Dies ist auf einem neuen VDIP1-Modul werkseitig der Fall. Meist ist jedoch eine ältere Version drauf. Das sollte man auch erst mal so lassen.

Der Stick muss mit FAT16 oder FAT32 formatiert sein. Es wird nur die erste primäre Partition unterstützt. NTFS oder andere Dateisysteme werden nicht erkannt.

Die VDAC-Firmware stellt praktisch alle Funktionen zum Zugriff auf dem USB-Stick bereit. Kenntnisse des FAT-Dateisystems sind

nicht erforderlich. Die Firmware arbeitet im Prinzip wie ein Monitorprogramm mit DOS-ähnlichen Kommandos. Nach einem erfolgreichen Kommando meldet sich die Firmware mit einem ‚D:\>‘ zurück.

Das Beispielprogramm ist in ANSI-C geschrieben und mit SDCC kompiliert. Es belegt inkl. der LCD-Funktionen und Test-Filedaten ca. 6kByte FLASH und 160Byte RAM. Eine Portierung auf andere Mikrocontroller sollte problemlos möglich sein. Spezielle SFR- und Zusatzkomponentennutzung wurde weitgehend vermieden. Das Programm läuft

in einer Endlosschleife und dürfte leicht zu verstehen sein.

Zunächst werden ein Reset des VNC1L und einige Initialisierungen durchgeführt:

- ECS - erweitertes Kommando-Set einstellen
- IPA - alle Zahlen werden als ASCII übertragen
- FWV - Anzeige der Softwareversion
- DIR - Anzeige des Stickinhaltes

Dazu wird die Funktion „vnc_wr_cmd“ verwendet.





AVRISP mkII
In-System-Programmer
19,90 €



AT90 USB Key
günstiges Board basierend auf AT90USB127 Mikrocontroller
19,90 €



NGW100
ideales Entwicklungsboard für AT32AP7000
49,90 €



AVR Butterfly
Evaluationsboard der neuesten AVR Technologie
11,90 €



STK500
AVR-Starter-Kit und Entwicklungssystem für alle 8-, 20- und 40-poligen AVR Mikrocontroller im DIL-Gehäuse von ATMEL
39,90 €



EVK1100
Evaluationskit und Entwicklungsumgebung für den AVR32 AT32UC3A Mikrocontroller
59,90 €



EVK1101
Evaluationskit und Entwicklungssystem für den AVR32 AT32UC3B Mikrocontroller
49,90 €



STK600
Starter-Kit und Entwicklungssystem für alle ATMEL AVR 8-Bit und 32-Bit Mikrocontroller
99,90 €



STK600-DIP
Zusatzmodul zum STK600 Socketboard mit DIP- und PDIP-Fassung
49,90 €



STK600-TQFP100
Zusatzmodul zum STK600 Socketboard mit TQFP100 Fassung
49,90 €



AVR Dragon
Low-Cost Entwicklungs-Tool
29,90 €

Wenn diese 4 Punkte funktionieren, kann man sich an ein Update der VDAP-Firmware trauen. Dazu die aktuelle Firmware von der Vinculum-Homepage [6] laden (endet mit .ftd), in „ftrfb.ftd“ umbenennen und diese Datei ins Root des USB-Sticks kopieren. Der Bootloader flasht (Anzeige „CHANGE MAIN“) die Datei dann automatisch beim nächsten anstecken.

Alle Funktionen wurden mit der VDAP-Version 3.66 und 3.68 ausprobiert. Die Software wird regelmäßig weiter entwickelt, aktuell ist zur Zeit Version 3.68.

Die danach folgenden Softwareteile zeigen, wie man eine Datei auf dem USB-Stick erzeugt bzw. einliest. Es können Text- und Binärfiles erstellt werden. Auch „anhängendes“ Schreiben ist möglich.

Besonderen Wert sollte man auf die elementaren SPI I/O-Funktionen „vnc_wr_byte“ und „vnc_rd_byte“ legen. Das Datenblatt [7] des VNC1L geht recht detailliert auf das Timing ein. Damit sollte man sich vorab gründlich beschäftigen. (Die Angaben zum SPI im Datenblatt des VDIP1 sind dagegen spärlich und teilweise nicht richtig!)

Fazit

Die Software wurde mit 5 unterschiedlichen USB-Sticks von 512MByte bis 16GByte getestet und funktionierte jedes mal ohne Änderungen. FAT16-Systeme sind schneller als FAT32-Systeme. Während der Testphase wurde, auch bei gravierenden Softwarefehlern, niemals ein Dateisystem auf den Stick zerstört, oder eine fremde Datei beschädigt. Die VDAP-Firmware scheint recht stabil zu funktionieren. Trotzdem sollte man nicht seinen einzigen Backup-Stick zum testen verwenden.

Links

- [1] Embedded Projects Journal 1. Ausgabe: http://www.ixbat.de/files/admin/projekte/epjournal/EPJ_01_download.pdf
- [2] VDIP1 Schaltung: <http://www.vinculum.com/documents/schematics/VDIP1%20Schematic%20Prints.pdf>
- [3] VDIP1 Datenblatt: http://www.vinculum.com/documents/datasheets/DS_VDIP1.pdf
- [4] PRO-51 Mikrocontrollerplatine: <http://pro-51.eltra-tec.de/>
- [5] Atmel 8051 Mikrocontroller: <http://www.atmel.com/products/8051>
- [6] Vinculum Downloads: <http://www.vinculum.com/downloads.html>
- [7] Datenblatt VNC1L: [http://www.vinculum.com/documents/datasheets/DS_VNC1L_V201\(FT_000030\).pdf](http://www.vinculum.com/documents/datasheets/DS_VNC1L_V201(FT_000030).pdf)
- [8] Vinculum User-Manual: http://www.vinculum.com/documents/fwspecs/UM_VinculumFirmware_V205.pdf
- [9] Beitrag im Mikrocontroller.net: http://www.mikrocontroller.net/articles/USB-Stick_am_Mikrocontroller
- [10] Youtube Video: <http://www.youtube.com/watch?v=oUX1U1ipfQk>



Abb. 4: Monitormeldung nach FWV-Kommando

Sehr hilfreich ist auch das verfügbare User-Manual [8]. Dort wird die VDAP-Firmware gut beschrieben. Alle Dokumente stehen leider nur in Englisch zur Verfügung.

Der vollständige Quelltext der Testsoftware ist im Projektarchiv des Beitrages [9] enthalten und kann als Basis für eigene Anwendungen verwendet werden.

Ein Video, welches dem Programmablauf auf dem LC-Display zeigt, kann unter [10] angesehen werden. Die Pausen zwischen den einzelnen Kommandos wurden zur besseren Demonstration hierfür etwas vergrößert.

Auf die LCD-Ansteuerung wird hier nicht näher eingegangen.

Riesige Schreib- und Lesegeschwindigkeiten kann sollte man bei der Verwendung der VDAP-Firmware nicht erwarten. Für die meisten Anwendungen, wie z.B. Datenlogger, dürfte diese jedoch ausreichend sein.

Mit dem VNC1L können grundsätzlich auch andere Geräte, wie z.B. USB-Drucker, Kameras und Handys angesteuert werden.

Das K-Modul Opensource-Projekt

Augsburger Computerforum (ACF) Mikrocontroller Projektgruppe (mcPG)

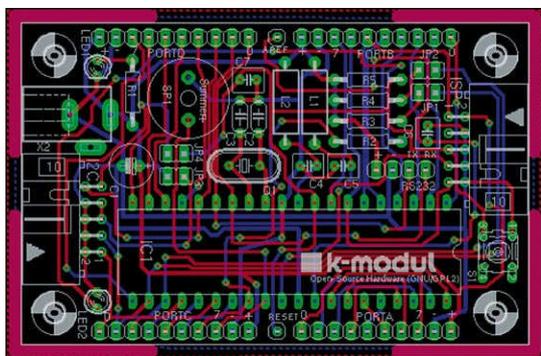
Wernfried Zolnhofer <zoli@augusta.de>

Wie alles begann

Vor gut einem Jahr haben sich nach einem Mikrocontroller-Workshop des Augsburger Computerforum e.V. (www.augusta.de) eine handvoll Interessierter zusammengenommen und die Mikrocontroller Projektgruppe (mcPG) ins Leben gerufen.

Die eigenen Anfangsschwierigkeiten noch gut im Gedächtnis, ist ein Ziel der Gruppe dem Anfänger den Einstieg in die Mikrocontrollertechnik (derzeit vorzugsweise Atmel AVR) zu erleichtern und dem fortgeschrittenen Anwender fertige Hard- und Softwaremodule (beides Opensource) an die Hand zu geben. (Siehe hierzu auch den Artikel „Open-Source Mikrocontroller Kurs“ auf Seite 13f in der *embedded-projects* Ausgabe 2.)

Abb. 2: Das K-Modul



Das K-CPU-Modul

Herzstück der K-Modulserie ist, wie der Name schon vermuten läßt, das K-CPU-Modul, das wir hier kurz vorstellen:

Gegenüber der ursprünglich geplanten Version erfolgte ein komplettes Redesign des Boards. Um für den Hobbyelektroniker die Hemmschwelle beim Lötaufbau möglichst niedrig anzusetzen, wurde auf SMD-Technik vollständig verzichtet. Bei Verwendung von Stecksockeln ist dadurch auch ein einfacher Austausch des Mikrocontrollers möglich, was man bei Versuchsaufbauten sehr schnell zu schätzen lernt. Die Ports des Mikrocontrollers (wahlweise ATmega 16/32/..644) sind komplett auf Buchsenleisten herausgeführt, so daß das Basismodul (K-CPU) einfach, z.B. über aufsteckbare Funktionsmodule, erweitert werden kann. Das Board ist dadurch nicht so überladen wie es häufig bei anderen

Entwicklerboards der Fall ist. Der Mikrocontroller wird standardmäßig mit einem externen 16 MHz Quarz getacktet. Neben dem obligatorischen Reset-Taster ist zur Status- und Fehleranzeige ein Piezosummer sowie eine Keepalive LED zu finden.

Auch für die Kommunikation ist gesorgt: Die ISP/SPI-Schnittstelle und der I²C-Bus (Interruptfähig) sind auf 10 pol. Wannensteckverbinder herausgeführt, die serielle Schnittstelle (TTL) liegt auf einer 4pol. (RX.TX,+,-) Stiftleiste.

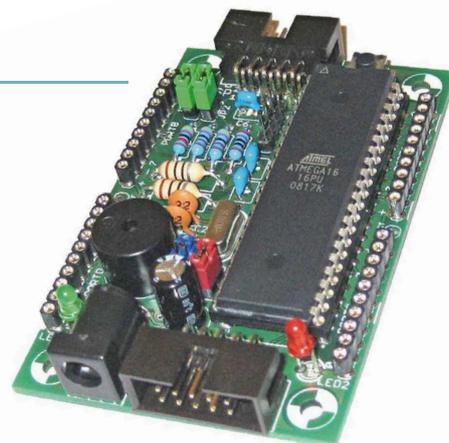


Abb. 1: Das K-Modul

Die Stromversorgung erfolgt wahlweise mit einem 5V Gleichstromnetzteil (>= 100 mA) über eine 3,1mm Hohlsteckerbuchse oder mit dem USB-Programmer über den Programmieradapter. Bei der Vernetzung mehrerer Module kann die Stromversorgung auch über den I²C-Bus erfolgen.

Um möglichst kompatibel zu bereits bestehenden Komponenten und Baugruppen zu sein, haben wir uns die RN-Normen (Abmessungen, Steckerbelegungen etc.) als Vorbild genommen.

Obwohl beim K-Modul auf SMD-Technik verzichtet wurde, konnte das Board im RN-Miniformat (80mm x 50mm) gehalten werden. Es eignet sich gut für die Ausbildung und als Basis für eigene Applikationen, zumal es universell eingesetzt werden kann.

Ausblick

Zur Zeit ist ein LCD-Anzeigemodul in Arbeit, weitere Funktionsmodule sollen folgen. Projektideen wie z.B.

- SD-Kartenspeicher
- USB-Anschluß
- Funkmodul
- Motorsteuerung (Gleichstrom/Schrittmotor/Servo)
- Testboard mit Schulungsunterlagen

sind reichlich vorhanden, doch leider sind unsere personellen Kapazitäten sehr begrenzt.

FAQs

Kann man mitmachen?

Zu dem Opensource-Projekt sind alle recht herzlich eingeladen, sich zu beteiligen. Bei Interesse einfach mit dem Autor dieses Artikels Kontakt aufnehmen.

Wie kommt man an das Board?

Das Board kann demnächst über den Internetshop von Michael Sauter bezogen werden.

Unterlagen zum Board

Unterlagen sind auf der Homepage des ACF zu finden.

Links

- [1] <http://www.augusta.de/AG/micro/>
- [2] http://www.ixbat.de/files/admin/projekte/epjournal/EPJ_02_download.pdf
- [3] <http://www.rn-wissen.de/index.php/RN-Definitionen>
- [4] <http://shop.embedded-projects.net/>
- [5] <http://www.augusta.de/AG/micro/folder.2009-08-26.0260596259/>

LED Flitzer

POV-AVR-Board

<info@zipfelmaus.com> www.zipfelmaus.com



Einleitung

Beim LED-Flitzer handelt es sich um ein kleines, tragbares, batteriebetriebenes Spielzeug zum Darstellen von Wörtern mittels des „Persistence Of View“-Prinzips (Nachbildwirkung). Seine pfiffigen Extras wie USB-Schnittstelle und Beschleunigungssensor machen es zu einem interessanten Bastelprojekt für Zwischendurch.

Begriff POV – Persistence Of View

„Persistence Of View“ zu Deutsch Nachbildwirkung, bezeichnet ein mangelhaftes zeitliches Auflösungsvermögen des menschlichen Auges. Sie wird erzeugt durch das Abklingen des Lichtreizes auf der Netzhaut, der für eine kurze Zeit nach seinem Ende nachwirkt. Bilder, die in genügend schneller Bildfrequenz aufeinanderfolgen, verschmelzen im Gehirn zu einer durchgehenden Bewegung. Dieses Prinzip ist somit die physiologische Grundlage des Films und des Fernsehens.

(Quelle: wikipedia.de)

Funktionsprinzip

Die Nachbildwirkung wird mit acht nebeneinander platzierten hellen LEDs erzeugt, die mit einem sehr kurzen, gleichbleibenden Zeittakt (~ 3ms) angesteuert werden. Zusätzlich muss die Platine dabei bewegt werden, damit der gewünschte Eindruck im Auge erzeugt wird (z.B. an den Fuß geschnallt beim Laufen oder Tanzen oder in der Hand beim Winken).

Für den LED-Flitzer existiert eine Fontlib mit Großbuchstaben. Jeder einzelne Buchstabe benötigt zwischen einem bis fünf LED-Takte. Das Auge kann beim LED-Flitzer etwa Wörter mit bis zu fünf Buchstaben gut erkennen. Längere Wörter können zwar programmiert werden, eignen sich aber nur für Langzeitbelichtungen mit dem Fotoapparat, da der Effekt bei längere Wörtern nicht zustande kommt.

Projektumfang

Neben der Hard- und Software für den LED-Flitzer wurde auch ein GUI-Tool entwickelt, um neue Texte ohne aufwendige Firmware-Anpassung via USB auf die Platine laden zu können. Wer will, kann die USB Schnittstelle und den Beschleunigungssensor auch erst später hinzufügen und den LED-Flitzer in einer abgespeckten Variante betreiben.

Features:

- µC Atmega8L
- µC Firmware in C (AVRStudio + gcc)
- font-lib für AVR mit Großbuchstaben
- USB-Schnittstelle mittels FTDI-Baustein (optional)
- 3-Achsen Beschleunigungssensor zum Ermitteln von Bewegungen (optional)
- GUI-Tool zum Programmieren von neuen Texten mittels USB-Schnittstelle (optional)

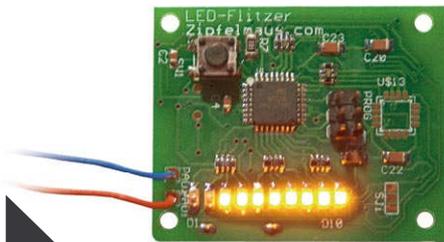


Abb. 1:



Abb. 2: Anwendung LED Flitzer

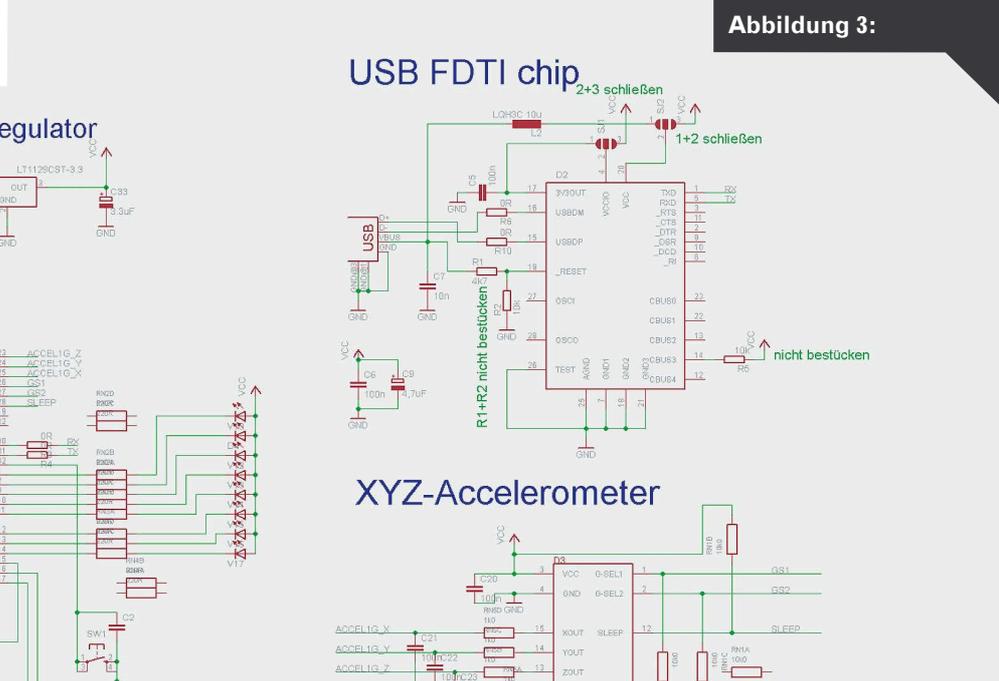
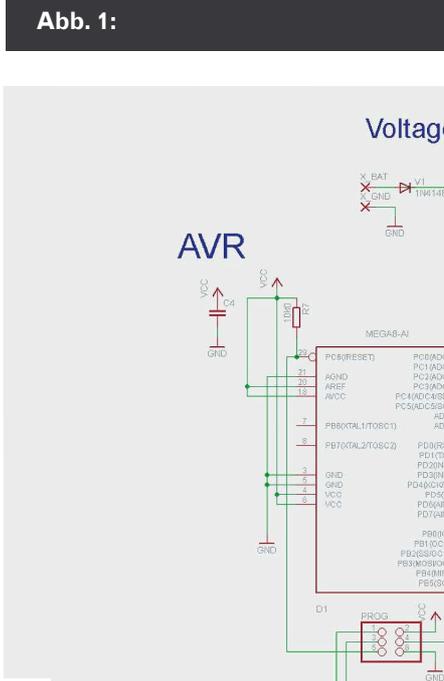


Abbildung 3:

Hardware

Spannungsversorgung

Der LED-Flitzer bezieht seine Spannung aus 4 AAA-Batterien. Um die Hardware vor Schäden beim falschen Einlegen der Batterien zu schützen, wurde die Diode V1 vor den Spannungsregler geschaltet. Der Spannungsregler LT1129 erzeugt eine Systemspannung von 3,3V.

Mikrocontroller

Beim Mikrocontroller D1 handelt es sich um einen Atmega8L. Im vorliegenden Schaltplan wurde er in der Minimalbeschaltung integriert. Die einzigen externen Komponenten sind ein Pullup-Widerstand für den Reset, ein Koppelkondensator und ein ISP-Programmierstecker.

Schnittstellen

Die Firmware wird über den ISP-Programmierstecker geladen. Über die UART-Schnittstelle ist der USB-Baustein (D2) angeschlossen. Außerdem liest der ADC die analogen Ausgangsspannungen des Beschleunigungssensors (D3) ein. Insgesamt acht Kanäle der Ports D und

B werden zum Ansteuern der LEDs verwendet. Ein Taster ist an den Interrupt-Pin INT0 angeschlossen.

USB Baustein

Eine vergleichsweise unkomplizierte Anbindung an die USB-Schnittstelle bietet der Baustein FT232R (D2). Er ist in der „USB Bus Powered with Selectable External Logic Supply“-Schaltung aufgebaut (siehe Datenblatt, Link [2]).

Beschleunigungssensor (optional)

Der MMA7260Q (D3) von Freescale ist ein 3-Achsenbeschleunigungssensor und wird zur Bestimmung von Bewegungen des LED-Flitzers verwendet (Datenblatt unter Link [3]). So können die LEDs angesteuert werden, sobald der Sensor eine Bewegung erkennt.

Dieser Baustein liegt leider nur in einem QFN16-Gehäuse vor, was das Löten zu einer echten Herausforderung macht. Mit dem richtigen Equipment (z.B. Heißluftlötlötstation) und/oder einiger Löterfahrung lässt sich das allerdings erfolgreich meistern.

100ms-Timer werden die ADC-Werte gemittelt und festgestellt, ob eine Bewegung stattgefunden hat. Im 3ms-Timer findet schließlich die Ansteuerung der LEDs statt.

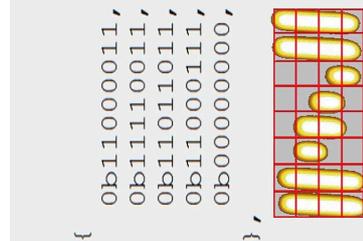
PC

Für den PC gibt es ein kleines Tool in C#. Mit dem man über USB neue Texte auf den LED-Flitzer übertragen kann.

Ablauf Übertragen neuer Text-Message

1. Power Off
2. Anschließen an PC via USB
3. Starten des PC-Tools
4. Power On und gleichzeitiges Betätigen des Tasters auf dem LED-Flitzer
5. Eingabe neuer Text-Message
6. Übertragen der Text-Message
7. Power Off
8. Entfernen der USB-Schnittstelle
9. Power On
10. neue Text-Message geladen

Abb. 4:



Komponenten

C1, C2, C5, C6, C20, C21, C22, C23	100nF
C7	10nF
C9	4,7uF
C33	3,3uF
D1	MEGA8-AI
D2	FT232RL
D3	MMA7260Q
L1	LQH3C 10u
PROG	Stiftleiste 2x3
R1	4k7
R2, R5, R7	10k
R3, R4, R6, R10	0R
RN2, RN3, RN4	330R Array
RN5	1k0 Array
SJ_2	jumper
SJ_2	jumper
SW1	Taster
T1	LT1129CST-3.3
USB1	USB Mini SMD
V1	1N4148
V10, V11, V12, V13, V14, V15, V16, V17	CHIPLED_1206

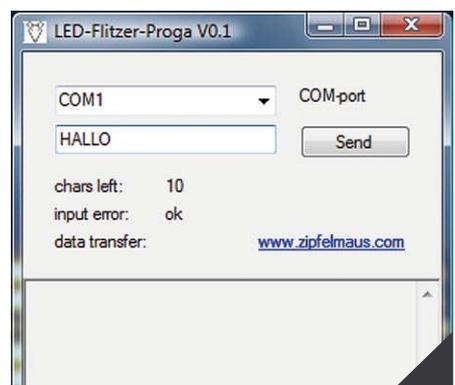


Abb.5:

Software

AVR

Die Software wurde in C im AVR-Studio geschrieben und mit dem gcc kompiliert. Mit nur vier Files ist der Code recht übersichtlich und kann leicht angepasst werden.

Ablauf

Initialisierung

Nach dem Reset werden die einzelnen Register des µC gesetzt. Anschließend wird geprüft, ob neue Textdaten an der USB-Schnittstelle vorliegen. Diese werden dann in das EEPROM abgelegt und stehen nach einem Hardware-Reset wieder zur Verfügung. Anschließend wird ein Timer gestartet, in dessen Interrupt-Routine ein Millisekundenzähler inkrementiert wird.

while(1)- Schleife

Der Ablauf der Schleife besteht aus drei Timerabfragen, nach jeweils 3ms, 10ms und 100ms. Nach 10ms liest der ADC die Werte vom Beschleunigungssensor ein. Im

Downloads & Links

Alle Sourcefiles, der Platinenschaltplan und das Layout können von meiner Webseite heruntergeladen werden (Link[1]).

[1.] <http://www.zipfelmaus.com/led-flitzer/>

[2.] http://www.ftdichip.com/Documents/DataSheets/DS_FT232R.pdf

[3.] http://www.freescale.com/files/sensors/doc/data_sheet/MMA7260QT.pdf



Die Anix GmbH entwickelt, fertigt und vertreibt seit 2004 moderne Prüfgeräte für das Bauwesen und die Industrie. Wir sind der Hersteller vom Plattendruckgerät AX01 und wir produzieren eine Auswertelektronik für den Planografen.

Wir suchen ein/zwei gute:

- Ingenieure
- Entwickler
- Programmierer
- Mechaniker



Bitte Bewerbung an:

Anix GmbH
Dipl.-Ing. Matthias Weingart
Meitzendorf, Hintern Hecken 1
39179 Barleben

Tel: 039202 8792-52
Fax: 039202 8792-57
E-Mail: bewerbung@anix.biz

EIN SOFTER JOB?



Software-Ingenieure - Echtzeit und Embedded Systeme

www.ibv-augsburg.net/jobs

IBV - ECHTZEIT- UND EMBEDDED GMBH & CO. KG



www.mixed-mode.de/jobs.htm

Technik

Mensch

Leidenschaft

**MIXED
MODE** Software- & Systementwicklung

Mit uns auf
Erfolgskurs



**weather
dock**

Die Firma Weatherdock AG ist Träger des IHK-Gründerpreises 2008 und entwickelt elektronische Produkte und Lösungen für die Sicherheit in der Sportbootschifffahrt.

Wir suchen einen/eine

E-Technik Studenten / in

für Praktikantentätigkeiten oder Diplom-Arbeiten.

Bitte senden Sie Ihre Unterlagen per Email an:
Frau Dominique Stojanovic · dstojanovic@weatherdock.de
Am Weichselgarten 7 · 91058 Erlangen · www.weatherdock.com

Sie suchen?

Hier werden Sie gefunden!

Stellenanzeigen

im Embedded Projects Journal
Werbung - zielgerichtet

Infos: www.embedded-projects.net

Studenten der Informatik
und E-Technik gesucht.

noT.

Net of Trust beschäftigt sich mit Forschung und Entwicklung von Software und Hardware mit dem Schwerpunkt Innovative Technologie für Sicherheit, Forensik und Messtechnik an der Universität der Bundeswehr München.

Werkstudenten, Praxissemester und Diplomarbeiten gesucht!

Wir suchen für unser Team in Augsburg Verstärkung:

- **Embedded Systeme** (AVR, ARM9, AVR32)
- **GNU/Linux Programmierung** (Treiber, Anwendungen)
- **Schaltungsentwurf / Platinenlayout**
- **Internet Programmierung** (PHP, Python, Perl, SQL)
- **Kryptografie bzw. Mathematik**

Net of Trust Solution GmbH · An der Universität der Bundeswehr München
Zweigniederlassung Augsburg · Holzbachstr. 4 · 86152 Augsburg
Telefon: 0821 / 27 95 99 02 · E-Mail: bewerbung@netoftrust.net

der Bundeswehr
Universität München

Werdet aktiv! _____

Das Motto: Von der Community für die Community!
Das Magazin ist ein Open-Source Projekt.

Falls Du Lust hast, Dich an der Zeitschrift durch einen Beitrag zu beteiligen, würden wir uns darüber sehr freuen. Schreibe Deine Idee an:

sauter@embedded-projects.net

Wir werden dann gemeinsam sehen, was wir daraus machen können.

Regelmäßig lesen! _____

Die Zeitschrift wird über mehrere Kanäle verteilt. Der erste Kanal ist der Download als PDF-Datei. Alle Ausgaben sind auf der Internetseite [1] verfügbar. Diejenigen, die lieber eine Papierversion erhalten möchten, können den zweiten Kanal wählen. Man kann sich dort auf einer Internetseite [1] in eine Liste für die gesponserten Abos eintragen. Beim Erscheinen einer neuen Ausgabe wird dank Sponsorengeldern an jeden auf der Liste eine Ausgabe des aktuellen Journal versendet. Falls man den Versandtermin verpasst hat, kann man das Heft auch zum Preis von einem Euro über einen Online-Shop [2] beziehen.

1. Internetseite (Anmeldeformular gesponserte Abos)

<http://www.embedded-projects.net/journal>

2. Online-Shop für Journal (Preis 1 EUR + Versand)

<http://www.eproo.de/journal>

Sponsoren gesucht! _____

Damit wir weiterhin diese Zeitschrift für jeden frei bereitstellen können, suchen wir dringend Sponsoren für Werbe- und Stellenanzeigen. Bei Interesse meldet Euch bitte unter folgender Telefonnummer: 0821/27 95 990 oder sendet eine E-Mail an die oben genannte Adresse.

[IMPRESSUM] _____

embedded projects GmbH
Holzbachstrasse 4
D-86152 Augsburg
Telefon: +49 (0) 821 / 27 95 99 - 0
Telefax: +49 (0) 821 / 27 95 99 - 20
Mail: journal@embedded-projects.net

Anzeigemöglichkeiten und Preisliste
auf Anfrage via Mail.

Herausgeber: Benedikt Sauter
Gestaltung/Satz: Das-Medienkollektiv.de

Veröffentlichung: 4x / Jahr
Ausgabeformate: PDF / Print
Auflage Print: 2500 Stk.
Einzelverkaufspreis: 1,00 EUR

Dies ist ein Open-Source Projekt.
Informationen zum ABO - www.embedded-projects.net/journal

Titelfoto: seen@fotolia.com
Fotocomposing: Das Medienkollektiv
EPJ Anzeige: Ramona Heim, Nicemonkey,
dip,...@fotolia.com



Alle Artikel in diesem Journal stehen unter der freien Creative Commons Lizenz. Die Texte dürfen - wie bekannt von Open-Source - modifiziert und in die eigene Arbeit mit aufgenommen werden. Die einzige Bedingung ist, dass der neue Text ebenfalls wieder unter der gleichen Lizenz, unter der dieses Heft steht, veröffentlicht werden muss, und zusätzlich auf den originalen Autor verwiesen werden muss. Ausgenommen Firmen- und Eigenwerbung.



Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by/3.0/> Ausgenommen Firmen- und Eigenwerbung.



SPEZIALISTEN GESUCHT

Dein Spaß an High-tech
Deine Herausforderungen an Bits und Bytes
Deine Sendung mit der Maus - für Erwachsene
Du willst kreativ Entwicklungsprozesse mitgestalten -
werde Teil der Elektronikwelt.

Software-Ingenieure (m/w) Echtzeit und Embedded Systeme

Zur Verstärkung unseres Entwicklerteams in Königsbrunn bei Augsburg suchen wir ab sofort deutschsprachige Informatiker, Ingenieure oder Naturwissenschaftler mit abgeschlossenem Hochschulstudium und Erfahrung in der Entwicklung technischer Software.

Infos: <http://www.ibv-augsburg.net/jobs>

IBV - ECHTZEIT- UND EMBEDDED GMBH & CO. KG

Keltenstraße 2 D-86343 Königsbrunn
Fon +49 (0) 82 31.95 86 -041 Fax +49 (0) 82 31.95 86 -049
info@ibv-augsburg.net www.ibv-augsburg.net

ibv.
Realtime is BLUE