



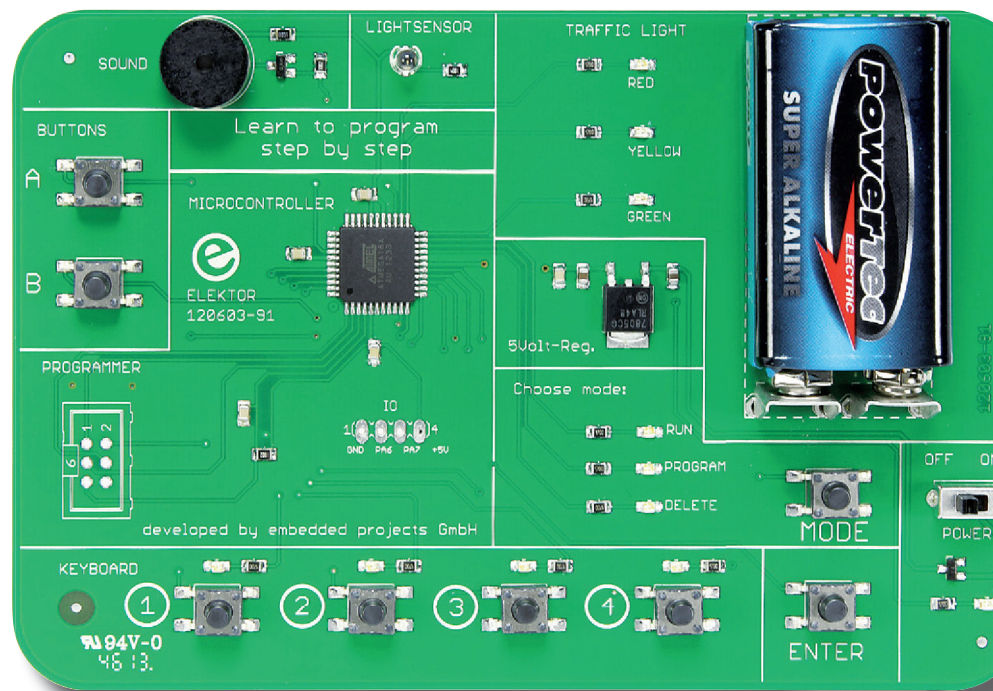
embedded projects

JOURNAL

OPEN SOURCE SOFT-AND HARDWARE PROJECTS

Eine Open-Source Zeitschrift zum Mitmachen!

back to the roots



[PROJECTS]

- EMV: Möglichkeiten und Grenzen externer AC/DC-Schaltwandler
- Gerber: Vergangenheit, Gegenwart und Zukunft
- BGAs mit Eagle layouts
- PowerPi: Funksteckdose per RaspberryPi
- Wettbewerb: Ein Protokoll für das Internet of Things
- Analogen Stromzähler elektronisch auslesen
- Backport Kernel: Neue Hardware auf altem Kernel
- Learn to program - Schritt für Schritt Programmieren kennen lernen

**Wussten Sie,
dass wir eine Firma
für kundenspezifische
Entwicklungen mit
Sitz in Augsburg sind?**



Wir bieten:

Hardware, Software,
Embedded, Software-
Entwicklung,
Mikrocontroller,
Anwendungsentwicklung,
Fachbeiträge/
Literatur, Schaltplan,
Webentwicklung,
Open-Source,
E-Commerce, Platinen-
layout, GNU/Linux

Kommen Sie vorbei!



embedded projects GmbH
HARDWARE FOR PROJECTS

Holzbachstraße 4, D-86152 Augsburg
Tel +49 (0) 821 279599-0
Fax +49 (0) 821 279599-20
info@embedded-projects.net

Einleitung

Ausgabe 01/2014

Embedded Projects Journal - Ausgabe No. 20

Einleitung

In dieser Ausgabe haben wir wieder einige spannende Artikel. Los geht es mit einem Artikel über EMV bzw. Möglichkeiten und Grenzen externer AC/DC Schaltwandler bezüglich der elektromagnetischen Verträglichkeit. Dieses Thema ist immer aktuell, das hat man auch an unserer letzten Schulung gemeinsam mit der Firma Würth Elektronik in Friedberg gesehen. Fast 60 Besucher waren zu Gast bei der kostenlosen Schulung. Eventuell haben es noch nicht alle mitbekommen, daher erwähne ich es an dieser Stelle gerne wieder. Regelmäßig bieten wir Schulungen für verschiedene Themen an. Wer noch keine E-Mail bekommt, kann uns gerne anschreiben, um mit in den Verteiler aufgenommen zu werden. Aktuell hatten wir als Themen: Eagle 6.x, SMD Bestückung, Spulen, Speichern und Drosseln bzw. deren EMV Verträglichkeit.

Im regelmäßigen Abständen möchten wir weitere Schulungen anbieten. Für Ideen und Anregungen sind wir jederzeit offen.

An dieser Stelle möchte ich noch den Artikel „Ein Protokoll für das Internet of Things!“ vorstellen. Dieser ist ein Aufruf zu einem Wettbewerb, den wir gemeinsam mit Elektor gestartet haben. Es gibt so viele Implementierungen und Protokolle. Wir möchten im Rahmen des Wettbewerbs eine Lösung bzw. eine Implementierung finden, mit der man so viele verschiedene Geräte wie möglich ansteuern kann - aber auch so einfach wie möglich. Bedingung: das Protokoll muss auf einem Poster erklärt werden können, mehr Platz gibts nicht!

Eure Ideen oder bereits bestehende gute Implementierungen können hier eingereicht werden: www.iot-contest.com

Benedikt Sauter
und das embedded projects Team

→ shop.embedded-projects.net

HARDWARE FOR YOUR PROJECTS – ONLINESHOP

Design your GNUBLIN

Sie suchen ein Board, das fast so wie GNUBLIN ist und brauchen davon nur eine kleine Menge pro Jahr? Wir passen Ihnen auf kurzem Weg die Schaltung an und ergänzen diese nach Ihrem Wunsch. Dank unserer internen Bestückung können wir Ihnen

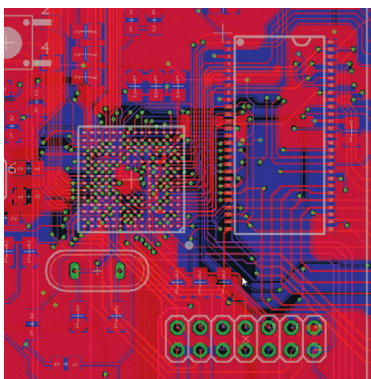
bereits ab kleinen Mengen ähnliche Stückpreise wie bei den GNUBLIN Boards in diesem Shop liefern.

→ www.gnublin.org/designer

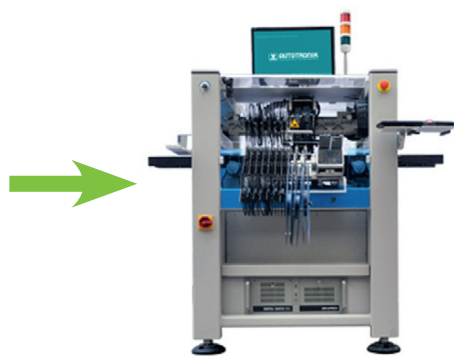
Die einfachste Möglichkeit an ein eigenes embedded GNU/Linux Board zu kommen!

Einfach und genial ist das Produkt „Design your GNUBLIN“. Für alle, die kleine Serien von Platinen mit embedded GNU/Linux benötigen. Basierend auf der Schaltung der Boards der GNUBLIN Familie können wir einfach und schnell kundenspezifische Lösungen erstellen. Innerhalb kürzester Zeit können wir Ihnen die Boards von 1 bis ca. 1000 Stück liefern.

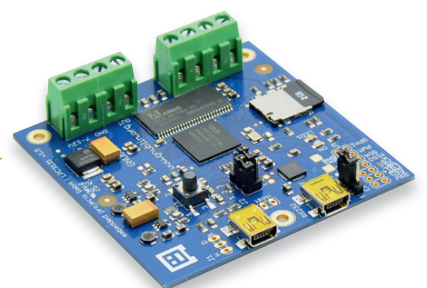
NEU:
online GNUBLIN-Designer
mit Kalkulator



Wir zeichnen den Schaltplan und das Layout ...



... bestücken mit einem Automaten in Augsburg ...



GNUBLIN

... und nehmen die Schaltung für Sie in Betrieb.

Holzbachstraße 4, D-86152 Augsburg
Tel +49 (0) 821 279599-0
Fax +49 (0) 821 279599-20
shop@embedded-projects.net



embedded projects GmbH
HARDWARE FOR PROJECTS

Kostensenkung durch Online-Pooling

- Keine Einrichtungskosten
- Keine Mindestbestellwerte - ab der 1. Platine
- Sofort-Bestellung Online - ohne Vorkasse

Zeitgewinn durch neue Online-Daten-Dienste



PCB Visualizer Sofort Design-Rule-Check



PCB Checker sofortige Anzeige von DRC Fehlern im Layout



PCB Configurator kalkulieren und prüfen von Layout-Parametern, Anzeige von Preis-Optionen



Wir gratulieren EAGLE
zu 25 Jahren Flug im
Aufwind

25 Jahre CadSoft



1988 - 2013

PCB proto

spezieller Prototypen-Service für Entwickler,
preiswert und schnell

- 1, 2 oder 5 LP in 2, 3, 5 oder 7 Arbeitstagen
- 2 und 4 Lagen
- DRC-geprüft, prof. Ausführung,
inkl. 2 x Lötstopplack, 1 x Bestückungsdruck,
150µm Technologie

STANDARD pool

die größte Auswahl an Eurocircuits Pooling
Optionen

- FR-4, für bleifreies Löten geeignet, bis 16 Lagen
- Layout-Technologie bis 90µm
- ab 2 AT

RF pool

alle Pooling-Vorzüge mit
Hochfrequenz-Material

- Material: Isola I-TERA und Rogers 4000 Serie
- 2-4 Lagen, bis 100µm Technologie
- ab 3 AT

IMS pool

Aluminiumkern-Leiterplatten für hohe
Wärmeableitung (z.B. LED-Anwendung)

- Leiterplatten mit einlagig isoliertem Metallsubstrat
- weisser/schwarzer Lötstopplack/Bestückungsdruck
oder umgekehrt
- ab 3 AT

www.eurocircuits.de

HOLEN SIE SICH DIE NEUESTE VERSION!

EAGLE

V6

Neue Funktionen der Version 6.4:

- Simulation Ihres EAGLE Schaltplans in LTspice IV
- Anzeige und Suchfunktion für Attribute im ADD- und REPLACE-Dialog
- Import von Designdaten aus P-CAD, Altium und Protel über das Zwischenformat ACCEL ASCII
- Verbesserte Benutzerführung und Voreinstellungen (Tool-tips, Shortcuts)



www.cadsoft.de

25 Jahre CadSoft



EMV - Möglichkeiten und Grenzen externer AC/DC-Schaltwandler bezüglich der elektromagnetischen Verträglichkeit

Kai Dorau M. Ed. <kai.dorau@gmx.net>

Soll ein elektronisches Gerät mit Strom versorgt werden, bietet sich die Nutzung von externen Stromversorgungsquellen geradezu an. Bei der Nutzung von solchen externen AC/DC-Schaltnetzteilen ist deren Spezifikation von sehr großer Bedeutung. In erster Linie sind die elektrischen und mechanischen Eigenschaften für die Auswahl eines Schaltreglers zu nennen. Möchte man sein Gerät mit externen Schaltwandlern in Europa vertreiben, benötigt man das CE-Zeichen, das mit den entsprechenden sog. EMV-Tests verbunden ist. Gerade diese EMV (Elektromagnetische Verträglichkeit) wird oft unterschätzt. Dieser Artikel gibt einen kleinen Einblick darüber, wie die Störstrahlung in Schaltnetzteilen entsteht und wie sie wirksam minimiert werden kann.

Schematischer Aufbau von AC/DC-Wandlern

AC/DC-Schaltregler als Spannungs- und Stromwandler existieren in unterschiedlichen Ausführungen. Abwärts-, Aufwärts- oder Inverswandler sind in der Elektronik am häufigsten anzutreffen. An dieser Stelle soll die Funktionsweise eines primär getakteten Wandlers der Einfachheit halber nur schematisch vorgestellt werden (Abbildung 1).

Die Netzeingangsspannung wird zuerst einmal gleichgerichtet und gesiebt (Spannungsverlauf 1 und 2). Danach sorgt ein Schalttransistor (meist MOSFET), der über einen Controller gesteuert wird, für das Zerhacken der Gleichspannung (Spannungsverlauf 3). Die Frequenz ist in der Regel zwischen 40 und 200KHz. Der Übertrager stellt die galvanische Trennung zwischen Primär- und Sekundärseite dar.

Die übertragene Primärschaltspannung (Spannungsverlauf 4) wird sekundärseitig wiederum gleichgerichtet und gesiebt als Ausgangsspannung zur Verfügung gestellt (Spannungsverlauf 5). Die in Abbildung 1 dargestellte Isolation ist ein Feedback-Kanal, um auf sekundärseitige Störungen zu reagieren. Dies geschieht häufig über eine Pulsweitenmodulation.

Die Vor- und Nachteile von Schaltwandlern sind in Tabelle 1 gezeigt.

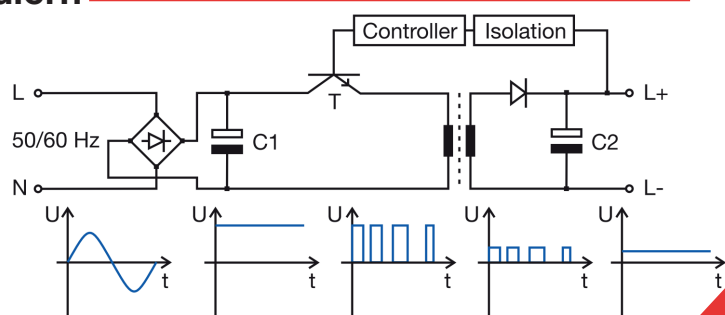


Abb. 1: Aufbau primär getakteter Schaltwandler (Quelle: unbekannt)

Vorteile	Nachteile
Kleine Bauform	Hoher Schaltungsaufwand
Geringes Gewicht	Netzurückwirkungen
Großer Eingangsspannungsbereich	Druch Hochfrequenz Entstörung notwendig
Leicht regelbar	Hohe Kosten
Hoher Wirkungsgrad	
Pufferung bei Netzausfall	
DC-Versorgung	

Tabelle 1: Vor- und Nachteile eines Schaltwandlers

Etwas Signaltheorie

Die in Abb. 1 gezeigten Signalverläufe legen dar, dass die zerhackten Spannungsverläufe rechteckförmig sind. Aus der Fourier-Analyse weiß man, dass jedes rechteckförmige Signal aus einer unendlichen Summe von sinus- und cosinusförmigen Signalen „nachgebildet“ werden muss. Sei der rechteckförmige Spannungsverlauf f gegeben, so setzt sich die Fourierreihe, deren Frequenzen ganzzahlige Vielfache der Grundfrequenz

$$\omega = \frac{2\pi}{T}$$

beinhalten, folgendermaßen zusammen:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cdot \cos(k\omega t) + b_k \cdot \sin(k\omega t))$$

Die Grundschwingung stellt sich bei $k = 1$ ein, die 1. Oberschwingung bei $k = 2$ usw. Graphisch sieht der Sachverhalt nun so aus (Abb. 2).

Die Addition der in der 1. Spalte aufgeführten Signale ist in der 3. Spalte zu sehen. Es formiert sich langsam ein Rechtecksignal. Die dazugehörigen Spektren sind in der 4. Spalte angeordnet. Man sieht, dass das Spektrum des Rechtecksignals mit zunehmendem Oberwellenanteil stark wächst. Die Fourierreihe des Rechtecksignals lautet:

$$f(t) = \frac{4h}{\pi} \sum_{k=1}^{\infty} \frac{\sin(2k-1)\omega t}{2k-1}$$

Ein paar Worte an dieser Stelle zum Dirac-Impuls. Er ist ein mathematisches Modell mit unendlich schmaler Pulsbreite und einer normierten Amplitude von 1

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

(siehe Abb. 3 li.).

Das Frequenzspektrum besteht aus unendlich vielen Oberwellen, die alle über eine Amplitude von 1 verfügen (siehe Abb. 3 re.). Das Phänomen der Störfrequenzbildung bei der Erzeugung von rechteckförmigen Pulsen ist gerade bei dem Dirac-Impuls sehr deutlich erkennbar.

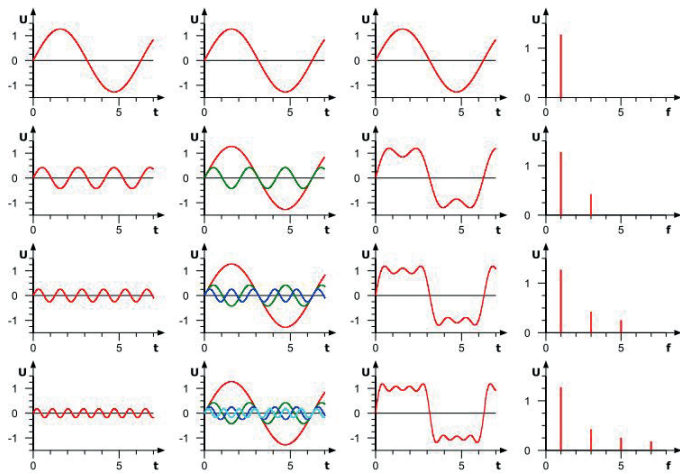
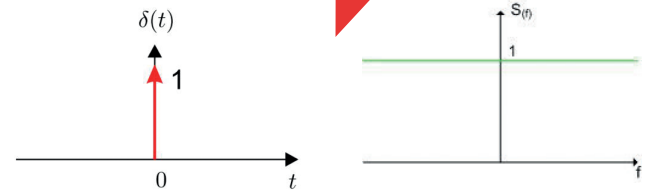


Abb. 2: Fourierreihe mit $k=4$ und den entsprechenden Spektren (Wikipedia)

Abb. 3: Zeitbereich und Frequenzspektrum des Dirac-Impuls (Quelle: Wikipedia)



Woher kommen nun die Störsignale?

Die für das Zerhacken der Gleichspannung benötigten Leistungshalbleiter sind durch eine sehr steiflankige Schaltcharakteristik gekennzeichnet. Die üblichen MOSFETs haben Anstiegs- und Abfallzeiten (t_r , t_f) von wenigen Nanosekunden. Typische Werte sind 5ns bis 10ns. Das hat zur Folge, dass das Spektrum an Oberwellen beim Zerhacken entsprechend groß ist. Dieses Spektrum trägt maßgeblich zur Gleichtaktstörung von Schaltwandlern bei.

Elektronische Geräte enthalten häufig Mikrocomputer- oder Mikrocontroller-Schaltungen. Die Stromaufnahme dieser Geräte ist von impulsförmiger Struktur. Die Stromversorgung in Gestalt eines Schaltwandlers muss ein hohes di/dt erbringen. Diese Situation trägt zusätzlich zu einer Erhöhung des Gegentaktstörspektrums bei.

Wird nun ein Gerät über einen externen AC/DC-Schaltwandler gespeist, so sind die leitungsgebundenen Störungen von besonderem Interesse. Das Störspektrum wird sich einerseits in Richtung AC-Netz über das Netzkabel, andererseits in Richtung Gerät über das Stromversorgungskabel ausbreiten. Die in das Gerät hineingehende Störstrahlung wird sich dann auf die störungseigenschaften der Interfaces auswirken. Die feldgebundene Störstrahlung ist häufig unproblematisch.

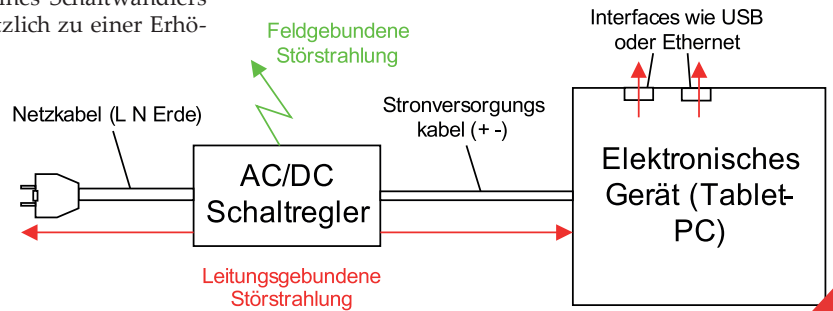


Abb. 4: Leitungs- und feldgebundene Störstrahlungsabstrahlung eines Schaltreglers

Was bedeutet EMV?

Die elektromagnetische Verträglichkeit ist gem. EMVG §4(1) und (2) „die Fähigkeit einer elektrischen Einrichtung, in ihrer Umgebung zufriedenstellend zu funktionieren, ohne diese Umgebung, zu der auch andere Einrichtungen gehören, unzulässig zu beeinflussen!“

Die folgende Grafik verdeutlicht die EMV-Anforderungen (Abb. 5).

In dem Artikel wurde schon öfters die leitungsgebundene Störstrahlung erwähnt. In der Grafik befindet sie sich ganz links: „Störaussendung (EMI) / geleitete Störenergie.“ Die Messung und die Grenzwerte dieser leitungsgebundenen Störstrahlung sind in einer Norm festgehalten und sind unter anderem Grundlage für das Führen des CE-Zeichens.

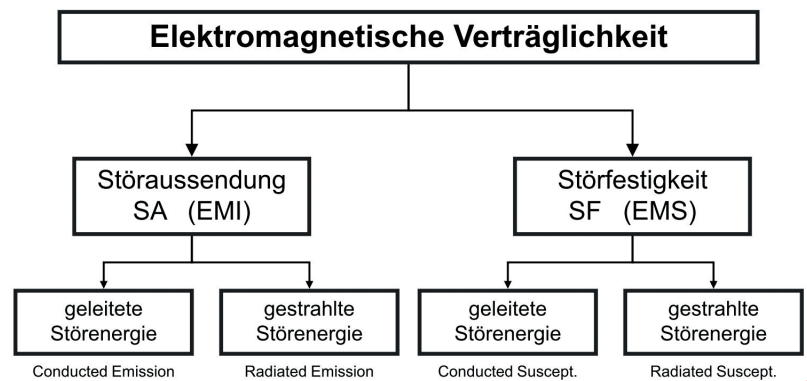


Abb. 5: Anforderungen der EMV (Quelle: Würth Elektronik)

Die Normen und Strahlungsmessung nach EN55022

Die entsprechenden Normen für Störaussendung sind in der Tabelle zusammengefasst (Tabelle 2). Wie man sieht, gibt es unterschiedliche Normen für unterschiedliche Geräte. Ein Tablet-PC in Verbindung mit einem externen Schaltnetzteil unterliegt der EN 55022, weil es eine informationstechnologische Einrichtung darstellt.

In dem nachfolgenden Beispiel werden die Messungen an einem Prüfling durchgeführt, der in die Kategorie „Informationstech- ➔

Internationale Norm	Europäische Norm	Bedeutung
CISPR 22	EN 55022	Feld- und leitungsgebundene Störstrahlung von informationstechnologischen Einrichtungen
CISPR 14	EN 55014	Feld- und leitungsgebundene Störstrahlung von Haushaltsgeräten
CISPR 11	EN 55011	Feld- und leitungsgebundene Störstrahlung von ISM-Geräten

Tabelle 2: Auszug der Störaussendungsnormen

➤ nologische Einrichtung“ fällt und entsprechend Norm EN 55022 zuständig ist. Wie wird nun gemessen? Ein Auszug aus der Norm verrät folgendes: Der Prüfling steht wie in Abbildung 6 gezeigt auf dem Messtisch. Er wird über eine AMN (230V@50Hz) gespeist. Die Ethernet-Verbindung vom Prüfling zum AE wird über eine ISN hergestellt. Die AMN und die ISN haben jeweils eine 50Ω Analysator-Auskopplung, um das Störspektrum aufzunehmen.

Für den Prüfling gelten die Grenzwerte der Klasse A, also Geräte, die im industriellen Umfeld ihren Dienst verrichten.

Folgende Grenzwerte sind in der EN 55022 definiert (Tabelle 4 und Tabelle 5).

Gemessen wird die Störstrahlung auf der Stromnetz-Versorgungsleitung mittels einer Störspektrumaufnahme via Spektrumanalysator über die AMN.

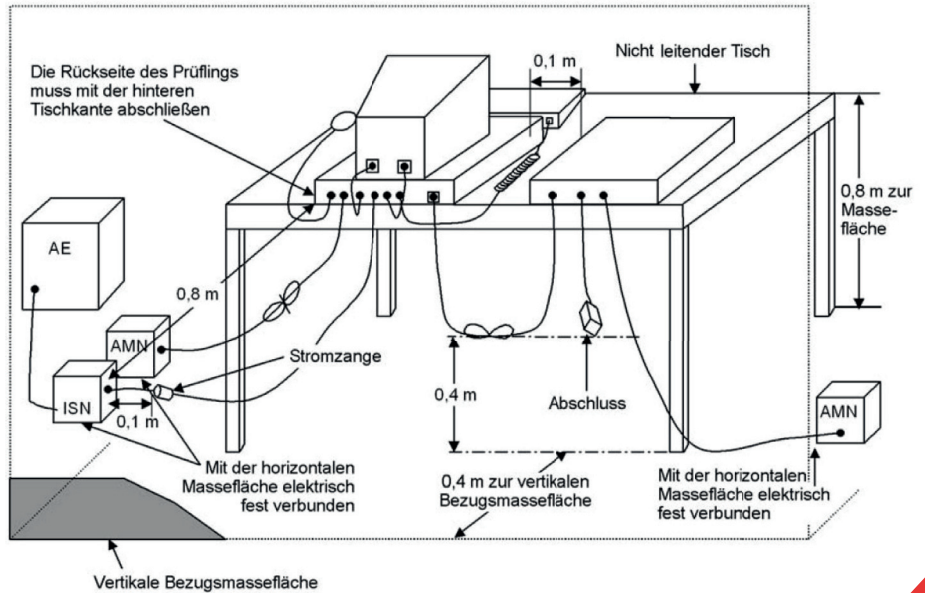


Abb. 6: Prüfaufbau zur Messung der leitungsgebundenen Störstrahlung

Die Störstrahlung auf der Ethernet-Leitung wird über den Spektrumanalysator der ISN aufgenommen. Die aufgenommenen Messwerte werden über einen Entscheidungsbaum bewertet. Diese Aussage gibt an, ob ein Prüfling bestanden hat oder auch nicht. Der Entscheidungsbaum hat folgendes Aussehen (Abb. 7).

Der Quasispitzenwert nach CISPR 16-1-1 ist ein Spitzenwert, der kontinuierliche, schmalbandige Störsignale stärker bewertet als breitbandige, gepulste Signale oder Rauschen. Der Mittelwert ist der arithmetische Mittelwert der Störsignale. Sobald eine Messung auf einer Leitung (Stromversorgungs- oder IT-Leitungen) nicht bestanden ist, darf der jeweilige Prüfling nicht betrieben werden!

Eine tatsächliche Beispiel-Applikation ist in der folgenden Darstellung gezeigt (Abb. 8).

AE	Additional Equipment, Hilfseinrichtung (z.B. Ethernet-Gegenstelle)
AMN	AC Mains Networking, Stromnetz-Emulation mit Spektrumanalysator-Auskopplung zur Messung der Störstrahlung in Richtung Stromnetz
EUT	Equipment Under Test, Prüfling
ISN	Impedance Stabilization Networks, Netzwerk-Emulation mit Spektrumanalysator-Auskopplung zur Messung der Störstrahlung, die über Ethernet aus dem Prüfling ausgesendet wird

Tabelle 3: Verwendete Abkürzungen in Abb. 6

Frequenzbereich (MHz)	Grenzwerte der Störspannung dB (µV)		Grenzwerte des Störstroms dB (µA)	
	Quasispitzenwert	Mittelwert	Quasispitzenwert	Mittelwert
0,15 bis 0,5	97 bis 87	84 bis 74	53 bis 43	40 bis 30
0,5 bis 30	87	74	43	30

Tabelle 5: Grenzwerte Klasse A an den IT-Interfaces wie Ethernet

Frequenzbereich (MHz)	Quasispitzenwert	Mittelwert
0,15 bis 0,50	79 dB (µV)	66 dB (µV)
0,50 bis 30	73 dB (µV)	60 dB (µV)

Tabelle 4: Grenzwerte Klasse A am Stromnetz-Versorgungsanschluss

Beispiel einer nichtbestandenem Strahlungsmessung mit Schaltregler 1

Die Störstrahlungsmessung auf der Stromnetz-Versorgungleitung des Prüflings mit Schaltnetzteil 1 ergab folgendes Bild (Abb. 9). Zuerst werden die QS-Messwerte via Entscheidungsbaum in Abbildung 7 betrachtet. Existiert ein QS-Messwert, der oberhalb des QS-Grenzwertes liegt, hat der Prüfling nicht bestanden. Im obenliegenden Fall ist der QS-Messwert bei ca. 180KHz über Grenzwert und der Prüfling ist damit durchgefallen.

Das entsprechende externe Schaltnetzteil 1 darf mit dem Tablet-PC in Europa nicht betrieben werden.

Die Störstrahlungsmessung auf der Ethernet-Leitung, die sich aus der Störstrahlung des externen Schaltwandlers 1 und der Störstrahlung des Tablet-PCs zusammensetzt, ist in Abbildung 10 zu sehen. Wie zu sehen ist, sind viele QS-Messwerte über der QS-Grenzwertlinie und endet laut Entscheidungsbaum (Abb. 7) in „nicht bestanden“.

Auch aus Sicht des Ethernet-Interface darf der externe Schaltwandler in Verbindung mit dem Tablet-PC in Europa nicht betrieben werden.

Beispiel einer bestandenen Strahlungsmessung mit Schaltregler 2

Da die Zusammensetzung des Prüflings aus Schaltwandler 1 und Tablet-PC die leitungsgebundene Störstrahlprüfung nicht bestanden hat, wurde ein Schaltwandler eines anderen Herstellers mit deutlich günstigeren EMV-Verhalten genutzt (Schaltwandler 2).

Die Störstrahlungsmessung auf der Stromnetz-Versorgungleitung des Prüflings mit Schaltnetzteil 2 ist wie in Abbildung 11 dargestellt.

Alle QS- und MW-Messwerte sind unterhalb des kritischen MW-Grenzwertes. Der Test ist damit laut Entscheidungsbaum (Abbildung 7) bestanden.

Die Störstrahlung auf der Ethernet-Leitung ist ähnlich gering (Abb. 12).

Auch hier sind alle QS- und MW-Messwerte unterhalb des kritischen MW-Grenzwertes. Auch dieser Test ist nach Entscheidungsbaum (Abbildung 7) bestanden.

Der Schaltwandler 2 in Verbindung mit dem Tablet-PC darf in Europa betrieben werden.

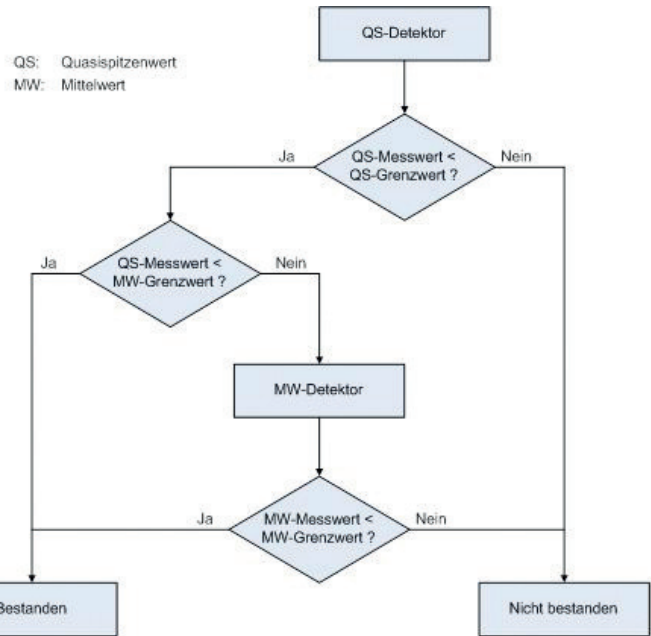


Abb. 7: Entscheidungsbaum für die Messungen

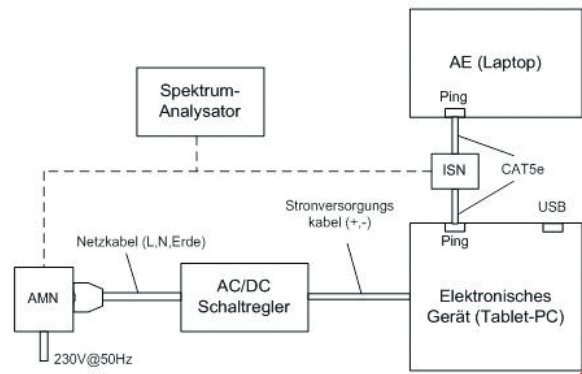


Abb. 8: Beispielapplikation

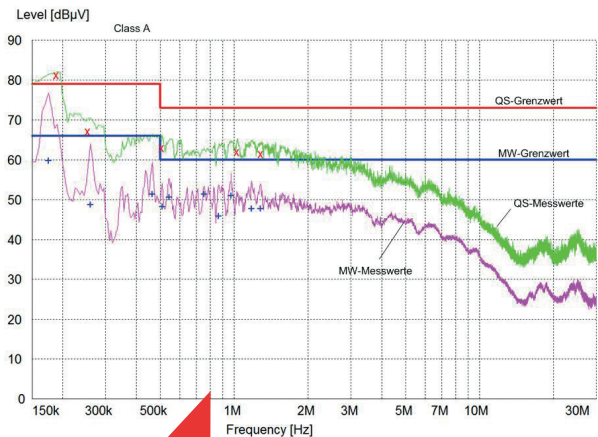


Abb. 9: Leitungsgebundene Störstrahlungsmessung der Stromnetz-Versorgungsleitung

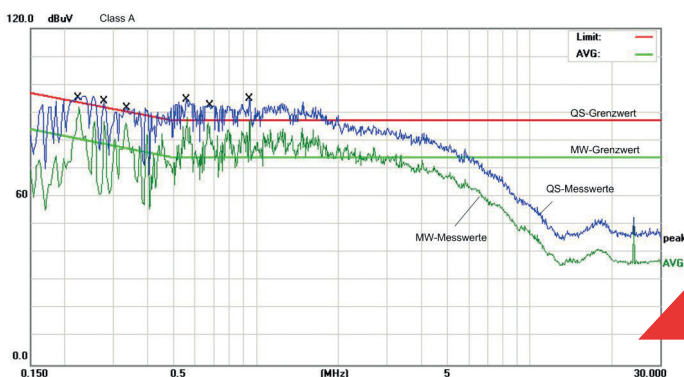


Abb. 10: Leitungsgebundene Störstrahlung auf der Ethernet-Leitung

Mögliche Lösungsansätze

Das Schaltnetzteil 1 erzeugte in Verbindung mit dem Table-PC auf der Netzversorgungsseite und auf der Betriebsspannungsseite eine hohe Störstrahlung, so dass diese Kombination in Europa nicht betrieben werden darf.

Die Primärseite kann in der Regel wie in Abbildung 13 dargestellt werden. Der hoch-frequente Nutzstrom i_{DM} der vom Leiter zum Neutralleiter fließt, führt zu einer Gegentaktstörung. Über den Schutzleiter und einigen parasitären Kapazitäten fließt auch ein Gleichtaktstrom i_{CM} der sich zusätzlich als Störstrom bemerkbar macht. Folgende sinnvolle Möglichkeiten sind anzuwenden:

- Verminderung der parasitären Kapazitäten
- Einsatz eines Netzfilters

Parasitäre Kapazitäten sind stark vom Layout und von den mechanischen Randbedingungen abhängig, auf die während der Designphase geachtet werden kann und muss. Bei einem externen Schaltwandler hat der Designer leider keinen Einfluss darauf.

Ein Netzfilter sollte auf alle Fälle vorhanden sein. Ein solcher ist in Abbildung 14 zu sehen.

Zentrales Bauteil ist die Gleichtakt-Drossel L_V , die in Verbindung mit C_{x1} , $C_{y1/2}$ eine hohe Einfügedämpfung bewirkt. Damit wird der Gleichtaktstrom I_{CM} stark gedämpft. Die Gegentaktstörungen werden jeweils über L_{1a} und C_{y1} oder L_{1b} und C_{y2} gedämpft. Die Werte der Bauteile hängen von der Gleichtakt-Störstromfrequenz ab.

Der Einsatz eines Netzfilters ist zwingend erforderlich und scheint bezüglich der gemessenen Störstrahlung des Schaltreglers 1 entweder fehlangepasst oder schlicht nicht vorhanden zu sein.

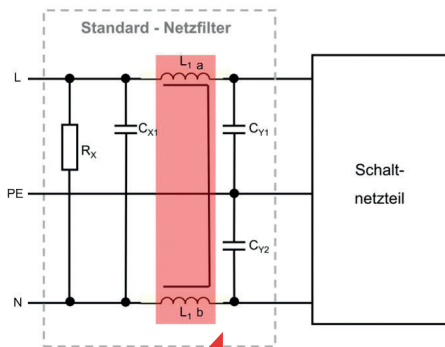


Abb. 14: Netzfilter auf Primärseite

Abb. 13: Störströme am Schaltwandler-eingang

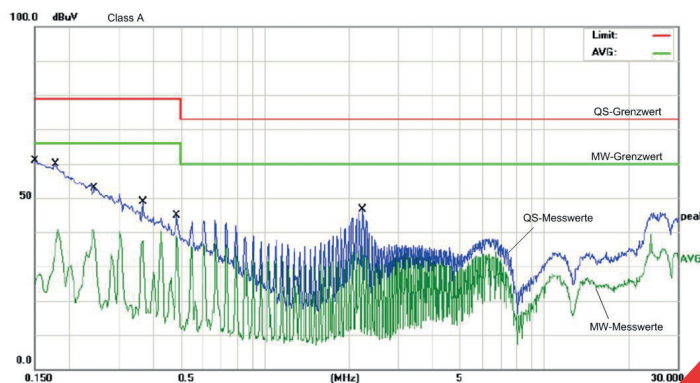


Abb. 12: Leitungsgebundene Störstrahlung auf der Ethernet-Leitung

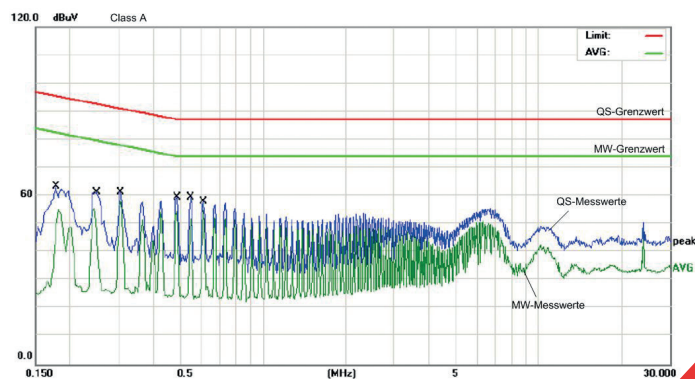
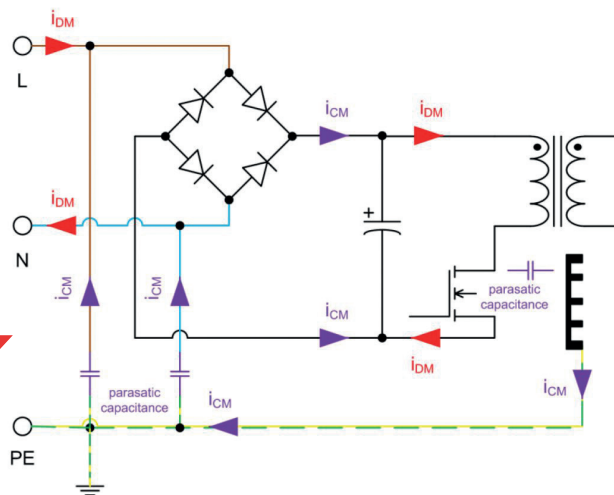


Abb. 11: Leitungsgebundene Störstrahlung auf der Stromversorgung



Sekundärseite: In der Regel tauchen auf der Sekundärseite „nur“ Gegentaktstörungen i_{DM} auf. Ein π -Tiefpass-Filter ist eine gute Wahl, um diese Störungen über die Einfügedämpfung zu begrenzen. Ein solcher Tiefpass 3. Ordnung mit L_1 , C_1 und C_2 hat eine Steilheit von 60db/Dekade und dämpft i_{DM} ausreichend. Die Auslegung der Bauteile ist wie bei den Netzfiltern von der Störfrequenz abhängig.

Ist der Schutzleiter der Primärseite mit dem Massebezug auf der Sekundärseite verbunden, könnte zusätzlich ein Störstromkreis entstehen. Aufgrund dieses Problems und im Sinne einer galvanischen Trennung ist unbedingt darauf zu achten, dass der Schutzleiter nicht mit dem sekundärseitigen Massebezug verbunden ist.

Der Schutzleiter des Schaltwandlers 1 ist mit dem Massebezug verbunden, beim Schaltwandler 2 nicht! Darüber hinaus ist auch beim Schaltregler 1 der sekundärseitige Filter nicht korrekt ausgelegt oder ist anderer Ordnung.

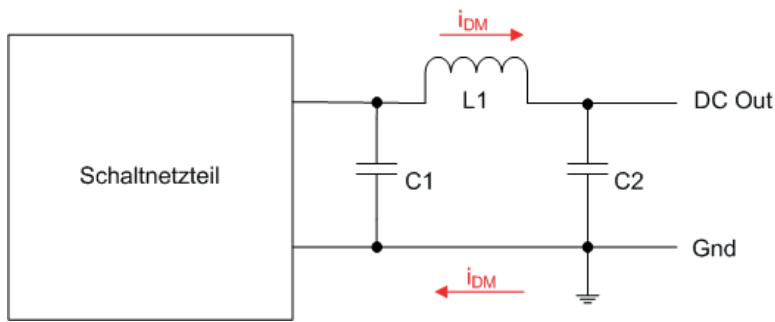


Abb. 15: π -Filter auf der Sekundärseite eines Schaltwandlers

Fazit

Wird ein informationstechnologisches Gerät, das in Deutschland oder in Europa vertrieben werden soll, mit einem externen Schaltnetzteil betrieben, braucht es ein CE-Zeichen und damit eine Störabstrahlung gemäß gesetzlicher Richtlinien.

Das eigene Gerät sollte idealerweise eine Filterschaltung bezüglich der Taktfrequenz des externen Schaltwandlers aufweisen. Da man diese Frequenz erst einmal nicht kennt, ist in erster Näherung von 50KHz-100KHz auszugehen.

Ein ausgesuchter Schaltwandler sollte vorzertifiziert sein und das CE-Zeichen tragen. Soll das eigene Gerät in Verbindung mit dem ausgesuchten Schaltwandler auch in anderen Ländern als Europa vertrieben werden, kommen noch andere Zertifizierungen dazu (In-Country-Certification). Fehlt eine, muss sie nachher teuer durchgeführt werden.

Die Filtereigenschaften sind mit dem Hersteller des Wandlers abzuklären. Häufig testen die Schaltwandler-Hersteller ihre Netzteile mit den für sie günstigen Bedingungen, um das CE-Zeichen zu bekommen. Jedes angeschlossene Gerät weist eine unterschiedliche Eingangsimpedanz auf und damit ist klar, dass manche Kombinationen aus Gerät und Netzteil nicht zielführend sind, wie das obige Beispiel zeigt. Um sicherzustellen, dass während der Zertifizierung keine unerwarteten Probleme auftauchen, sollte das gesamte Produkt aus Schaltwandler und Gerät frühzeitig im Labor durchgemessen werden (Pre-Compliance-Tests).

Nicht zuletzt ist die Auswahl eines externen Schaltwandlers vom Preis abhängig. In der Entwicklungsphase eines Geräts sollte der Preis nicht außer Acht gelassen werden. Er sollte aber nicht kompromisslos zu einem strahlenden Netzteil führen, das am Ende der Entwicklung hin zum Produkt ein Show-Stopper werden könnte.

Quellenangaben

- [1] European Standards EN 55022:2007-04 (Abb. 6, Abb. 7, Tabelle 4, Tabelle 5)
- [2] EMV-Seminar 2013 von Würth Elektronik (Abb. 5, Abb. 13)
- [3] Netzfilter – Die letzte Hürde im Schaltnetzteil, Application Note
- [4] Labormessergebnisse von Eurofins Product Service GmbH
- [5] Labormessergebnisse von CETECOM GmbH

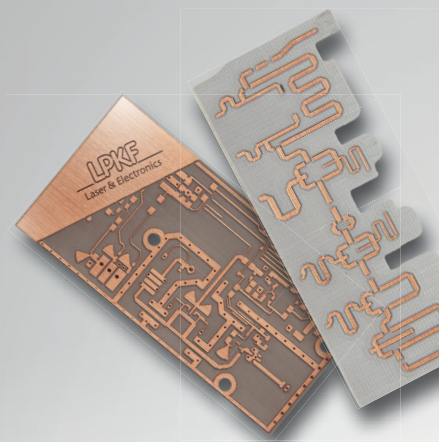
Anzeige



Schneller fertig als gedacht

PCB-Prototypen in nur einem Tag mit LPKF ProtoMaten. Noch einfacher – und automatisch – produzieren.

www.lpkf.de/prototyping



LPKF

Laser & Electronics

Gerber

Vergangenheit, Gegenwart und Zukunft

Olaf Davidsmeyer <olaf.davidsmeyer@ec-aachen.de>

Mehr als 95% aller weltweit produzierten Leiterplatten-Designs werden als Gerber-Daten vom Entwickler zum Produzenten übermittelt. Die meisten CAD Systeme geben automatisch Gerber-Daten aus. Nur selten müssen sich die Entwickler damit befassen, wie eine Gerber-Datei ihre Daten repräsentiert. Dies ist ein gutes Zeichen in Bezug auf die Leistungsfähigkeit und Verbreitung des Formats, aber gelegentlich treten auch Probleme auf, für die ein wenig Hintergrundwissen hilfreich wäre. Zudem werden für das Format Entwicklungen geplant, für die Hintergrundwissen zukünftig um so nützlicher wäre.

Gerber Vergangenheit

Joe Gerber (1924 – 1996) war ein US-amerikanischer Erfinder, der 1940 aus Österreich in die USA floh. Schon als Student war er am akkuraten Plotten von Daten interessiert. Während der 50er Jahre entwickelte er den digitalen XY-Koordinaten-Tisch, welcher den Kern seines zukünftigen Geschäfts, Gerber Scientific, bildete. Das erste Produkt, welches er unter Verwendung des neuen Tisches herausbrachte, war eine der ersten digitalen Zeichenmaschinen weltweit. Zu den späteren Produkten gehörte eine automatische Zuschnittmaschine für Stoffe, welche in der Bekleidungsindustrie immer noch stark verbreitet ist. In den 80er Jahren entwickelte er computergestütztes Zubehör zur maschinellen Herstellung von Brillengläsern, welches ebenfalls noch heute eingesetzt wird.

In den 60er Jahren fand Joe Gerber eine weitere Verwendung für seinen XY-Tisch. Er führte den weltweit ersten numerisch gesteuerten Photoplotter ein, um Fotowerkzeuge (Filme) zur Produktion von Leiterplatten zu erzeugen. Es funktionierte, indem zunächst ein optischer Kopf mit einer Lichtquelle zur korrekten Stelle über den Film auf dem Plottertisch bewegt wurde. Ein kreisrundes Rad mit unterschiedlichen geformten/großen Löchern darin (Blenden), wurde dann gedreht, bis sich die richtige Blende unter der Lichtquelle befand. Für ein Pad wurde die Lichtquelle kurz ein und ausgeschaltet (geblitzt) und dieses damit auf den Film belichtet. Eine Leiterbahn wurde erzeugt, indem die Lichtquelle eingeschaltet war, während der Kopf sich bewegte und damit die Bahn auf den Film "gezeichnet" wurde. Aus diesem Grund sprechen wir noch heute von "Blendentellern" (aperture tables) und seltener von flashes ("geblitzt") und draws ("ge-

zeichnet"). Die Plotter waren als Vektor Plotter bekannt, weil der Kopf den jeweiligen Muster der Leiterplatte folgte. Das passende Datenformat basierte auf einem bestehenden Format RS-274-D, welches von der US Electronic Association (EIA) entwickelt wurde, um damit jegliche NC Maschine zu steuern. Die ersten dieser Gerber Photoplotter luden die Daten über Lochkarten.

RS-274-D wird zu RS 274x

In den frühen 80er Jahren wurden CAD (Computer-aided Design) Systeme für Leiterplatten populärer und ersetzten die alten handgeklebten 2:1 Reprovorlagen. CAD Systeme konnten die Plottedaten direkt an den Plotter ausgeben, um die Fotowerkzeuge (Filme) zu erstellen. Zu diesem Zeitpunkt waren die meisten Fotoplotter Gerber Plotter. Andere Anbieter drängten in den Fotoplotter-Markt, aber da Gerber 1980 die vollständige Spezifikation zu ihrem Format veröffentlicht hatte, wurde Gerber RS-274-D der de facto Standard.

Als Behelf beim Transfer der Leiterplattenbilder, hatte das Format eine kritische Einschränkung: die Größe, Form und Anzahl der Blenden war durch das physische Blendenrad beschränkt. Für Designs mit konventionellen Durchsteck-Bauteilen mit runden oder quadratischen Pads funktionierte dies (mehr oder weniger) gut, aber das Format konnte nicht mit den neuen surface-mount (auf der Oberfläche angebrachten) Bauteilen umgehen, welche eine große Auswahl hauptsächlich rechteckiger Pad-Größen nutzte. Mit RS-274-D war die einzige Möglichkeit die Pads mit winzigen "draws" zu "zeichnen". Ähnlich konnte eine einfache Massefläche umgekehrt geplottet werden; soll heißen, die Freistellungs-Löcher

in der Fläche werden schwarz geplottet und der Leiterplattenhersteller dreht die Polarität entweder in seinem CAM-System um, oder indem er einen physischen Kontaktabdruck macht. Aber dies funktionierte nicht für gemischte Masseflächen oder Masseflächen auf Signallagen. Diese mussten alle mit "draws" gezeichnet werden. Ein großes Bild mit SMDs und Masseflächen konnte bei einem solchen Plotter bis zu 24 Stunden für den Plot beanspruchen.

Die Lösung war ein neuer Typus von Fotoplotter und ein neues Datenformat. Der Raster-Fotoplotter verwendete eine Lichtquelle, typischerweise einen Laser, um den Film in einem kontinuierlichen Muster zu rastern. Das Bild wurde dabei in einer Sequenz von "Laser ein"- und "Laser aus"-Befehlen aufgebaut. Jetzt konnte jedwede Form aus Rasterpixeln aufgebaut und geplottet werden. Heute ist dies ein Standard Industrie-Werkzeug, um Leiterplatten-Filme herzustellen. Die Laserplotter nutzen bis zu 48 unabhängig geschaltete, gleichzeitige Strahlen und plotten mit einer Auflösung von bis zu 50.000 DPI, oder mehr.

Jetzt war es möglich das Gerber-Format flexibler zu gestalten und and die Bedürfnisse der Leiterplatten-Entwickler anzupassen. RS-274X oder Extended Gerber wurde 1991 herausgebracht. Es erlaubte dem Nutzer jegliche Form als Pad, Leiterbahn oder Polygon (Massefläche) zu definieren. Die Blenden-Definitionen hingen nicht länger vom physischen Blendenrad ab, so dass sie automatisch aus dem CAD Job abgeleitet werden und als Teil der Daten-Ausgabe ausgegeben werden konnten.

Gerber Heute

RS-274X ist heute das Standard-Übertragungsformat für Leiterplatten-Daten. Es ist klar, eindeutig und - falls irgendwelche Fragen aufkommen - durch den Menschen/Bediener ohne Hilfsmittel lesbar. Jede Datei ist vollständig und erlaubt Ihnen jede Pad-Form oder Kupferfläche zu zeichnen die Sie wollen.

Das alte Standard Gerber RS-274-D besteht trotz seiner Nachteile fort. Es ist sehr eingeschränkt, benötigt einen separaten Blendenteller, welcher scheinbar oft verloren geht; es produziert große, unhandliche Dateien; die Ausgabe kann das Verschmelzen positiver und negativer Bilder erfordern, welche im besten Fall umfangreiche Aufräumarbeiten erfordern und im schlechtesten Fall schwer zu findende Fehler bedeuten.

Eurocircuits akzeptiert das alte Format weiterhin, soweit benötigt (z.B. für alte Jobs), obwohl es nicht mit PCB Visualizer® funktioniert. Wie auch immer, Extended Gerber RS-274X ist unser bevorzugtes Datenformat und hat keine der Einschränkungen von RS-274-D. Jede Datei ist vollständig mit eingebetteten Blenden-Definitionen, funktioniert mit PCB Visualizer® und bietet Ihnen dabei sämtliche Vorteile unserer fortschrittlichen Datenanalyse-Technologie. Alle aktuellen und die meisten älteren CAD-Systeme sind in der Lage RS-274X auszugeben. Falls Ihr CAD System immer noch das alte RS-274-D Format ausgibt, sehen Sie bitte in die Ausgabeeinstellungen. Manchmal kann die Ausgabe auf RS-274X umgestellt werden. Auf verschiedenen Systemen können unterschiedliche Terminologien verwendet werden. Fragen Sie im Zweifel einfach bei uns nach, oder werfen Sie einen Blick in unsere PCB Design Guidelines für mehr Ratschläge zu Eingabeformaten.

Gerber Zukunft

Extended Gerber RS-274X, bietet ein exaktes und eindeutiges Bild der Lagen einer Leiterplatte. Dennoch fehlen noch notwendige Lagen-Informationen für die Produktion (insbesondere für die automatische Daten-Aufbereitung), welche nicht im Format vorgesehen sind.

Folgende Beispiele:

- Welche Funktion hat die Lage: Top Kupfer, Top Lötstopmaske, etc.?
- Zeigt das Bild eine einfache Leiterplatte, oder einen Nutzen?
- Welche Funktion besitzt dieses Objekt: ist es ein SMD Pad oder Via Pad, Fiducial etc.?
- Was ist die Kontur/das Profil? Automatische Erkennungs-Software wie PCB Visualizer® kann rechteckige Konturen erkennen, aber keine komplexen Formen.
- Welche Bohrtoleranzen hat dieses Loch? Beispielsweise könnte es ein Loch für Einpresstechnik sein.
- Welche Leiterbahnen sind Impedanz-kontrolliert?
- Welche Vias müssen gefüllt werden?

Der nächste Schritt ist die Einbeziehung dieser Informationen in das Datenübermittlungsformat. Jedwede Erweiterung des Datenformats, muss mit dem existierenden Format und den bestehenden CAD-Systemen kompatibel sein. Obwohl andere Formate vorgeschlagen wurden, die nicht-Bildinformationen einschliessen können, ist Gerber so weit verbreitet und effektiv im Einsatz, dass es - ebenso wie QWERTY, QWERTZ and AZERTY Tastaturen - nicht leicht zu ersetzen ist.

Heute wird das Gerber-Format von der belgischen Firma UCAMCO, die diese Abteilung von Gerber Scientific 1997 gekauft hat, gewartet und entwickelt. UCAMCO haben kürzlich eine Vorlage für die nächste Generation von RS-274X, Gerber RS-274X2 veröffentlicht. Diese führt Eigenschaften in das Format ein, welche die oben genannten Informationen übermitteln.

Diese neue Entwicklung wird in unserem technischen Blog zu Gerber X2 näher beschrieben. Eurocircuits arbeitet eng mit UCAMCO am neuen Format zusammen, und unterstützt UCAMCOs Antrieb bessere Werkzeuge für die europäische und globale Gemeinschaft der Leiterplatten-Entwickler zu bieten. Da die neuen Attribute in die CAD Systeme implementiert werden, werden wir neue Funktionen zu unserer Dateneingabe hinzufügen, sowie Prozeduren zur Überprüfung, um mit diesen umzugehen. Selbstverständlich werden wir auch weiterhin die älteren Gerber-Formate akzeptieren.

```
D11*
X1785250Y2173980D02*
X1796650Y2177730D01*
X1785250Y2181480D01*
X1796650Y2184580D01*
D12*
X3421095Y1407208D03*
X3422388Y1406150D03*
M02*
```

Listing 1: Beispielhafte Gerber-Datei

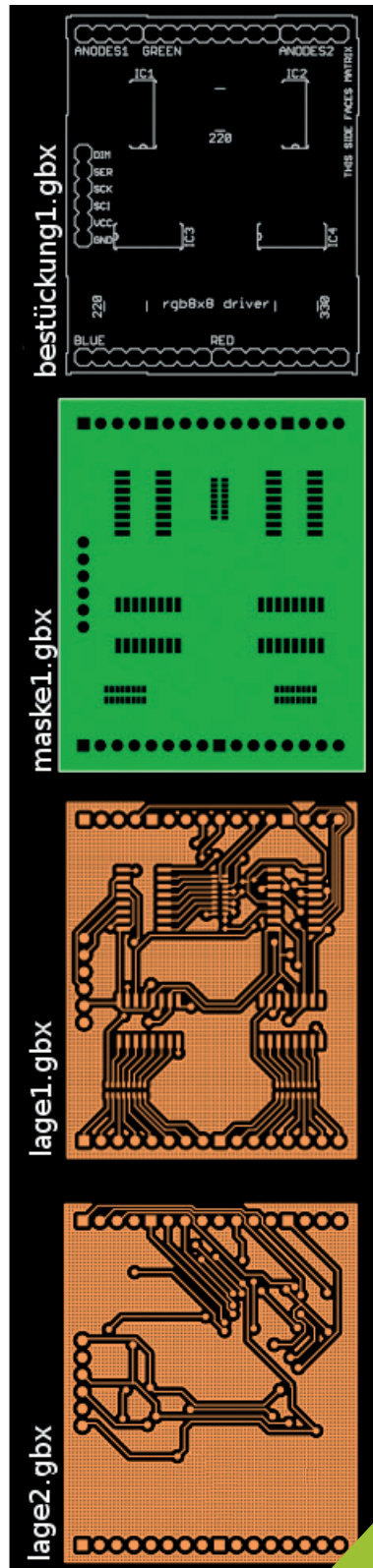


Abb. 1: Vier Layer (Gerberformat) Quelle: <http://de.wikipedia.org/wiki/Gerber-Format>

BGAs mit Eagle

Design Rules & Platinenherstellung

Benedikt Sauter <sauter@embedded-projects.net>

Immer seltener kommt man bei Platinenlayouts um BGA-Bausteine herum. In diesem kleinen Artikel möchte ich ein paar Tips über die Besonderheiten für Eagle und die zugehörige Herstellung der Platinen durch die Standardhersteller geben. Beginnt man die ersten BGA- bzw. Fine-Pitch Layouts mit Eagle zu zeichnen, kann es schnell dazu kommen, das man voller Stolz über die versendete Bestellung schnell durch eine E-Mail vom Platinenhersteller einen Schritt im Projektverlauf zurückgeholt wird. Der Hersteller meldet: Die Platine kann so leider nicht bzw. nur gegen einen erheblichen Aufpreis produziert werden. Normalerweise startet jetzt ein Ping-Pong Spiel zwischen Hersteller und Entwickler. Als Entwickler versteht man die jedoch nicht, denn man hat eigentlich sehr gut auf Leitbahnbreiten, Isolationsabstände und Bohrungen geachtet. Was genau passt denn nicht? Bzw. wie muss man arbeiten das es passt? Wir haben einen sehr guten Kontakt zu Cadsoft (Eagle) und Eurocircuits (vielen Dank nochmal an Richard und Uwe!) und hatten so die Chance nach und nach herauszufinden, was genau das Problem dabei eigentlich ist. Ich hoffe, dass mit diesem Artikel der eine oder andere Punkt erklärt und so vermieden werden kann.

Ausgangslage

Nach dem man den Schaltplan für das Layout freigeben hat, schaut man typischerweise bei den Platinenproduzenten nach, was für Strukturen hergestellt werden können. Zum Beispiel (Abb. 1):

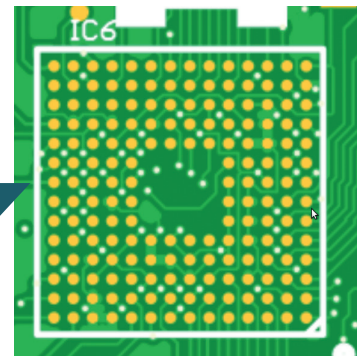
Breite: 0.125 Durchmesser: auto Bohrung: 0.2

Abb. 1: Struktureinstellungen

- 150 µm Strukturbreite und 0,3mm der kleinste Bohrung
- oder für etwas mehr Geld aber auch bezahlbar 125 µm und 0,2 Bohrung

Damit lässt sich eigentlich viel realisieren. Bei einem BGA Baustein mit einem Pitch (Abstand von Ball zu Ball) von 0,8 mm muss man bei einer Bahnbreite von 125 µm diesen Wert mal 3 nehmen (Isolation + Leiterbahn + Isolation) 375 µm. Ein Ball hat meist einen Durchmesser von ca. 300 – 400 µm. D.h. man kriegt eine Leitung zwischen zwei Balls durch.

Abb. 2: BGA (Online PCB Visualizer®)



Wenn man eine Bohrung mit einem Durchmesser von 0,2mm nimmt, sieht es im Eagle auch so aus, als ob man die 0,125 um die Durchkontaktierung ebenfalls einhält. Wenn man jetzt die Leitungen entsprechend auf die Werte einstellt und schaut, dass man schön in der Mitte von allem bleibt, sieht es so aus, als ob es perfekt passen müsste...

Eigentlich hat man als Entwickler ja sehr diszipliniert die Leitungen so gut wie möglich zwischen all den Balls und Durchkontaktierungen verlegt. Wirft man jetzt den Design-Rule-Check an, wird man etwas besserem belehrt, denn die Fehlerliste explodiert beinahe. Motiviert beginnt man am Anfang noch die Liste nach und nach abzarbeiten, doch schnell merkt man, dass kriegt man alles gar nicht richtig weg.

Schaut man aber selber die Stellen an und misst diese kurz nach, so sieht man, dass es eigentlich ganz gut passen würde. OK ab und an hat man vielleicht nicht ganz den gewünschten Abstand, aber solange das noch weit unter 10% von den Grenzwerten entfernt ist denkt man sich, dass es schon gehen wird.

Nicht produzierbar: Es sind einfach zu viele Abstände unter dem Mindestmaß von z.B. 0,125mm. Und hier unterscheiden sich die Platinenhersteller. Es gibt manche, bei denen man sagen kann, OK mach auf mein Risiko weiter, oder die anderen die sagen: „Nein du musst die teurere Klasse verwenden“.

Es stellt sich nun die Frage, wie man das Layout ohne diese Fehlermeldungen hin kriegt bzw. wie man den Wechsel in eine teure Technologieklasse vermeidet?

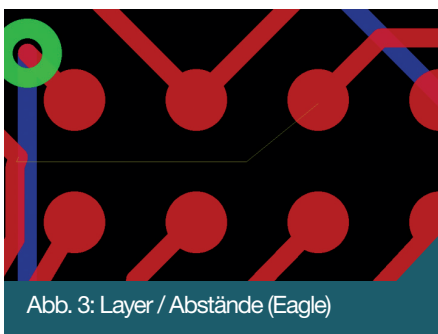


Abb. 3: Layer / Abstände (Eagle)

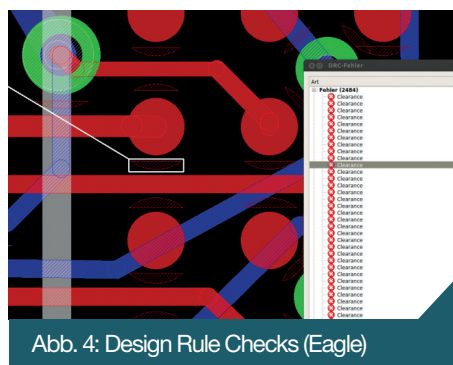


Abb. 4: Design Rule Checks (Eagle)

Die Abstandsfehler

Es sind einfach noch viele Abstände auf dem Board, die kleiner als der minimale Abstand sein dürfen.

Jeder Platinenhersteller bietet Design-Rules an, wobei die man die auch mit dem Wissen von der Homepage gut selber einstellen kann.

- Leiterbahnbreiten 0,125 mm und Isolationen ebenfalls
- Minimale Bohrung 0,2 mm

Abhängig vom eingestellten Raster kann man jetzt schnell immer ein paar 0,0xx – 0,xx mm daneben liegen. Daher kommt Tip Nummer 1.

TIP 1: Beim Layouten das Raster passend zum BGA einstellen

Die Abstandsverletzungen kann man ganz einfach in den Griff bekommen, wenn man beim Layouten das Raster

passend zum BGA einstellt. Also am Besten ist es den BGA (mm-gerastert) in ein mm-Raster zu platzieren. Das kann man mit Strg + MOVE: Raster einstellen und dann auf den BGA geklickt. Dann springt er in das aktuelle Raster. Nun braucht man ein Raster mit dem man einfach zwischen den BGA-Anschlüssen durchkommt.

TIP 2: Ein günstiger Wert entspricht dem halben/viertel/achtel Abstand

Ein günstiger Wert entspricht dem halben/ viertel/achtel Abstand zwischen den Lötflächen. Wenn man die Verbindungen auf einem alten Layout etwas verschiebt, kann man so meist alle Abstandsfehler beseitigen.

Ein paar Abstandsverletzungen bleiben z.T. übrig, diese sind aber kein Problem für die Fertigung, da es eben nur minimale Unterschreitungen sind.

Category	Value	Violation
Galvanischer Aufbau		
Bestell-Details	STANDARD pool	
Leiterplatten Eigenschaften		
Technologie		
Leiterbahnbreite	8	10
Leiterbahnbreite Aussenlagen (CL-TW)	0.100 mm	0.100 mm
Isolationsabstand Aussenlage (CL-TT-TP-PP)	0.100 mm	0.072 mm
Kupfer top	0.100 mm	0.085 mm
Kupfer bottom	0.100 mm	0.072 mm
Aussenlagenreizing (OAR)	0.100 mm	0.072 mm
Kupfer top	0.100 mm	0.072 mm
Kupfer bottom	0.100 mm	0.072 mm
Leiterbahnbreite Innenlagen (IL-TW)	0.100 mm	0.125 mm
Isolationsabstand Innenlage (L-TT-TP-PP)	0.100 mm	0.067 mm
Innenkupfer 1	0.100 mm	0.083 mm
Innenkupfer 2	0.100 mm	0.067 mm
Innenlagenreizing (IRI)	0.125 mm	0.072 mm
Innenkupfer 1	0.125 mm	0.072 mm
Innenkupfer 2	0.125 mm	0.072 mm
Bohrklasse	D	D
Kleinster Endbohrdurchmesser	0.15 mm	0.20 mm
Minimaler Loch zu Loch Abstand	0.15 mm	0.239 mm
Überlagerte Bohrungen	0	0
vergrabene Bohrungen	0	1
Bohrdicke	< 1000/dm ²	411/dm ²
Erweiterte Optionen		
Oberfläche		
Material		

Abb. 5: Typisches Ergebnis, welches sich aber durch Verwendung der richtigen Raster vermeiden lässt.

Die Durchkontaktierungen

Die Vias (Durchkontaktierungen) passen auch, wenn man die Bohrdurchmesser auf 0.2mm und den Gesamtdurchmesser auf 0.4mm setzt.

TIP 3: Bohrung auf 0.2mm setzen und den Gesamtdurchmesser auf 0.4mm bzw. 0.45 auf Innenlagen (und nicht auto wie voreingestellt)

Wichtig ist dieser Gesamtdurchmesser! Pro Via kann man das angeben. Standardmäßig hatten wir hier immer auto eingestellt.

Abb. 6: Stellt man 0,4 bzw. 0,45 als Durchmesser und 0,2 als Bohrung ein passt alles. (Screenshot von Eagle)

Online Prüfung und Produktion

Die Screenshots sind mit dem Online Tool PCB Visualizer der Firma Eurocircuits gemacht worden. Dort kann man eine Eagle oder Gerberdatei hochladen und dann direkt online im Browser sehen was passt oder nicht.

Per Browser kann man sich durch die Platine, wie in einem Layoutprogramm, klicken und entsprechend alles prüfen.

Das Tool findet man, im Login Bereich bei Eurocircuits. Dieses kann man auch ohne etwas kaufen zu wollen verwenden. Gerne nehme ich das auch für die Kontrolle von Gerberdaten her. Ohne Fehler sieht das dann wie in Abbildung 7 aus.

Nun sollte eine Produktion schon wesentlich besser funktionieren. Viel anders ist das Layout jetzt nicht aber eben produzierbar.

Danke an Uwe Dörr von Eurocircuits und Richard Hammerl von Cadsoft!

Name	Required	Measured
Import		
Imported 23 layers		
Plating		
Top copper density		0.76
Top plating index	> 0.4	0.94
Bottom copper density		0.85
Bottom plating index	> 0.4	0.95
Order details	STANDARD pool	
Board definition		
Board name	test	
Number of layers	4	4
Delivery term	7	
Delivery format	Single PCB	
Board quantity	11	
PCB width (X)	75.00 mm	75.00 mm
PCB height (Y)	54.00 mm	54.00 mm
Technology		
Pattern class	9	9
Drill class	E	D
Advanced options		
Treatment		
Top soldermask	Green	Detected
Bottom soldermask	Green	Detected
Top legend	White	Detected
Bottom legend	White	Detected
Surface finish	Che Ni/Au selectif	
Bare Board Testing	Yes	

Abb. 7: Keine Fehler (Screenshot Online PCB Visualizer® von Eurocircuits)

PowerPi: Funksteckdose per RaspberryPi

Tool zum Steuern von 434Mhz Steckdosen und Gpio's

Anton Hammerschmidt <hammerschmidt@embedded-projects.net>

Der Raspberry Pi

Das Raspberry Pi ist ein von der gleichnamigen Raspberry Pi Foundation in Großbritannien gefertigter Mikrocontroller. Ziel der Stiftung ist es, Schülern und Studenten einen preiswerten Einstieg in die Embedded Linux-Entwicklung zu ermöglichen. Vor allem durch die vielfältigen Anschlussmöglichkeiten und der relativ hohen Leistung eignet es sich für Projekte aller Art. Das Raspberry Pi ist in zwei Ausführungen erhältlich. Das Modell A ist im Gegensatz zum Modell B mit nur 256 MB RAM, einer USB-Buchse und ohne Ethernetanschlüsse ausgestattet. Laut Herstellerangaben verbraucht das A-Modell dafür nur 1/3 Energie der B-Version. Um einen Preis von ca. 35 € zu ermöglichen wird das Raspberry Pi ohne jegliches Zubehör verkauft. Für die Inbetriebnahme muss deshalb ein 5V MicroUSB-Netzteil und eine mindestens 4GB große SD-Karte nachbestellt werden. Das Board mitsamt Zubehör ist bei Farnell sowie Amazon bestellbar. Wie jedes Embedded Linux-Board benötigt auch das Raspberry Pi einige anfängliche Schritte zur Inbetriebnahme.

- Wahl der Distribution
- MicroSD-Karte vorbereiten
- Anfängliches Setup

Die Wahl der Distribution in diesem Projekt fiel auf Raspbian [1]. Raspbian ist die empfohlene Standarddistribution für das Raspberry Pi. Es basiert auf Debian und gilt als sicher, stabil und sehr gut optimiert.

Nun muss das neueste Raspbian Image heruntergeladen und entpackt werden.

```
wget http://downloads.raspberrypi.org/download.php?file=/images/raspbian/2013-02-09-wheezy-raspbian/2013-02-09-wheezy-raspbian.zip
```

```
unzip 2013-02-09-wheezy-raspbian.zip
```

Nun eine leere SD-Karte (mindestens 4GB) in den Kartenleser schieben.

```
df -h
```

```
umount /dev/sdd1
```

Ist die SD-Karte mit dem Tool ‚df‘ identifiziert (z.B.: /dev/sda1), dann muss diese wieder ausgehängt werden.

```
dd bs=1M if=2013-02-09-wheezy-raspbian.img of=/dev/sdd1
```

```
sync
```

Mit ‚dd‘ wird nun das Raspbian Image auf die SD-Karte kopiert. Der Befehl ‚sync‘ schließt alle Schreibvorgänge ab.

Da beim Raspbian SSH erst nachträglich aktiviert werden muss, wird das Raspberry Pi mit einem Bildschirm sowie einer USB-Tastatur verbunden. Nachdem das Netzteil mit dem Board verbunden und der Bootvorgang abgeschlossen ist, wird das Raspbian Setup - auch ‚raspi-config‘ genannt - sichtbar. Hier können elementare Einstellungen wie Zeitzone, Root-Passwort und Prozessorgeschwindigkeit eingestellt werden. Um nicht jedes mal den Bildschirm anzuschließen wird SSH aktiviert.

```
expand_rootfs  Expand root partition to fill disk
overscan       Change overscan
configure_keyboard  Set keyboard layout
change_pass    Change password for 'pi'
change_locale  Set locale
change_timezone Set timezone
memory_split   Change memory split
overclock      Configure overclocking
ssh            Enable or disable ssh server
boot_behaviour Start desktop on boot
```

Abb. 1: Enable ssh server

Anzeige



FRONTPLATTEN & GEHÄUSE

Kostengünstige Einzelstücke und Kleinserien

Individuelle Frontplatten können mit dem Frontplatten Designer mühelos gestaltet werden. Der Frontplatten Designer wird kostenlos im Internet oder auf CD zur Verfügung gestellt.

- Automatische Preisberechnung
- Lieferung innerhalb von 5-8 Tagen
- 24-Stunden-Service bei Bedarf

Preisbeispiel: 34,93 € zzgl. USt./Versand



Nginx

Nginx ist ein sehr leistungsfähiger und gleichzeitig speicher-sparender Webserver. Im Gegensatz zu Apache werden Anfragen eventbasiert abgearbeitet und erzeugen nur wenig Overhead. Somit eignet er sich hervorragend für embedded Linux Systeme. Neben Nginx wird noch die PHP-Laufzeitumgebung und aus sicherheitstechnischen Gründen ein Benutzer und eine Gruppe namens ‚www-data‘ benötigt.

```
apt-get update
apt-get upgrade
sudo apt-get install nginx php5-fpm php5-cgi
php5-cli php5-common
sudo useradd www-data
sudo groupadd www-data
sudo usermod -g www-data www-data
sudo mkdir /var/www
sudo chmod 775 /var/www -R
sudo chown www-data:www-data /var/www
```

Damit der Server richtig funktioniert müssen einige Standard-einstellungen in der Nginx-Konfigurationsdatei vorgenommen werden.

```
sudo vi /etc/nginx/sites-enabled/default
```

```
server {
    listen 80;
    root /var/www;
    index index.html index.php;
    server_name localhost;
    location / {
        try_files $uri $uri/ /index.php?$args;
    }
    location ~ /\.php$ {
        try_files $uri =404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Listing 1: Inhalt der Nginx Konfigurationsdatei

Nun können HTML-Dateien und PHP-Skripte nach ‚/var/www/‘ kopiert werden und sind im Browser unter ‚http://localhost/‘ erreichbar.

PowerPi

PowerPi ist ein aus einer Weboberfläche und einer Serveranwendung bestehendes Programm zum Anlegen und Steuern von 434Mhz Funksteckdosen sowie Gpio's für das Raspberry Pi (Rev2). Funktionen und Merkmale:

- Konfiguration und Steuerung des Servers über eine Webanwendung (nur Webserver benötigt)
- Steuerung von 434 Mhz Steckdosen (z.B.: Elro AB440S [2])
- Steuerung der Raspberry Pi Gpio's
- Auslösen von zeitbasierten Ereignissen
- Webanwendung im Responsive Design

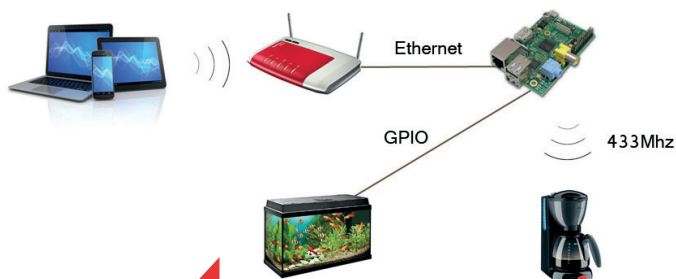


Abb. 2: Übersicht des Projektes

WiringPi [3] ist eine in C/C++ geschriebene Zugriffs-API für die BCM2835 CPU. Sie ermöglicht das einfache Ansteuern der Raspberry Pi GPIO's. Neben einer C/C++ Api gibt es noch einen Wrapper für Python. Nach der Installation ist der WiringPi-Header systemweit verfügbar und kann in eigenen Projekten benutzt werden.

```
tar xzf wiringPi-27afc01.tar.gz
```

```
cd wiringPi-27afc01
```

```
./build
```

PowerPi installieren [4]:

```
tar xzf powerpi-0.1.tar.gz
```

```
cd powerpi-0.1
```

```
sudo make clean
```

```
sudo make
```

```
sudo make install
```

```
sudo /etc/init.d/powerpi start
```

Webserver installieren (Nginx, LightHttp oder Apache) und den Inhalt von ‚www‘ zur Webserverumgebung kopieren (meist /var/www). Im Makefile sind drei Ziele definiert. Das erste Ziel ‚powerpi‘ kompiliert die Binärdatei und speichert diese unter ‚./bin‘ ab.

Das Ziel ‚install‘ kopiert die Vorlage der Konfigurationsdatei (./src/settings) nach ‚/etc/defaults/powerpi‘ und die Binärdatei ‚./bin/powerpi‘ nach ‚./bin/powerpi‘. Ausserdem wird das Start-/Stopskript (./init/powerpi) nach ‚/etc/init.d/powerpi‘ kopiert und im init-System per ‚update-rc.d powerpi defaults‘ registriert.

Das Ziel ‚clean‘ löscht alle Binärdateien und macht die Aktionen des Ziels ‚install‘ wieder rückgängig. Das Init-Skript („/etc/init.d/powerpi“) wird beim Bootvorgang aufgerufen und startet die Serveranwendung. Hierbei wurde nicht die Init-Skriptvorlage („/etc/init.d/skeleton“) verwendet, da die Serveranwendung nicht dem Standard entspricht. Stattdessen wurden die Initziele ‚start‘ und ‚stop‘ wie folgt implementiert:

```
do_start() {
    echo „Starting powerpi
server: “
    /bin/powerpi &
    log_success_msg „done“
    return 0
}
```

Listing 2: Startskript

```
do_stop() {
    echo „Stopping powerpi
server: “
    pkill powerpi > /dev/null
    log_success_msg „done“
    return 0
}
```

Listing 3: Stopskript

Hardware

Die Ansteuerung der Funksteckdosen erfolgt über einen vorgefertigten 434 Mhz Funksender. Der aufgelötete Oszillator versendet die von dem Daten-Gpio empfangenen und hochmodulierten Signale. Dabei muss ein genauestens vordefiniertes proprietäres Übertragungsprotokoll der Funksteckdosenhersteller (hier Elro) eingehalten werden. Durch Reverse Engineering [5] konnten die zu sendenden Bitfolgen isoliert werden.



Abb. 3: Die Hardware

Der Funksender TWS-BS-3 [6] der Firma Wenshing erreicht Datenraten von 2400 bzw. 4800 bps und sendet laut Datenblatt bis zu 150m weit. Dies sollte selbst für ein großes Haus ausreichen. Der Funksender wird wie in Abbildung 4 gezeigt angeschlossen. Dabei kann der Datenkanal mit einem beliebigen freien Gpio verbunden werden.

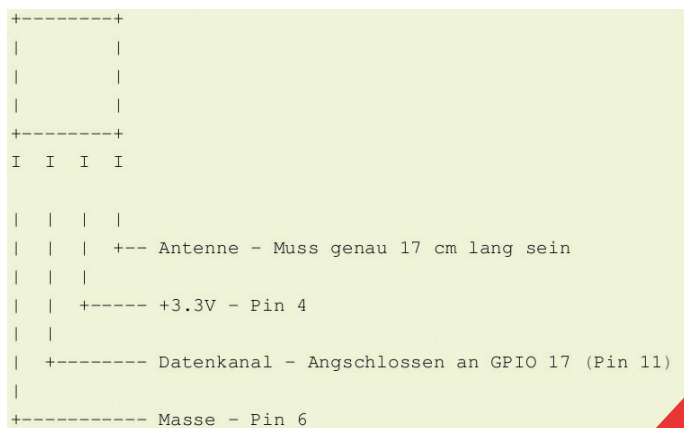


Abb. 4: Der Funksender TWS-BS-3

Jede Funksteckdose hat auf der Rückseite einen durch Pins einstellbaren Code. Dieser besteht aus einem 5-stelligen Hauscode (z.B.: 10101) und einer Geräte-ID (A,B,C,D). Möchte man nun eine Steckdose schalten dann muss z.B. der Code 11001B und 1/0 für An/Aus per Funk gesendet werden. Damit der Empfänger die Signale interpretieren kann, muss das zuvor erwähnte Protokoll (Abb. 5) eingehalten werden.

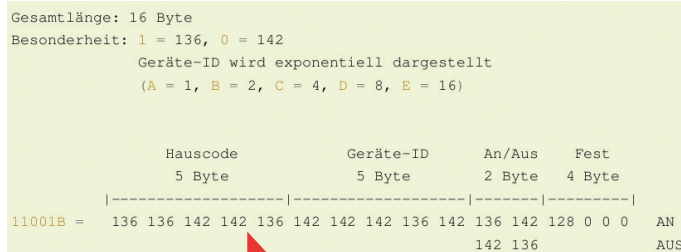


Abb. 5: Protokoll der Funksteckdosen

PowerPi ist ein Zusammenspiel aus zwei Softwareteilen. Der Server wird beim Systemstart mithilfe des init-Systems gestartet. Durch einen UDP-Port (6767) wartet er auf eingehende Befehle der Weboberfläche und speichert diese in einer Konfigurationsdatei („/etc/default/powerpi“) oder schickt direkte Befehle an die Gpio’s. Eine Aktion auf der Webobfläche lässt sich wie folgt darstellen (Socket steht hier für Funksteckdose). Beim Klick auf einen Socket wird zunächst eine Ajax-Anfrage mit allen wichtigen Informationen an die ‚lib/powerpi.php‘ geschickt. Die ‚lib/powerpi.php‘ überprüft daraufhin die übergebenen Parameter auf Gültigkeit und leitet die Anfrage über einen UDP-Socket weiter an den PowerPi-Server. Je nach Anfrage, werden Einträge in der Konfigurationsdatei hinzugefügt, gelöscht oder geändert. Konnte der Socket in der Konfigurationsdatei gefunden werden, wandelt der Server die Funksteckdosenadresse in einen binären Code um und schreibt ihn in den GPIO 17. Der Sender moduliert daraufhin den Code hoch und schaltet ihn als Spannung auf die Antenne. Die Funksteckdose empfängt die erzeugten Wellen und schaltet den gewünschten Zustand durch. Über die Weboberfläche können zeitbasierte Aufgaben (Schedules) angelegt werden. D.h. es können Funksteckdosen und/oder GPIO’s zu gewünschten Uhrzeiten geschaltet werden. Die Serverlogik sowie der Scheduler zum Verwalten der Aufgaben sind über eine Endlosschleife realisiert. Damit die beiden Komponenten sich nicht gegenseitig blockieren sind sie in POSIX-Threads ausgelagert und gemeinsame Betriebsmittel per Mutexe gesichert. Zur Überwachung sowie Analyse der Serveranwendung wird syslog verwendet. Dabei werden Verbindungsabbrüche und Fehlzugriffe auf die Konfigurationsdatei als kritisch gewertet. Zusätzlich wird die Client-/Serverkommunikation gespeichert. Beispielausgabe nach /var/log/syslog:

```
powerpi[1911]: Received: setsocket:Lampe:0
powerpi[1911]: Set Socket Lampe to 0
powerpi[1911]: Received: list:all
powerpi[1911]: Sent: socket:Lampe:11001B;schedule:w1:Lampe::18:5:1:1
```

Listing 4: Inhalt der Datei /var/log/syslog

Die Konfigurationsdatei `/etc/default/powerpi` enthält alle wichtigen Programm-einstellungen sowie die über die Weboberfläche angelegten Funksteckdosen, Gpio's und zeitgesteuerten Aufgaben. Am Anfang des Projektes hatten der Server als auch die Webanwendung Schreib- und Lesezugriff auf die Konfigurationsdatei. Es wurde schnell klar das dadurch die Datenkonsistenz nicht gewährleistet werden kann.

Hat z.B. die Webanwendung die Konfiguration verändert so musste der Server die Konfiguration ebenfalls neu einlesen und umgekehrt. Da die Serveranwendung jedoch keinen Zugriff auf den Browser hat und die Seite nicht neuladen kann, wird der alte Inhalt angezeigt und evtl. Aktionen können zum Absturz des Servers führen. Durch eine Überarbeitung beider Programme verarbeitet der Server nun ankommende Anfragen und verändert die Konfiguration dem entsprechend.

Die Weboberfläche bildet die zentrale Schnittstelle für den User. Diese wurde mit PHP und jQuery für die Logik und HTML/CSS für das Design umgesetzt. Das PHP-Grundgerüst ist eine Eigenentwicklung und kommt bei fast allen meinen Projekten zum Einsatz. Die Gestaltung der Oberfläche ist im Flat Design gehalten (Abb. 6). Flat Design ist ein durch Windows 8 aufkommender Trend mit dem Tendenz zur Minimalisierung. Ist der Web- sowie der PowerPi-Server gestartet, so kann innerhalb des Heimnetzwerkes auf die Weboberfläche zugegriffen werden.

Zusätzlich sollte die Oberfläche auch für mobile Endgeräte wie Smartphones und Tablets optimiert sein. Mit dem Bootstrap-Framework [7] und dem no-more-table-Plugin [8] zur Visualisierung großer Tabellen auf kleinen Bildschirmen ist die Oberfläche bei jeder Auflösung bequem bedienbar.

Durch die Verwendung von UDP-Sockets kann der Server auch einfach per Bash gesteuert werden. Somit können eigene Oberflächen oder Programme die Serveranwendung steuern. Unter Windows kann das Tool Telnet verwendet werden. Für Linux gibt es das Tool netcat das jedoch evtl. nachinstalliert werden muss.

The screenshot shows the PowerPi web interface with a dark blue header containing 'Home', 'Settings', and 'Help'. The main content is divided into four sections:

- Settings**: A sub-section titled 'Wireless sockets' containing a table with columns 'Name' and 'Code'. It lists 'Licht' (code 11001B), 'Fernseher' (code 11001A), and 'Hi-Fi' (code 11001C). Each row has a red 'X' button. Below the table are input fields for 'Name' and 'Code', and a green 'Add' button.
- GPIO's**: A table with columns 'Name' and 'GPIO'. It lists 'Kaffeemaschine' (GPIO 10) and 'Ventilator' (GPIO 3). Each row has a red 'X' button. Below the table are input fields for 'Name' and a dropdown for 'GPIO 2', and a green 'Add' button.
- Scheduler**: A table with columns 'Name', 'Socket', 'GPIO', 'Time (Now: 22:46:23)', and 'On/Off'. It lists 'Kaffee am Morgen' (Socket: Kaffeemaschine, Time: 07:15, On/Off: ON) and 'Wecker' (Socket: Hi-Fi, Time: 07:20, On/Off: ON). Each row has a red 'X' button. Below the table are input fields for 'Name', dropdowns for 'Choose Socket' and 'Choose GPIO', time input fields, an 'ON' dropdown, and a green 'Add' button.

Abb. 6: Weboberfläche zum Steuern der Funksteckdosen

```
echo -n "[Befehl]" | nc -4u [IP-Adresse] [Port]
```

Beispiel: `echo -n "setsocket:Licht:1" | nc -4u 192.168.178.26 6677`

Mögliche Befehle:

<code>list all</code>	Alle gespeicherten Einträge anzeigen
<code>setsocket:[socketname]:[AN/AUS]</code>	Funksteckdose ein-/ausschalten
<code>addsocket:[socketname]:[code]</code>	Funksteckdose hinzufügen
<code>delsocket:[socketname]</code>	Funksteckdose löschen
<code>setgpio:[gpioname]:[an/aus]</code>	Gpio ein-/ausschalten
<code>addgpio:[gpioname]:[gpionummer]</code>	Gpio hinzufügen
<code>delgpio:[gpioname]</code>	Gpio löschen
<code>setschedule:[schedulingname]:[aktiv/inaktiv]</code>	Aufgabe de-/aktivieren
<code>addschedule:[schedulingname]:</code>	
<code>[socketname]:[gpioname]:[stunde]:[minute]:[an/aus]</code>	Aufgabe hinzufügen
<code>delschedule:[schedulingname]</code>	Aufgabe löschen

Abb. 7: Steuerung der Funksteckdosen per Bash

```
sudo apt-get install netcat
```

Literatur

- [1] Raspbian Homepage <http://www.raspbian.org/>
- [2] Elro Datenblatt <http://tinyurl.com/ohy4wxq>
- [3] WiringPi Download wiringpi-27afc01.tar.gz
- [4] PowerPi Download powerpi-0.1.tar.gz
- [5] Elro Protokoll entschlüsselt <http://tinyurl.com/q8zggos>
- [6] Wenshing TWS-BS-3 Datenblatt <http://tinyurl.com/ctwbgeb>
- [7] Bootstrap-Framework <http://twitter.github.io/bootstrap/>
- [8] No-more-table Plugin <http://elvery.net/demo/responsive-tables/>

Ein Protokoll für das Internet of Things!

Jens Nickel <j.nickel@elektor.de>

Benedikt Sauter <sauter@embedded-projects.net>



Das Internet der Dinge wird unser Leben verändern. Wenn alles mit allem vernetzt ist, benötigt man eine gemeinsame Sprache; doch bisher ist man sich nur einig, dass Netz-Protokolle wie TCP/IP zum Einsatz kommen werden. Es fehlt ein einheitliches Protokoll auf Anwendungsebene, damit Sensoren und Aktoren beim Messen, Steuern und Regeln zusammenwirken können – und dies über Herstellergrenzen hinweg. Wir meinen: Eine ideale Aufgabe für die Community! Die Elektor-Leser und alle anderen interessierten Entwickler sind aufgerufen, uns Ideen und Lösungen zuzuschicken.

Einleitung

Das Internet of Things (IOT) ist gegenwärtig in aller Munde. Kein Wunder, denn die Vernetzung von Smartphones, Fahrzeugen, Stromzählern, Wettersensoren, Beleuchtungen, Heizungen, Haushaltsgeräten und vielem mehr eröffnet ungeahnte Möglichkeiten. Das IOT wird unser Leben noch sicherer und komfortabler machen und helfen, Energie und weitere Ressourcen möglichst effizient und umweltschonend einzusetzen.

Zwei Faktoren machen diese „vierte industrielle Revolution“ möglich. Zum einen werden Mikrocontroller zunehmend billiger, leistungsfähiger und stromsparender. Dies erlaubt es, immer mehr Geräte so intelligent zu machen, dass diese über ein Netzwerk miteinander kommunizieren können. Zum anderen hat IPv6 – die neue Version des „Internet Protocols“ (IP) – die Zahl der möglichen Internet-Adressen auf unvorstellbare 340 Trillionen mal Trillionen erweitert. Lichtschalter, LED-Beleuch-

tungen, Temperatursensoren und vieles mehr kann man nun von überall auf der Welt aus über eine eigene Adresse zugänglich machen.

Neben IP werden weitere bekannte Protokolle wie TCP und HTTP zum Einsatz kommen, damit die Teilnehmer am Internet der Dinge untereinander Nachrichten austauschen können. Diese Protokolle definieren allerdings nur einen Rahmen für die eigentlichen Daten (wie zum Beispiel Messwerte und Steuerkommandos). Damit sich Geräte über Herstellergrenzen hinweg verstehen können, benötigt man noch ein anwendungsorientiertes Protokoll. Dies definiert zum Beispiel wie ein Sensor-Messwert (der aus einer Zahl, einer Einheit und einem Zeitstempel bestehen kann) in einer Nachricht codiert werden muss.

Protokoll-Zoo

Unglücklicherweise gibt es im Moment eine große Zahl von unterschiedlichen Protokoll-Lösungen für die verschiedensten IOT-Einsatzbereiche. Manche davon sind sehr komplex, weil alle denkbaren Anwendungsfälle abgedeckt werden müssen. Gerade für kleinere Unternehmen wie Ingenieurbüros und Start-Ups,

die einzelne IOT-Produkte auf den Markt bringen möchten, ist dieser Dschungel fast undurchschaubar.

Elektor und die Zeitschrift embedded projects Journal rufen daher die Community auf, gemeinsam ein einheitliches IOT-Protokoll zu entwickeln. Wir suchen

eine Lösung, die einfach ist, so dass sie mit geringem Aufwand in eigenen Projekten eingesetzt werden kann und auch auf kleinen Mikrocontrollern „läuft“. Dass das gemeinsam entwickelte Protokoll danach von allen interessierten Entwicklern frei genutzt werden darf, versteht sich von selbst.

Randbedingungen

Das Internet der Dinge umfasst die verschiedensten Einsatzbereiche – vom Straßenverkehr bis zum Warentransport. Ganz bewusst wollen wir uns hier auf diejenigen Anwendungen konzentrieren, die für kleinere Unternehmen, Ingenieurbüros, Hochschulteams und Open-Source-Initiativen wichtig sind, um Produkte und Lösungen für das IOT beisteuern zu können. Wir haben daher ein paar Rahmenbedingungen festgelegt, die das Protokoll erfüllen sollte.

- Zu Testzwecken soll man mit Teilnehmern per Standard-Browser kommunizieren können (Eingabe in der Adresszeile, Ausgabe im HTML-Fenster).
- Das Protokoll sollte daher ASCII-basiert und nicht binär sein.
- Als Datenrahmen sollten bekannte Formate wie JSON, XML o.ä. verwendet werden, die bereits in vielen Programmiersprachen unterstützt werden.

- Die Umsetzung soll auf einem Internetserver (PC, Tablet) oder Mikrocontroller erfolgen können.
- Eine Verschlüsselung der Kommunikation soll möglich sein.

Darüber hinaus haben wir drei Anwendungsszenarien entworfen, für die sich das Protokoll eignen sollte.

Diese „Use cases“ stammen aus den Bereichen Messtechnik/Labor, der Hausautomatisierung und der allgemeinen Gerätekommunikation (siehe www.iot-contest.com/index.php?content=infos).

Die eingesandten Ideen und Lösungen können sich auf diese Anwendungsfälle beziehen, sie müssen das aber nicht. Sicherlich gibt es noch weitere Anwendungen oder abweichende Ideen, die wir uns gemeinsam anschauen sollten. Gleiches gilt für die Rahmenbedingungen; auch diese stehen nicht unverrückbar

im Raum, sondern sollen zu Diskussionen und der Ideenfindung anregen. Selbstverständlich können auch bereits bestehende Protokolle als Lösungen eingereicht werden!

Roadmap

Bis zum 1. August 2014 sind alle interessierten Entwickler aufgerufen, uns über die Website www.iot-contest.com Lösungen und Ideen einzusenden. Danach schließt sich eine Bewertungsphase an, in der wir – wissenschaftlich begleitet – die Lösungen diskutieren werden. Wer hier als Experte dabei sein will, kann sich bereits jetzt auf der Website in einen Newsletter eintragen.

In dieser Phase gilt es, aus der (vermutlich) breiten Palette von eingesandten Vorschlägen ein gemeinsames Protokoll zu extrahieren. Wie schon beschrieben haben wir uns dabei die Einfachheit auf die Fahnen geschrieben – das Ergebnis sollte mit allen Erklärungen auf ein Poster passen, das selbstverständlich für jedermann frei downloadbar sein wird.

Danach wird in kurzer Zeit Hardware und Software erhältlich sein, die unser IOT-Protokoll nutzt. Hierfür sorgen unter anderem die Partner dieses Projekts (siehe Kasten).

Wir werden Sie in den nächsten Ausgaben über die Initiative auf dem Laufenden halten. Bleiben Sie dran und machen Sie mit!

Wir suchen Partner!

Ein allgemein verwendbares Protokoll zu etablieren ist eine große Aufgabe – wir werden nur erfolgreich sein, wenn das Protokoll möglichst rasch von vielen Produkten (Hardware/Software) unterstützt wird. Deshalb haben wir bereits Partnerunternehmen gewonnen – und suchen noch weitere. Natürlich können auch Partner eigene Ideen einreichen!

Um Partner zu werden, müssen Unternehmen, Start-Ups und Open-Source-Initiativen nur wenige Bedingungen erfüllen (siehe www.iot-contest.com). Die Teilnahme ist komplett kostenfrei.

Bisherigen Partner der Initiative

-  Elektor International Media (www.elektor.de)
-  Embedded Projects (www.embedded-projects.net)
-  WIZnet (www.wiznet.eu)
-  TinkerForge (www.tinkerforge.com)
-  NetIO (<http://netio.davideickhoff.de>)
-  Xtronic GmbH (www.xtronic.de)



Anzeige

Elektronikbausätze




BS1019 - MEMS Modul
mit MMA7361: 8,00 €*



BS1016 - DC Motortreiber
mit L298: 12,00 €*

Weitere Bausätze und Leiterplatten (u.a.):

- LP1004 - Pi Add On Leiterplatte, für den Raspberry Pi: 3,50 €*
- BS0903 - Steppermodul mit TMC222: 20,00 €*
- BS0703 - AVR910 Programmieradapter für Atmel-Controller: 15,00 €*
- BS1006 - Androino, ein Arduino Clone: 15,00 €*
- BS1013 - Android Interface Board, ein IOIO Clone: 34,00 €*
- BS1009 - Bausatz für ein Universallauflicht: 8,00 €*

* inkl. MwSt. zzgl. Versandkosten: DE: 5,00 €, EU: 10€

Hier im Shop: b-redemann.de

B. Redemann, Mahlower Str. 204 14513 Teltow (kein Laden!)

Lesen Sie die neue Elektor ein Jahr lang in der ultimativen GOLD-Mitgliedschaft und profitieren Sie von allen Premium-Vorteilen!



Die Elektor-GOLD-Jahresmitgliedschaft bietet Ihnen folgende Leistungen/Vorteile:

- Sie erhalten **10 Elektor-Hefte** (8 Einzelhefte + 2 Doppelausgaben Januar/Februar und Juli/August) pünktlich und zuverlässig frei Haus.
- **Extra:** Jedes Heft steht Ihnen außerdem als PDF zum sofortigen Download unter www.elektor-magazine.de (für PC/Notebook) oder via App (für Tablet) bereit.
- **Neu & Exklusiv:** Sie erhalten alle 2 Wochen per E-Mail ein neues Extra-Schaltungsprojekt (frisch aus dem Elektor-Labor).
- **Neu & Exklusiv:** Wir gewähren Ihnen bei jeder Online-Bestellung 10% Rabatt auf alle unsere Webshop-Produkte – dauerhaft!
- **Neu & Exklusiv:** Der Online-Zugang zum neuen Community-Bereich www.elektor-labs.com bietet Ihnen zusätzliche Bauprojekte und Schaltungsideen.
- **Extra:** Die neue Elektor-Jahrgangs-DVD (Wert: 27,50 €) ist bereits im Mitgliedsbeitrag inbegriffen. Diese DVD schicken wir Ihnen sofort nach Erscheinen automatisch zu.
- **Extra:** Top-Wunschprämie (im Wert von 30 €) gibts als Dankeschön GRATIS obendrauf!



UMWELTSCHONEND – GÜNSTIG – GREEN

Möchten Sie Elektor lieber im elektronischen Format beziehen? Dann ist die neue GREEN-Mitgliedschaft ideal für Sie! Die GREEN-Mitgliedschaft bietet (abgesehen von den 10 Printausgaben) alle Leistungen und Vorteile der GOLD-Mitgliedschaft.



Jetzt Mitglied werden unter www.elektor.de/mitglied!

Analogen Stromzähler elektronisch auslesen

Johannes Kaufmann <Johannes.Kaufmann@hs-augsburg.de>

Nikolai Kraler <Nikolai.Kraler@hs-augsburg.de>

Michael Liebert <Michael.Liebert@hs-augsburg.de>

Ältere Stromzähler - wie in den meisten deutschen Haushalten vorhanden - bieten keine Schnittstelle an, um den aktuellen Stromverbrauch oder den Zählerstand elektronisch auslesen zu können. Diese Dokumentation sucht nach Möglichkeiten, mittels technischer Unterstützung durch das GNUBLIN Embedded Linux-Board, solch einen Stromzähler automatisiert zu "beobachten" und die gewonnenen Informationen im Heimnetzwerk bereit zu stellen.

Variante 1: Texterkennung

Mit einer am GnuBLIN angeschlossenen Webcam sollen in regelmäßigen Abständen Bilder gemacht werden, aus welchen dann der aktuelle Zählerstand ausgelesen werden soll. Die erhaltenen Informationen sollen auf einer kleinen Webseite dargestellt werden. Als Basis für weitere Nachforschungen haben wir einen Stromzähler im Keller bei nicht idealen Lichtbedingungen mit der Webcam fotografiert:

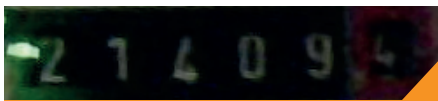


Abb. 1: Bildqualität wird zum Problem

Mit Hilfe der Open-Source OCR-Tools (Optical Character Recognition) tesseract und GOCR, haben wir versucht, den Text auszulesen.

```
tesseract strom.tif out
```

```
cat out.txt
```

```
vE`ILUB.·g
```

Dieses schlechte Ergebnis ist, angesichts der eigentlich für einen Menschen recht gut lesbaren Ziffern, sehr überraschend. Als nächsten Schritt haben wir tesseract so konfiguriert, dass es nur Ziffern erkennt, womit die Anzahl an Möglichkeiten auf 10 (0-9) eingeschränkt wird. Seit der Version 3.0 von tesseract kann dies, mit Hilfe des Schlüsselworts "digits", ganz einfach erreicht werden.

```
tesseract strom.tif out
nobatch digits
```

```
cat out.txt
```

```
5 777 5
```

```
8110523
```

Auch mit diesem Ergebnis können wir nicht wirklich zufrieden sein, weshalb wir es schließlich mit einem hochauflösendem Bild eines Stromzählers versuchten:

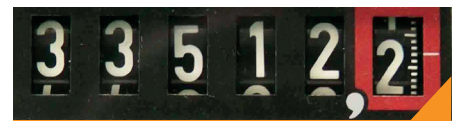


Abb. 2: Hochauflösendes Bild

```
tesseract stromi.tif out
nobatch digits
```

```
cat out.txt
```

```
3 3 5 1 2 9
```

Da selbst bei diesem perfekten Bild eines Stromzählers die "2" nach dem Komma als "9" erkannt wird, haben wir erkannt, dass wir mit Hilfe von Texterkennung nicht weiter kommen.

Variante 2: Ferrariszähler-Drehscheibe beobachten

Da die Texterkennung des aktuellen Zählerstandes nicht umsetzbar war, haben wir uns entschlossen, die Ferrariszähler-Drehscheibe mit Hilfe einer Lichtschranke zu beobachten. Dabei entspricht jede Umdrehung der Ferraris-Zählerscheibe einem angegebenen Teil einer Kilowattstunde. In Abbildung 3 ist zu erkennen, dass 71 Umdrehungen einer KW/h entsprechen.

Den Durchlauf einer Umdrehung ermitteln wir mit Hilfe eines Phototransistors, welcher auf die unterschiedlich stark reflektierenden Farben der Zählerscheibe reagiert – Silber spiegelt besser als Rot. Da wir eine klassische Lichtschranke einsetzen steigt die Spannung, wenn die rote Markierung vorbeikommt – somit haben wir bei jeder Umdrehung einen kurzen Impuls.

Die folgende Grafik (siehe Abb.4) zeigt den Schaltplan einer Lichtschranke – D1 ist eine Infrarot-LED und T2 ein Phototransistor. Zur Vereinfachung der Schaltung, wurde in einem ersten Versuch auf den Transistor T2 verzichtet und die Messspannung direkt am zwischen R und T1 abgegriffen.

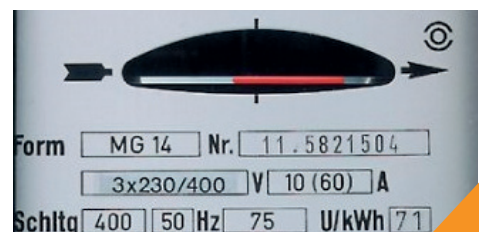


Abb. 3: Eine Ferrariszähler-Drehscheibe

Angeschlossen wird die Platine mit 3 Adern: Die Versorgungsspannung (+3,3 V) und Masse wird über die Anschlüsse 3,3V sowie GND am GnuBlin-Board bereitgestellt. Als Mess-Eingang entschieden wir uns zuerst für den GPIO-Anschluss IO11 und konfigurierten ihn wie im GnuBlin-Wiki (<http://wiki.gnublin.org/index.php/GPIO>) beschrieben als Eingang. Wenn nun die Infrarot-LED frontal auf den Fototransistor ausgerichtet wird und der Eingang mittels `cat /sys/class/gpio/gpio11/value` ausgelesen wird, erscheint eine 0 (keine Unterbrechung der Lichtschranke). Wenn die beiden Bauteile voneinander entfernt werden und `value` ausgelesen wird, wird eine 1 ausgegeben (Lichtschranke unterbrochen). Um den Wert des GPIO automatisch auszulesen, entschieden wir uns für die ressourcenschonende Variante mit Hilfe des System Calls `poll` und `Interrupts`. Dazu wird zuerst mittels `echo „falling“ > /sys/class/gpio/gpio11/edge` der Eingang so konfiguriert, dass bei fallender Flanke, also wenn die Lichtschranke unterbrochen wird, ein Interrupt ausgelöst wird. Der Sourcecode eines ebenfalls im Wiki unter GPIO abgelegten Tools, wurde von uns modifiziert (`poll2.c`), sodass es dauerhaft läuft und bei jedem Interrupt die aktuelle Uhrzeit und eine Zählervariable auf der Konsole ausgibt. Nun ist es an der Zeit, die Schaltung im vorgesehenen Zielumfeld zu testen. Infrarot-

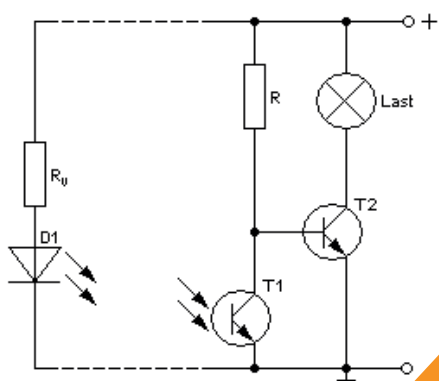


Abb. 4: Detektionsschaltung

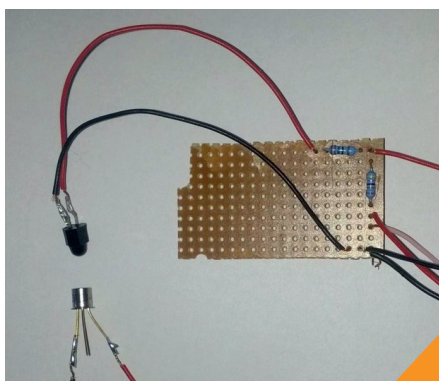


Abb. 5: Aufbau der Lichtschranke

```
#!/bin/bash
LOG=adc.log
THRESHOLD=2700
VAL=0;
while true
do
    VAL=`gnublin-adcint -c 1 -b`
    if [ $VAL -gt $THRESHOLD ]
    then
        echo „$VAL $(date) !!!“ | tee -a $LOG
        echo „$VAL $(date)“ >> peaks.log
    else
        echo „$VAL $(date)“ | tee -a $LOG
    fi
done
```

Listing 1: adclog.sh

```
2581 Sun Jun 23 15:41:05 UTC 2013
2581 Sun Jun 23 15:41:06 UTC 2013
2581 Sun Jun 23 15:41:06 UTC 2013
2581 Sun Jun 23 15:41:07 UTC 2013 !!!
2581 Sun Jun 23 15:41:07 UTC 2013 !!!
2581 Sun Jun 23 15:41:08 UTC 2013
2581 Sun Jun 23 15:41:09 UTC 2013
2581 Sun Jun 23 15:41:00 UTC 2013
```

Listing 2: Ausgabe des Testscripts

LED und Fototransistor werden dazu mit Hilfe eines Pappkartons, in den zwei Löcher für die beiden Bauteile gebohrt sind, leicht schräg ausgerichtet direkt vor der Ferraris-Drehscheibe des Zählers montiert. Dadurch kann die Drehscheibe direkt als "Spiegel" verwendet werden. Nach Starten des Tools `poll2` erfolgt das ernüchternde Ergebnis: auch nach 5 Minuten warten erscheint keine einzige Ausgabe auf dem Terminal. Offenbar ist die Spiegelung zu gering, sodass die Lichtschranke immer als unterbrochen angesehen wird. Da beim Digitaleingang wohl ein uns unbekannter Schwellwert im Treiber entscheidet, bei welcher Eingangsspannung zwischen 1 und 0 umgeschaltet wird, konnten wir hier erstmal nicht ohne weiteres fortfahren.

Anstatt die Messleitung an den GPIO anzuschließen, entschieden wir uns testweise für das Verwenden des Analogeingangs GPA1. Da dort kein Polling mittels `Interrupts` möglich ist, schrieben wir ein kurzes Shell-Script, welches in einer Dauerschleife `gnublin-adcint -c 1 -b` aufruft, um die aktuell anliegende Spannung in mV am GPA1 zu messen. Nach ersten Tests konnte ein Grenzwert (Variable `THRESHOLD`) festgelegt werden, unterhalb welchem "silber" und überhalb welchem "rot" angenommen werden soll.

Aktuellen Stromverbrauch berechnen und bereitstellen

Nachdem der Analogeingang als für das Vorhaben geeignet evaluiert wurde, ging es an die Implementierung von `gnublin-ferraris`. Diese in C++ geschriebene Anwendung läuft als Daemon auf dem GNUBLIN und liest in einstellbaren, periodischen Abständen die Spannung aus, die am Analogeingang anliegt. Basierend auf einem konfigurierten Schwellwert wird analog zu `adclog.sh` erkannt, wann die rote Markierung eine Runde im Zähler vollzogen hat. Aus Zählerkonstante und verstrichener Zeit wird mithilfe der Formel $3600 / \text{Umdrehungen_pro_kWh} / \text{Sekunden_Umdrehung} * 1000$ der aktuelle Stromverbrauch

in Watt berechnet. Alle Informationen werden dann nach jeder Runde in der Datei `/var/www/gnublin-ferraris.csv` gespeichert, welche im Browser über <http://gnublin-ip/gnublin-ferraris.csv> abgerufen werden kann. Alternativ können die Informationen über eine auf Lesbarkeit optimierte Version unter <http://gnublin-ip/cgi-bin/powermeter> betrachtet werden - mehr Details zur Implementierung mittels `cgi-bin` im Kapitel `Lighttpd` mit CGI-Script. Der Sourcecode von `gnublin-ferraris` kann über Github bezogen werden und kann direkt auf dem GNUBLIN oder cross-kompiliert werden [1].

Web-Frontend: Node.js testen

Um stets die aktuellsten Stromzähler-Informationen über ein Web-Frontend bereitzustellen, wurde versucht, node.js auf dem GNUBLIN zum Laufen zu bekommen. Unter <https://github.com/downloads/ant9000/FoxNode/nodejs-foxboard-debs.tar> stehen bereits für ARM vorkompilierte nodejs-deb-Pakete bereit. Nachdem deren Abhängigkeit libc-ares2 mittels apt-get install nachinstalliert wurde, ließen sich die Pakete manuell mittels dpkg -i <paketname> installieren. Leider ließ sich ein kleines Beispiel-Projekt mit Web- und WebSocket-Server aufgrund der Fehlermeldung RangeError: Maximum call stack size exceeded nicht starten. Aufgrund dessen und bedingt durch die Tatsache, dass node eher zu ressourcenlastig für das GNUBLIN-Board ist, wurde fürs erste entschieden, das Web-Frontend mit Lighttpd und cgi-bin zu realisieren.

Lighttpd mit CGI-Script

CGI-Modul von lighttpd aktivieren:

```
lighty-enable-mod cgi
```

CGI-Config wie folgt bearbeiten:

```
$HTTP[„url“] =~ „^/cgi-bin/“
{
    cgi.assign += ( „.py“ =>
„/usr/bin/python“ )
    cgi.assign += ( „“ => „“ )
}
```

Listing 1: CGI-Config

Damit auch python-Scripte ausgeführt werden können, muss zusätzlich zu "" => "", die Endung .py mit dem python-Interpreter verbunden werden (Achtung: Reihenfolge beachten!):

```
„.py“ => „/usr/bin/python“
```

Standardmäßig werden die cgi-Scripte in /usr/lib/cgi-bin/ abgelegt. Damit diese Scripte unter <server-ip>/cgi-bin/ erreichbar sind, muss zusätzlich in der lighttpd-Config /etc/lighttpd/lighttpd.conf ein Alias eingefügt werden:

```
alias.url += ( „/cgi-bin/“
=> „/usr/lib/cgi-bin/“ )
```

Außerdem müssen die Endungen .sh und .py exkludiert werden:

```
static-file.exclude-extensions = ( „.php“, „.pl“, „.fcgi“, „.sh“, „.py“, )
```

Nachdem alles richtig Konfiguriert wurde muss lighttpd neugestartet bzw. die Konfiguration neu geladen werden:

```
service lighttpd restart/
reload
```

Im cgi-bin-/Verzeichnis können nun beliebige CGI-Scripte geschrieben werden. Unser CGI-Script schrieben wir, der Einfachheit halber als bash-Script powermeter. Das Script liest lediglich die wichtigsten Daten von einem, von unserem Backend erstellten, File und stellt diese einem Browser zur Verfügung. Damit das Script im Browser angezeigt wird, muss es noch ausführbar gemacht werden:

```
chmod +x powermeter
```

Listing 1: Auszug aus der cgi-bin Datei powermeter (vgl. Abb. 6)

```
IN=$(head -n 1 /var/www/gnublin-ferraris.csv)
IFS=";" read -a ARR <<< „$IN“

echo „<table border=“1“>“
echo „<tr><td>Current power consumption</td><td>${ARR[0]} W</td></tr>“
echo „<tr><td>Last cycle duration</td><td>${ARR[1]} s</td></tr>“
echo „<tr><td>Last cycle completed on</td><td>${ARR[2]}</td></tr>“
echo „<tr><td>Cycles per kWh</td><td>${ARR[3]}</td></tr>“
echo „</table>“
```

Powermeter Stats

Current power consumption	2668.6 W
Last cycle duration	19 s
Last cycle completed on	Thu Jul 18 01:36:40 2013
Cycles per kWh	71

Abb. 6: Ausgabe auf der Benutzeroberfläche

Links

[1] Sourcecode zu diesem Projekt <https://github.com/michael-liebert/gnublin-ferraris>

Backport Kernel

Neue Hardware auf altem Kernel

Filou (aus Gnublin Wiki)

Als Grundlage diente mir ein auf buildroot basiertes Gnublin Linux. Die Vorgehensweise bei einer Debian Distribution ist analog, da sie auf dem gleichen Kernel aufbaut. Kleine Anmerkung: Das Ganze ist ein iterativer und kein linearer Prozess, deswegen wird es auch etwas sprunghaft zugehen. Es ist auch noch keine allgemeingültige Anleitung, sondern eher ein Erfahrungsbericht, ein Blick hinter die Kulissen, Hilfe zur Selbsthilfe.

Buildroot

Ausgehend von benbreasons version (<http://code.google.com/p/gnublin-buildroot-git/>) mit make ein Gnublin Linux bauen und testen. Damit es später keine Probleme mit den Pfaden gibt, am besten nicht das eigene Homeverzeichnis sondern /opt/gnublin/buildroot als Basisver-

zeichnis verwenden. Der weitere Sourcecode für die extern zu bauenden Treiber und die Quellen für Compat-Wireless (<http://backports.wiki.kernel.org>) kommen dann nach opt/gnublin/src. Alle Patches und Pfadangaben beziehen sich auf dieses Layout.

Ist die Toolchain, der Kernel und das RootFS gebaut und getestet, kann / muss der Kernel passend konfiguriert werden. Bevor wir uns allerdings damit beschäftigen, erst noch ein paar Anmerkungen zum Kernel-Build-System.

Kernel

Seit Kernel 2.6 wird ein neues Konfigurationsschema verwendet, das die verschiedenen Optionen verwaltet, die notwendig sind um ein flexibles und modulares Betriebssystem zu bauen. Auch der Bau von Treibern profitiert davon, da hiermit auch die Abhängigkeiten zwischen den einzelnen Komponenten und Modulen gesteuert werden. Prinzipiell gibt es zwei Arten Kernel-Module zu bauen: extern in einem eigenen Verzeichnis, oder aber innerhalb des Kernelbaums. Beides hat Vor- und Nachteile. Da Backports auf verschiedene Kernelversionen angewandt werden, ist es sinnvoll die Quellen außerhalb des jeweiligen Kerns zu

verwalten und nur über das Buildsystem zur Übersetzungszeit die Verknüpfung herzustellen.

Ein typisches Makefile für ein externes Module sieht in etwa folgendermaßen aus (Listing 1).

Dieses funktioniert auch für Embedded-Systems problemlos, wenn man Toolchain und Kernelverzeichnis explizit angibt. Im Gegensatz zu vielen Hardwareherstellern haben sich die Maintainer des Compat-Wireless-Projekts daran gehalten und damit die besten Voraussetzungen für das Erstellen der notwen-

digen Module geschaffen. Der Wermutstropfen liegt nun darin, dass man die Abhängigkeiten, die uns das Kernel-buildsystem normalerweise auflöst, nun selbst bearbeiten muss.

Damit das mit dem Übersetzen grundsätzlich funktioniert, sind Ergänzungen im Makefile notwendig (Listing 2).

Ein ‚make clean‘ führt dann zu einigen Warnungen bezüglich nicht gesetzter Build-Optionen und zum oben beschriebenen Problem, da diese nicht in der üblichen Weise in Kernel-Configuration explizit gesetzt werden können. Ein ➔

```
obj-m := ext_kernel_modul.o
KERNELDIR := /opt/gnublin/buildroot/output/build/linux-2.6.33/
CROSS_COMPILE := /cvopt/gnublin/buildroot/output/host/usr/bin/arm-linux-
PWD := $(shell pwd)
ARCH := arm
all:
$(MAKE) ARCH=$(ARCH) CROSS_COMPILE=$(CROSS_COMPILE) -C $(KERNELDIR) M=$(PWD) modules
clean:
rm -rf *.o *~ core .depend *.cmd *.ko *.mod.c .tmp_versions
```

Listing 1: Typisches Makefile für externe Module

```
KLIB:= „/opt/gnublin/buildroot/output/build/linux-2.6.33“
ARCH:= „arm“
CROSS_COMPILE:= „/opt/gnublin/buildroot/output/host/usr/bin/arm-linux-“
```

Listing 2: Grundsätzliche Änderungen am Makefile zum Übersetzen

➤ Blick in ein typisches Config-File zeigt das Dilemma, Optionen werden implizit bei der Auswahl einzelner Module ausgewählt, nur diese wollen wir ja unter Umständen in der neuen Version erst extern übersetzen oder benötigen sie gar nicht. Nach längerem experimentieren blieben drei Kandidaten übrig, die dann der entsprechenden Steuerdatei hinzugefügt wurden.

In Datei Kconfig in output/build/linux-2.6.33/arch/arm/mach-lpc313x/ am Ende folgende Ergänzungen vornehmen:

Die neuen Module benötigen den Netzwerktreiber Scheduler, die Wireless Extensions (für iwconfig und wpa-tools) und Atomare 64Bit Operationen. Letztere werden uns nachher nochmals begegnen.

Im Menu sind dann die Flags zu setzen, wenn Treiber aus der Compat Bibliothek hinzugefügt werden sollen.

Es gibt noch weitere Abhängigkeiten die im Laufe des Build-Prozesse auftreten, teilweise erst ganz zum Schluss, wenn das Root-Filesystem gebaut wird, dazu später mehr. Damit verlassen wir fürs erste das Kernel-Build-System und wenden uns der Compat-Wireless zu.

Compat-Wireless

Als Erstes, download der stabilen Release von <http://www.kernel.org/pub/linux/kernel/projects/backports/2012/10/08/> in meinem Fall compat-drivers-2012-10-08-c.tar.gz, die den Kernel bezüglich des Wireless-Subsystems auf den Level von 3.x anhebt. Wie oben schon angedeutet nach /opt/gnublin/src entpacken und das Makefile anpassen.

Ein erster Blick in das Quellverzeichnis zeigt zwei wichtige Ordner, compat und driver. Der Erstere ist für die Rückwärtskompatibilität zuständig, der Letztere enthält dann die aktuellen 3.x Kernel Treiber. Hier ist auch das Modul, das letztlich Anlass für den ganzen Aufwand war zu finden, der Treiber für die neuesten WLAN-Sticks mit dem Ralink RT5370 Chipsatz. Die notwendige Firmware hatte ich schon auf meinem Debian-Entwicklungssystem, firmware-ralink.deb aus dem Archive von <http://backports.debian.org/debian-backports>. Damit sind zumindest alle Zutaten vorhanden und das es prinzipiell funktioniert, habe ich in einer Debian6 VM mit x86 Architektur schon ausprobiert.

Jetzt wird es Zeit für einen ersten Versuch, im Buildroot-Verzeichnis, die Optionen setzen und Kernel neu kompilieren:

```
make linux-menuconfig
make
```

Das nächste Problem! Da der 80211 Layer gegen einen aus dem Backport ausgetauscht werden muss, muss dieses zwangsläufig als Modul übersetzt werden, nächste Runde:

```
NetworkingSupport --->
Wireless --->
  <M>  cfg80211 -
wireless configuration API
```

Nächstes und auf den ersten Blick nicht triviales Problem, Syntaxfehler in lib/atomic64.c. Da dieses Problem in der Vorstudie auf der x86 Architektur nicht auftrat, muss hier das Problem auf der ARM Seite liegen!

```
// #include <asm/atomic.h>
#include <asm-generic/atomic64.h>
```

```
.....
config NET_SCHED
bool „Gnublin Extra Flag NET_SCHED“
help
    Say Y here if you are using extra flags

config WIRELESS_EXT
bool „Gnublin Extra Flag WIRELESS_EXT
& LIBIPW“
help
    Say Y here if you are using extra flags

config GENERIC_ATOMIC64
bool „Gnublin Extra Flag GENERIC_ATOMIC64“
help
    Say Y here if you are using extra flags
.....
```

Listing 3: Ergänzungen an der Datei Kconfig

```
SystemType --->
Special settings --->
  [*] Gnublin Extra Flag NET_SCHED
  [*] Gnublin Extra Flag WIRELESS_EXT & LIBIPW
  [*] Gnublin Extra Flag GENERIC_ATOMIC64
```

Listing 4: Flags für Treiber aus der Compat Bibliothek

ARM Prozessoren kennen keine atomaren 64Bit Operationen, deshalb muss auf eine Softwarelösung mit Spinlocks zurückgegriffen werden. Die Einbindung von arch/arm/... ist da noch unvollständig, die passende Include-Datei wird fürs erste von Hand eingebunden. Das nächste ‚make‘ führt dann zu einem fehlerfrei übersetzten und lauffähigen Kernel. Wir verlassen das Buildroot-Verzeichnis und versuchen die compat drivers zu übersetzen. Damit auftauchende Probleme besser eingegrenzt werden können, beschränken wir uns auf die gesuchten WLAN-USB-Treiber.

```
scripts/driver-select rt2x00
```

Das Script wählt alle notwendigen Bestandteile aus und erzeugt eine passende Konfiguration für das nachfolgende make. Im Makefile für die compat drivers nun die oben beschriebenen Änderungen vornehmen (Listing 5).

Ein weitere Fehler in Bezug auf die Atomic-Operationen taucht auf! Das gleiche Spiel wie oben, in der Datei net/mac80211/mesh_pathtbl.c die notwendigen Ergänzungen einfügen.

```
#include <asm/atomic.h>
static inline int
atomic_add_unless(          atomic_t *v,
int a, int u ) {          return __atomic_
add_unless(v, a, u);
}
```

Immerhin jetzt läuft der Compiler durch, übrig bleiben drei Linker Warnings, crc_ccitt, cpufreq_cpu_get und cpufreq_cpu_put undefined! Die Ursache für die erste Warning ist ein noch fehlendes Kernel-Module, die letzten beiden sind modernen CPUs mit variabler Taktfrequenz geschuldet. Mit grep den Quelltext durchsucht und in compat/compat-3.1.c auskommentiert.

Das Kernel-Module aktivieren:

```
Library routines --->
<M> CRC-CCITT functions
```

Kernel und Treiber neu übersetzen. Wenn jetzt alles ohne Fehler durchgelaufen ist, ein mutiges ‚make install-modules‘ und in `/opt/gnublin/buildroot/output/target/lib/modules/2.6.33/updates/` stehen die neuen Treiber.

```
#----- GNUBLIN -----
KLIB_BUILD:= /opt/gnublin/buildroot/output/build/linux-2.6.33
export KMODDIR?= updates
KMODDIR_ARG:= „INSTALL_MOD_DIR=$(KMODDIR)“
KMODPATH_ARG:= „INSTALL_MOD_PATH=/opt/gnublin/buildroot/output/target“
export KLIB:= /opt/gnublin/buildroot/output/build/linux-2.6.33
MAKE:= make ARCH=arm CROSS_COMPILE=/opt/gnublin/buildroot/output/host/usr/bin/arm-linux-
#----- GNUBLIN -----
```

Listing 5: Änderungen im Makefile vornehmen

Hotplug

Wer nun glaubt das wäre es, der irrt! Nun gilt es, die Treiber auf der Zielplattform auch zum laufen zu bringen und dazu benötigen wir, wie bei den meisten neueren WLAN Sticks und Karten, Firmware! Die gibt's auf besagten <http://backports.debian.org/debian-backports> und die müssen nun ebenfalls wie die Treiber ins Target-Verzeichnis gespeichert werden. Wo genau hängt vom verwendeten Hotplug System ab! Wie Hotplug im Detail genau funktioniert steht in einem eigenen Artikel. Hier nur so viel, wie wir für die ersten Gehversuche benötigen.

Im 2.6 Kernel ist eine, im Prinzip einfach zu handhabende, Hotplug-Schnittstelle eingebaut. In `/proc/sys/kernel/hotplug` steht der Name eines Scriptes das bei einem Event aufgerufen wird, die notwendigen Informationen werden auf der Kommandozeile und in Umgebungsvariablen übergeben. Standardmäßig lautet der Name `/sbin/hotplug`. Der Kernel ruft wenn ein USB Gerät ein/abgesteckt wird `/sbin/hotplug` mit folgenden Argumenten auf:

```
argv [0] = hotplug_path
argv [1] = „usb“
argv [2] = 0
HOME=/
PATH=/sbin:/bin:/usr/sbin:/usr/bin
ACTION=action
PRODUCT=idVendor/idProduct/bcdDevice
TYPE=device_class/device_subclass/device_protocol
INTERFACE=class/subclass/protocol
DEVFS=/proc/bus/usb
DEVICE=/proc/bus/usb/bus_number/device_number
```

Listing 5: Aufruf von hotplug

Zum Testen reicht uns fürs erste ein simples Script,

```
#!/bin/sh
env >> /dev/ttyS0
```

das alle Ausgaben auf die Gnublin Konsole umleitet. Damit es auch sicher vom Kernel aufgerufen wird, den Pfad im Proc-Filesystem abspeichern.

```
echo /sbin/hotplug > /proc/sys/kernel/hotplug
```

Nach Einstecken des WLAN-Sticks meldet uns das USB-Subsystem das neue Gerät. Ein ‚lsmod‘ hingegen zeigt kein geladenes Modul und das ist auch korrekt so, denn fürs laden der Module ist der Useranteil des Hotplug-Systems verantwortlich, es ist also fürs Erste Handarbeit angesagt. Unser Script gibt uns allerdings eine ganz Reihe von Informationen zurück:

```
DEVTYPE=usb_device
SUBSYSTEM=usb
DEVPATH=/devices/platform/lpc-ehci.0/usb1/1-1/1-1.4
DEVNUM=005
MINOR=4
ACTION=add
BUSNUM=001
MAJOR=189
DEVNAME=bus/usb/001/005
DEVICE=/proc/bus/usb/001/005
PRODUCT=148f/5370/101
```

Ein ‚modprobe rt2800usb‘ lädt den Treiber nebst den notwendigen Modulen. Auf einem 8MB Board unbedingt swap-space aktivieren. Trotzdem ist der Ladevorgang wegen des geringen Speichers sehr holprig, daher möglichst kleinen und kompakten Kernel verwenden und

nicht benötigte Services deaktivieren. ‚lsmod‘ zeigt uns dann die geladenen Module.

Ein neues Gerät mit dem Namen wlan0 ist vorhanden, ein ‚ifconfig wlan0 192.168.6.99‘ verlangt nach einem Firmware upload:

```
SUBSYSTEM=firmware
DEVPATH=/devices/platform/lpc-ehci.0/usb1/1-1/1-1.4/1-1.4:1.0/firmware/1-1.4:1.0
FIRMWARE=rt2870.bin
```

Wer ein sehr kompaktes System bauen will oder muss, kann sich mit dem Beispielscript aus der Kernel-Dokumentation behelfen (`linux-2.6.33/Documentation/firmware-class/hotplug-script`). Zuvor muss allerdings noch das Verzeichnis `/lib/firmware` angelegt und die `rt2xxx.bin` Dateien darin abgespeichert werden.

```
HOTPLUG_FW_DIR=/lib/firmware/
echo 1 > /sys/$DEVPATH/loading
cat $HOTPLUG_FW_DIR/$FIRMWARE > /sys/$DEVPATH/data
echo 0 > /sys/$DEVPATH/loading
```

Listing 6: Simple hotplug script

Das Tor zu einem unverschlüsselten WLAN steht uns nun offen. Für eine endgültige, vor allem automatische Lösung sollte man doch besser eines der beiden fertigen Userland-Hotplug-System verwenden. Ich habe mich wegen des geringen Speicherbedarfs und der Einfachheit für mdev entschieden und es in der Buildroot BusyBox Konfiguration aktiviert.

mdev - Automatisch ins WLAN

Wir benutzen mdev als Hotplug-Script und aktivieren das Hotplug im System-Startscript `/etc/init.d/rcS`:

```
#!/bin/sh

# Init hotplug interface

echo /sbin/mdev > /proc/sys/kernel/hotplug
```

Die Konfiguration befindet sich in der Datei `/etc/mdev.conf`:

```
$MODALIAS=.*      0:0 0660 @modprobe „$MODALI-
AS“

wlan[0-9]*       0:0 0660 */etc/mdev.wlan
```

Die erste ruft `modprobe` auf und lädt die notwendigen Module. Wenn alles gut geht und das USB-Gerät ein WLAN-Device ist, existiert das dazu passende Netzwerkinterface `wlan0` und als Aktion wird das Script `mdev.wlan` ausgeführt:

```
#!/bin/sh

LOG="logger -p user.info -t mdev-wlan"
WARN="logger -p user.warn -t mdev-wlan"

case „$ACTION“ in
    add | „“ )

if [ -s /etc/wpa_supPLICANT.conf ];

then
$LOG „trying to bring $MDEV up“

wpa_supPLICANT -B -Dwext -i $MDEV -c /etc/
wpa_supPLICANT.conf

ifup $MDEV

else

$WARN „/etc/wpa_supPLICANT.conf missing or
empty, not trying to bring $MDEV up“

fi
;;

    remove )

$LOG „trying to bring $MDEV down“

ifdown $MDEV
killall wpa_supPLICANT
;;

esac
```

➔ Jetzt noch die Konfiguration fürs WLAN nach `/etc/wpa_supplicant.conf` speichern:

```
# home network; allow all valid ciphers

network={

    ssid="FLAN6"

    key_mgmt=NONE

#----- WPA -----
#   key_mgmt=WPA-PSK
#   psk="very secret
#   passphrase"
}
```

für `ifup` noch das `wlan0` aktivieren und in `/etc/network/interfaces` eintragen:

```
# cat /etc/network/interfaces
# Configure Loopback

auto lo

iface lo inet loopback
iface wlan0 inet dhcp
```

USB-WLAN-Stick anschließen und der Rest sollte nun automatisch funktionieren:

```
usb 1-1.2: new high speed USB device using
lpc-ehci and address 3
usb 1-1.2: New USB device found,
idVendor=148f,
idProduct=5370
usb 1-1.2: New USB device strings: Mfr=1,
Product=2,
SerialNumber=3

usb 1-1.2: Product: 802.11 n WLAN
usb 1-1.2: Manufacturer: Ralink
usb 1-1.2: SerialNumber: 1.0

Compat-drivers backport
release: compat-drivers-2012-10-08-c

Backport based on linux-next.git next-
20121008

compat.git: linux-next.git

cfg80211: Calling CRDA
to update world regulatory
domain

usb 1-1.2: reset high speed USB device using
lpc-ehci and address 3
```



```
Registered led device: rt2800usb-phy0::radio
Registered led device: rt2800usb-phy0::assoc
Registered led device: rt2800usb-phy0::quality

usbcore: registered
new interface driver rt2800usb

rt2800usb 1-1.2:1.0: firmware: requesting rt2870.bin

wlan0: authenticate with 90:f6:52:c5:0c:58
wlan0: send auth to 90:f6:52:c5:0c:58 (try 1/3)
wlan0: authenticated
wlan0: associate with 90:f6:52:c5:0c:58 (try 1/3)wlan0: RX AssocResp from 90:f6:52:c5:0c:58
(capab=0x421 status=0 aid=1)
wlan0: associated

# lsmod

Module Size Used
by Not tainted

rt2800usb 12116 0
rt2800lib 48147 1 rt2800usb
crc_ccitt 1031 1 rt2800lib
rt2x00usb 7103 1 rt2800usb
rt2x00lib 29634 3 rt2800usb,rt2800lib,rt2x00usb
mac80211 225585 3 rt2800lib,rt2x00usb,rt2x00lib
cfg80211 151782 2 rt2x00lib,mac80211
compat 19454 4 rt2800usb,rt2x00lib,mac80211,cfg80211
led_class 2091 2 rt2x00lib,compat

#ifconfig

wlan0 Link encap:Ethernet HWaddr 7C:DD:90:0B:B0:FB

inet addr:192.168.6.100 Bcast:192.168.6.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:29 errors:0 dropped:0 overruns:0 frame:0

TX packets:2 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:10332 (10.0 KiB)
TX bytes:696 (696.0 B)
```

Verbindung steht !

Das war's, viel Spass beim nachvollziehen.



➔ www.MEsstechnik24.de

Deutschlands großer B2B Messtechnik-Web-Shop

Riesen-Spektrum an Tisch- und USB-PC-Oszilloskopen, Multimetern, Signalgeneratoren, Stromversorgungen, Spektrum-Analysatoren, TDR, Komponenten für die Messwerterfassung, Motorsteuerung, Schnittstellen.

Produkte namhafter Hersteller: Acromag, Adlink, Agilent, CableEye, Cleverscope, HBM, Icron, LabJack, Pico Technology, Pro-Dex OMS, Rigol, VSCOM/Vision Systems und viele mehr.



www.meilhaus.de



www.MEsstechnik24.de



MEILHAUS
ELECTRONIC

Meilhaus Electronic GmbH

Fischerstraße 2

82178 Puchheim/Germany

www.meilhaus.com

www.MEsstechnik24.de

TEL (0 89) - 89 01 66-0

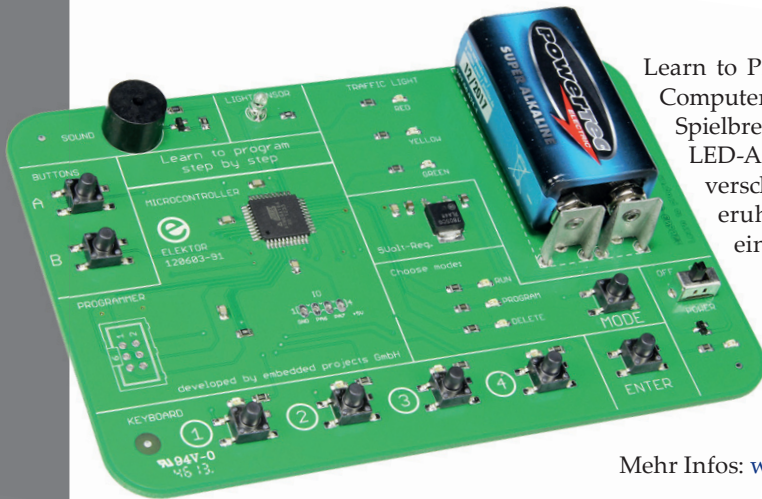
FAX (0 89) - 89 01 66-77

E-Mail sales@meilhaus.com

Marktplatz / Neuigkeiten

Die Ecke für Neuigkeiten und verschiedene Produkte

Learn to program - Programmieren ohne Tastatur



Learn to Program ist ein einfaches Spiel, das - im Stil der Anfangsjahre des Computerzeitalters - das Programmieren mit nur vier Tasten erlaubt. Auf dem Spielbrett befinden sich alle nötigen Ein-/Ausgabeeinheiten, wie etwa eine LED-Ampel, ein Helligkeitssensor und ein Piezopiepser. Damit lassen sich verschiedene Programme entwickeln, wie zum Beispiel eine einfache Eieruhr, ein Wecker (der morgens wie ein Hahn Kikeriki ruft) oder auch einfache Reaktionsspiele.

Im Vordergrund steht, dass man sich nur mit dem Handbuch und Spielbrett - komplett abgekapselt von allen anderen technischen Geräten und Hilfsmitteln - in so ein einfaches Programmiersystem eindenken kann - so wie man es machen musste, als es mit dem Computer langsam los ging.

Mehr Infos: www.embedded-projects.net/learntoprogram

SMD Prototypen-Bestückung von BGA-Bausteinen am Beispiel des Embedded Linux Boards GNUBLIN

Immer häufiger werden in Schaltungen insbesondere Linux-Boards SMD Bausteine oder BGAs verwendet. Moderne Prozessoren, AD-Wandler oder neue Speicher werden oft nur in sehr kleinen Bauformen angeboten. Dort, wo für Prototypen früher der Lötcolben und Lötzinn gereicht haben, benötigt man jetzt SMD-Lötpaste, eine passende Schablone und etwas Erfahrung bzw. die notwendigen Tricks um solch ein Projekt erfolgreich durchzuführen.

Oft hört man von Entwicklern, die noch zu großen Respekt vor diesen Bausteinen haben und sich deshalb an solche Bauteile gar nicht erst herantrauen.

In diesem Workshop werden wir gemeinsam anhand praktischer Beispiele demonstrieren, wie man solche Hürden überwinden und Prototypen mit vermeintlich schwierigen Bausteinen per Hand bestücken kann.

Jeder Teilnehmer wird ein eigenes BGA Board aufbauen und anschließend in Betrieb nehmen.

In der verwendeten Schaltung kommt ein BGA, QFN und TSOP54 Gehäuse zum Einsatz.

Ziel ist es, einen BGA ohne großen Aufwand löten zu können.

Jeder Teilnehmer wird mit Hilfe von unseren Experten seine eigene Platine bzw. BGA-Bausteine bestücken, das Board im Ofen löten und anschließend in Betrieb nehmen.

Programm:

- Einführung SMD Reflow löten / Vorstellung Equipment / Tips fürs Layout
- Vorführung diverser Techniken (SMD Schablonen-Druck, Löten, Platzieren)
- Praktische Einheit 1: Aufbau des Temperatursensors (Handbestückung mit LM75)
- Mittagspause
- Praktische Einheit 2: Aufbau des Embedded Linux Boards GnuBLIN
- Bestückung per Hand oder Teilbestückung der Mikrocontroller
- Optional Rest-Bestückung mit SMD-Automat
- Optische Kontrolle / Reflow Lötgang

- Inbetriebnahme des Linux Boards
- Fragen / Diskussion / Erfahrungsaustausch

Max. 10 Teilnehmer

Veranstaltungsort/-termin:

Augsburg 15.05.2014, 09.00 bis ca. 17.00 Uhr bei embedded projects GmbH (ausgebucht)

Die Schulung soll noch öfters stattfinden. Bei Interesse melden, dann setzen wir neue Termine fest.

Teilnahmegebühr (inkl. MwSt.): 349 €

Im Preis sind jeweils Material, Seminarunterlagen, Mittagessen sowie Dokumentation und Aushändigung eines Teilnahmezertifikats inbegriffen.

Jeder Teilnehmer erhält:

- Platine + Material für 1 x GNUBLIN und 1 x Temperatursensor
- Stencil Fix zum einfachen Bedrucken von SMD / BGA Platinen
- Handout

DAS ORIGINAL SEIT 1994
PCB-POOL[®]
Beta LAYOUT

Kostenlos!

Edelstahl SMD-Schablone
bei jeder PCB Prototyp-Bestellung
inklusive

EAGLE: Kalkulationsbutton
pcb-pool.com/download_button
20% RABATT! auf Ihre erste Bestellung

Alle eingetragenen Warenzeichen sind eingetragene Warenzeichen der jeweiligen Hersteller!



PCB-POOL[®] ist ein eingetragenes Warenzeichen der Beta LAYOUT GmbH

www.pcb-pool.com

25 Jahre Beta
LAYOUT
create : electronics

PROTOTYPES

Beta LAYOUT

Entwerfen, Bestellen, Anfassen

3D-Druck online

mit neuester
Lasersinter-Technik

**3D-Modelle und Gehäuse im
Hi-Tech Lasersinterverfahren**

Die Vorteile:

- Hohe Präzision
- Glatte Oberflächen
- Feine Strukturen
- Flexibel bei Wandstärken
von nur 0,4 mm - 2 mm

www.beta-prototypes.com

25 Jahre Beta
LAYOUT
create : electronics

Interesse an einer Anzeige?

info@embedded-projects.net

→ firma.embedded-projects.net

DAS HARDWARE FOR YOUR PROJECTS-PORTAL



In unserem Online-Shop finden Sie eine große Auswahl verschiedenster Mikrocontrollerboards, Programmer, Debugger u.v.m.

→ shop.embedded-projects.net

Unser Büro in Augsburg besteht aus leidenschaftlichen Entwicklern. Sprechen Sie uns an, wir finden eine Lösung für Ihr Problem.

→ projekte.embedded-projects.net



Speziell für Studenten und Hochschulen, bieten wir diese Ausbildungsinitiative an. Mikrocontrollerboards für den kleinen Geldbeutel.

→ student.embedded-projects.net



Holzbachstraße 4, D-86152 Augsburg
Tel +49 (0) 821 279599-0
Fax +49 (0) 821 279599-20
info@embedded-projects.net



embedded projects GmbH
HARDWARE FOR PROJECTS

FIND

www.f-y-e.de

your engineer

Der Experten-Wegweiser
zu Ihrem Elektronikentwickler

Elektronik- / Softwareentwicklung

Layout

Mechatronik

Bestücker / EMS-Dienstleister

EMV-Dienstleister

Find-Your-Engineer ist ein persönliches Empfehlungsnetzwerk. Firmen die Elektronik-Experten suchen, wenden sich bitte direkt an:

Markus Kessler
kontakt@find-your-engineer.de

*Mit der besten
Empfehlung!*

STX0x570x650x72SP0x640x610x73SP
0x760x650x720x730x730x650x680x74SP

... ist richtig bei uns! Bewirb Dich
als Embedded Entwickler (m/w)! **MIXED
MODE**

www.mixed-mode.de
technik.mensch.leidenschaft

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:

journal@embedded-projects.net

Regelmäßig

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 [2] in eine Liste für die gesponserten Abos oder ein Spendenabo 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 über einen Online - Shop [2] beziehen.

[1] Internetseite (Anmeldeformular gesponserte Abos):
<http://journal.embedded-projects.net>

[2] Online - Shop für Journal:
<http://www.embedded-projects.net>

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 / 2795990 oder sendet eine E-Mail an die oben genannte Adresse.

Impressum

embedded projects GmbH
Holzbachstraße 4
D-86152 Augsburg
Telefon: +49(0)821 / 279599-0
Telefax: +49(0)821 / 279599-20

Verteilt durch:



Veröffentlichung: 4x / Jahr
Ausgabenformat: PDF / Print

Auflagen Print: 2500 Stk.
Einzelverkaufspreis: 1 €

Layout / Satz: EP
Druck: flyeralarm GmbH
Titelbild: elektor

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.

Dies ist ein Open Source Projekt.



embedded projects GmbH
Holzbachstraße 4
D - 86152 Augsburg

Name / Firma

Straße / Hausnummer

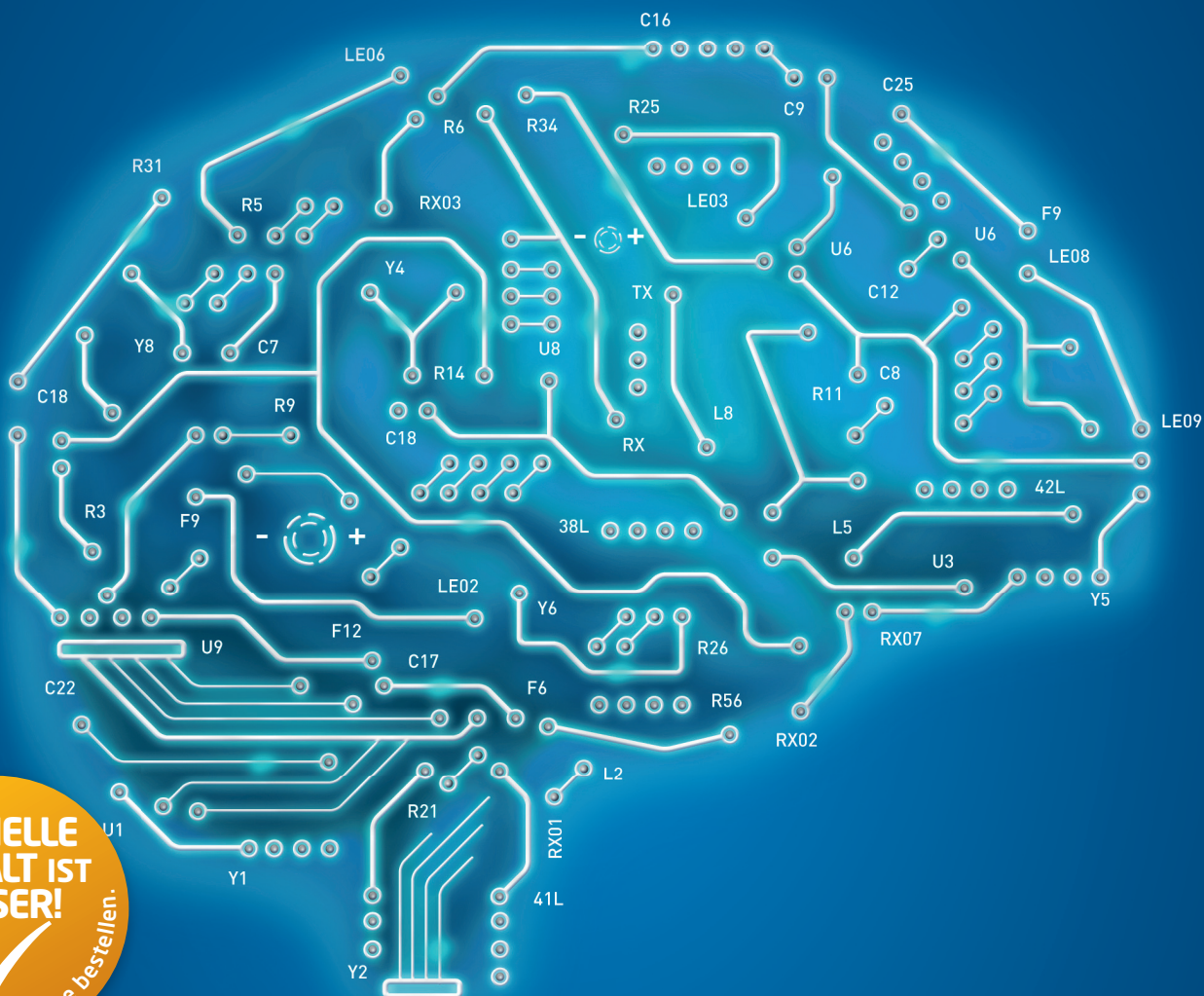
PLZ / Ort

Email / Telefon / Fax

- Ich möchte jede zukünftige Ausgabe erhalten
- Wir möchten als Hochschule / Ausbildungsbetrieb jede weitere Ausgabe bekommen. Bitte gewünschte Anzahl der Hefte pro Ausgabe ankreuzen. 5 10
- Ich möchte im embedded projects Journal werben oder eine Stellenanzeige aufgeben. Bitte schicken Sie mir Infomaterial, Preisliste etc. zu.

UNSER WICHTIGSTES WERKZEUG.

LEITERPLATTEN WEITER GEDACHT.



LEITON 
RECHNEN SIE MIT BESTEM SERVICE

Wir denken weiter. Denn als lösungsorientiertes Unternehmen möchten wir unsere anspruchsvollen Kunden immer wieder mit neuen Ideen und sinnvollen Innovationen begeistern. Unsere Vision, **ein führender Online-Anbieter von Leiterplatten** zu sein, haben wir nicht nur im Kopf - wir setzen sie auch um. Und zwar mit erstklassigen Leistungen in den Bereichen **Service, Fairness, Technologie, Auswahl, Komfort, Geschwindigkeit und Zuverlässigkeit**. Unsere Webseite bietet eine enorme Auswahl an Lösungen in der Online-Kalkulation sowie weiterführendes Wissen, Informationen und Entwicklerwerkzeuge. Und sollte einmal etwas sehr Ausgefallenes gefordert sein, finden wir gerne **Ihre persönliche Lösung**. Für unsere Kunden heißt das: Sie können bei LeitOn immer mit bestem Service rechnen.

LeitOn GmbH

www.leiton.de

kontakt@leiton.de

Info-Hotline +49 (0)30 701 73 49 0