

microSRCP

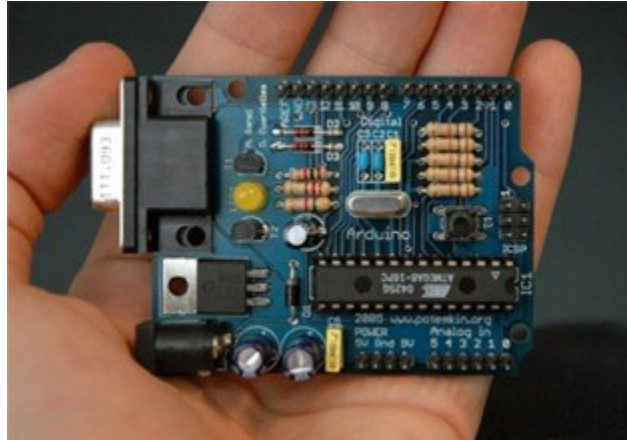


Photo by Nicholas Zambetti, www.arduino.cc

Modelleisenbahnsteuerung auf Basis von
standarddisierten Microcontrollerboards

<http://developer.berlios.de/projects/microsrcp/>

Haftung

- Bei den Bauanleitungen bzw. irgendwelcher Software gibt es keine Haftung für irgendwelche Schäden oder Funktionsgarantie, bitte immer nur als Anregung auffassen.
- Ich haften nicht für Schäden, die der Anwender oder Dritte durch die Verwendung der Software oder Hardware verursachen oder erleiden. In keinem Fall haften ich für entgangenen Umsatz oder Gewinn oder sonstige Vermögensschäden die bei der Verwendung oder durch die Verwendung dieser Programme oder Anleitungen entstehen können.
- Und wer mit Strom umgeht, soll sich bitte bei höheren Spannung und Strömen der Gefahren bewusst sein. Modellbahn gehört VDE-technisch zur Kategorie Spielzeug, dementsprechend streng sind die Vorschriften.
- Siehe hierzu auch die Sicherheitshinweise des Fremo.
- Die Schaltungen und die Software werden als Anregung und Hilfe unter Modellbahnern veröffentlicht. Sie sind auf Grund von Beispielen aus dem Netz bzw. eigenen Ideen entstanden.
- Natürlich sind sinnvolle Anregungen, Fehlermeldungen und Verbesserungen zu den Schaltungen immer willkommen. Allerdings möchte ich aus gegebenem Anlass darauf hinweisen, dass ich leider keine Zeit für langwierige Diskussionen des Typs "Ich habe die Schaltung nachgebaut, warum funktioniert sie bei mir nicht" habe. Auch die Frage "Ich habe noch diesen oder jenen Chip in der Schublade, kann ich den auch verwenden", möge sich bitte jeder selbst beantworten.
- Bitte bei Problemen möglichst exakt schildern, was wie schief geht, am besten mit logfiles bzw. exakter Fehlermeldung. Dazu stellt das Projekt ein Forum zur Verfügung.

Danksagung

- Mein Dank gilt:
 - Dem Arduino Projekt (www.arduino.cc), für die Idee und Umsetzung der qualitativ hochstehenden Arduino Boards.
 - Dem OpenDCC (www.opendcc.de) Projekt und speziell Wolfgang Kufer für die Zurverfügungstellung der Sourcen aus dem OpenDCC Projekt.
 - Allen Personen welche Shields zu den Arduino Boards herstellen und so das interessante Projekt weiterbringen.

Inhalt

- Hardware
- Der I2C Bus
- Software

Hardware



Steuerung der Zentrale
mittels Computer

Übersicht

Stromkreise mit Lok



Signale (LED)

Rückmelder
(Reedkontakte)



Servos



Computer Netzwerk

(TCP/IP, SRCP)

Zentrale

1..n
Motorboards

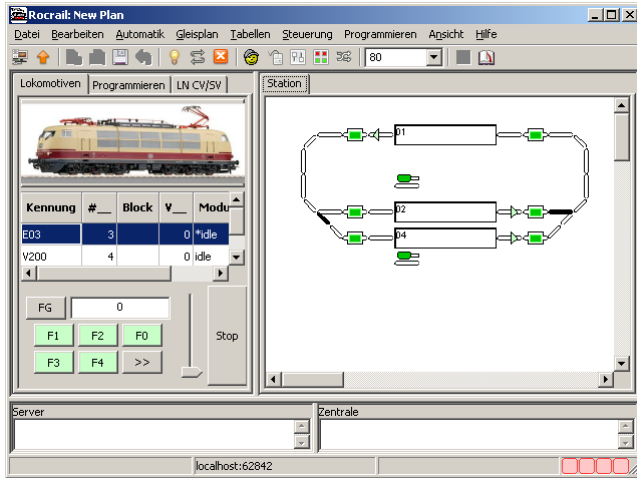
1..n
Geräteboards

I2C, als lokaler Gerätebus über 2 Kabel

- 1 bis n Microcontrollerboards welche über den I2C Bus kommunizieren steuern die Modelleisenbahn.
- Die Befehle dazu erhalten Sie von einem PC mit einer Modelleisenbahnsteuerung welche das SRCP Protokoll (z.B. RocRail) unterstützt.
- Elektronikkenntnisse sind von Vorteil, aber durch den einfachen Aufbau und Verwendung vorhandener Komponenten, nicht nötig.

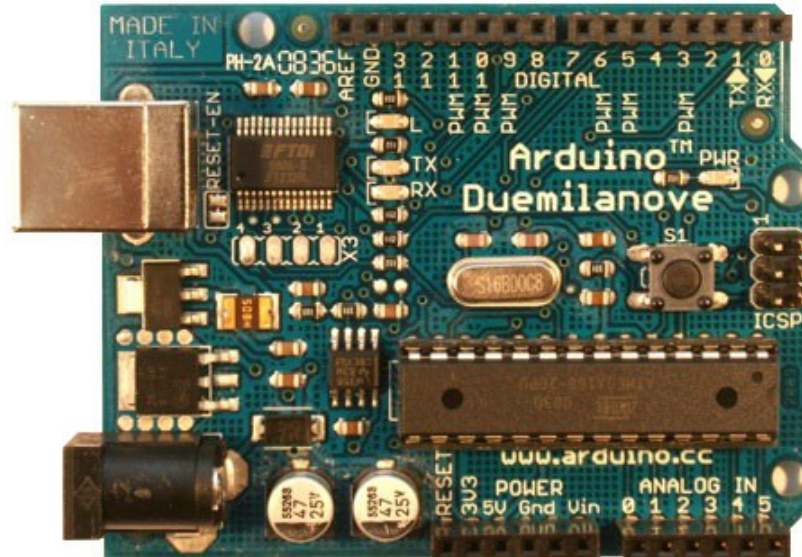
Der Versuchsaufbau

PC Software RocRail



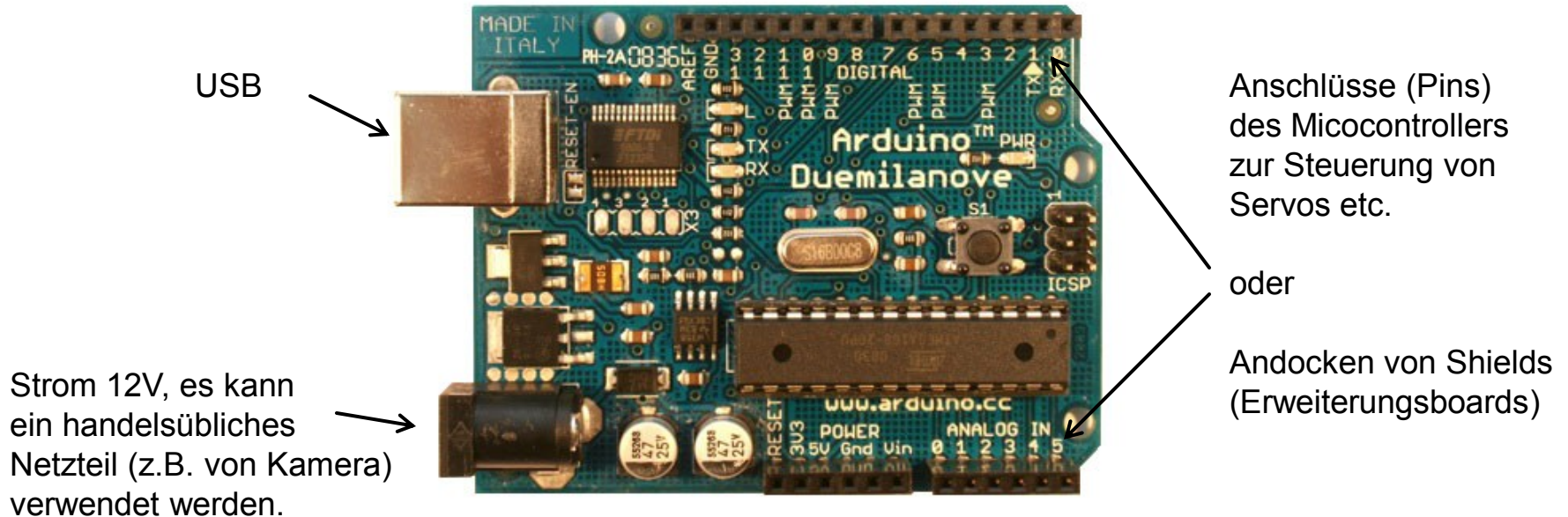
- Der Versuchsaufbau besteht aus einem PC mit RocRail und 3 Arduino Boards.
- Ein Board mit Ethernet Shield, eines mit dem Motor Shield und einer selbst gelöteten Platine zum Anschluss von Servos, Lichtsignalen und Rückmeldern.
- Damit lassen sich 2 Stromkreise, 6 Servos, 2 Lichtsignale und 8 Rückmelder auswerten.

Das Hauptboard (1)



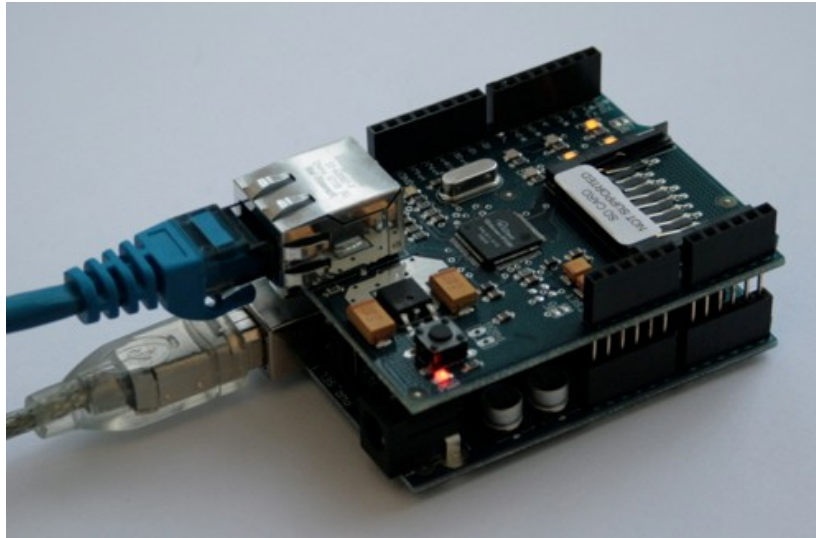
- Basis aller Boards bildet die Arduino Plattform -> www.arduino.cc
- Dabei handelt es sich um ein Microcontroller Board auf dem Erweiterungsboards, sogenannte Shields, aufgesteckt werden können -> <http://www.arduino.cc/en/Main/ArduinoShields>
- Boards können gekauft oder selbst hergestellt werden, die Informationen dazu sind frei erhältlich.

Das Hauptboard (2)



- Der grosse Chip ist der Microcontroller, über den USB Anschluss werden Sketche (Programme) geladen, eine 12V Stromanschluss sorgt für genügend Power.
- Alle Beine (Pins) des Microcontrollers sind nach aussen geführt und auf dem Board als Digital 0 – 13 und Analog 0 – 5 ersichtlich.
- Die Anschlüsse liefern genügend Strom um Servos, Lichtsignale mit LED direkt anzusteuern. Umgekehrt können an den Anschlüssen Momentkontakte abgefragt werden.

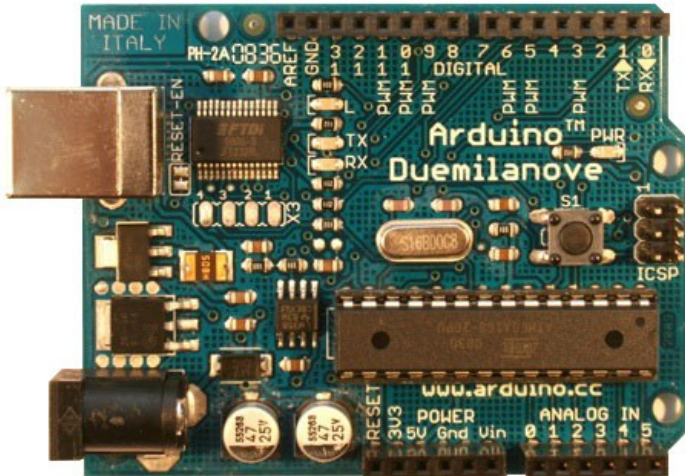
Die Zentrale



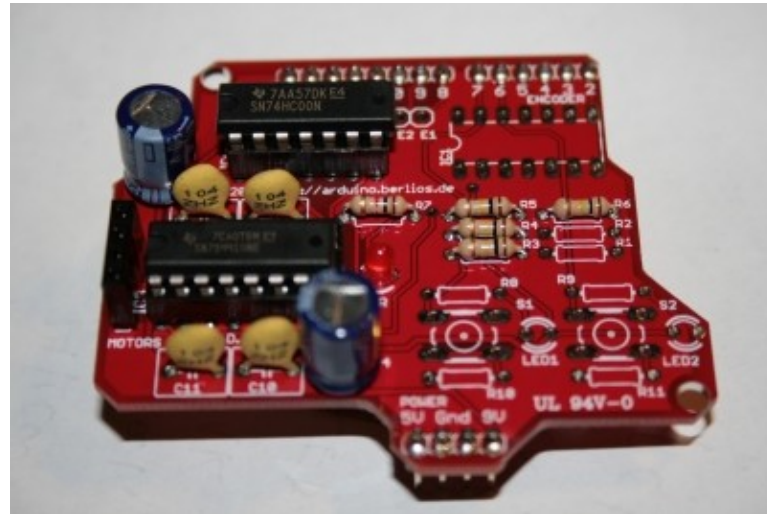
- Die Zentrale besteht aus einem Arduino mit Ethernet Shield.
- Darauf wird der Sketch SRCPServer geladen.
- Strom (5V!), TCP/IP und I2C Kabel anschliessen fertig!
- Details zum Board
 - <http://arduino.cc/en/Main/ArduinoEthernetShield>

Steuerung von Lokomotiven

Ein Arduino D. oder kompatibles Board



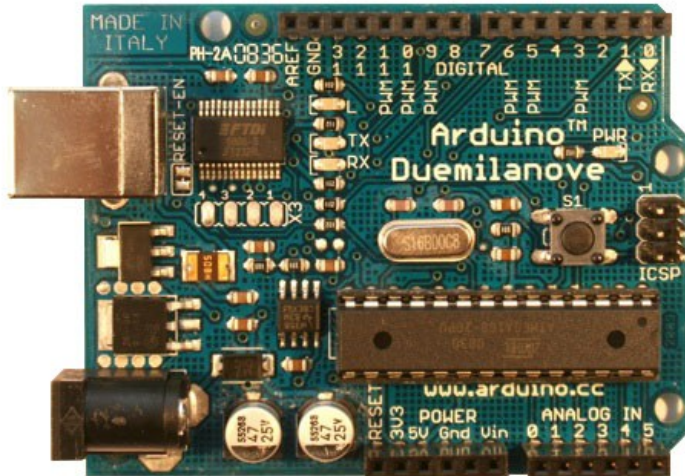
Arduino Motor Shield



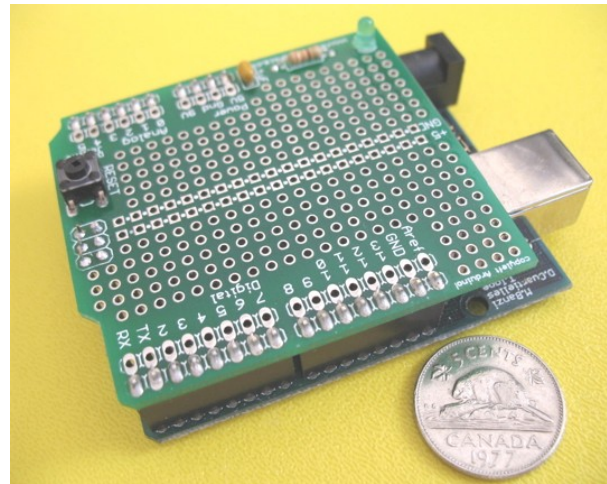
- Um Analoge Loks zu steuern wird der Arduino Motor Shield mit dem Sketch SRCPGLSlave verwendet
- Der Motor Shield hat eine Ausgangleistung von 2 x 2 Ampere, genügend um ein paar Lokomotiven zu steuern.
- Details zum Board
 - <http://arduino.cc/en/Main/ArduinoMotorShield>

Steuerung von Servos, Lichtsignalen, Rückmelder (1)

Ein Arduino D. oder kompatibles Board



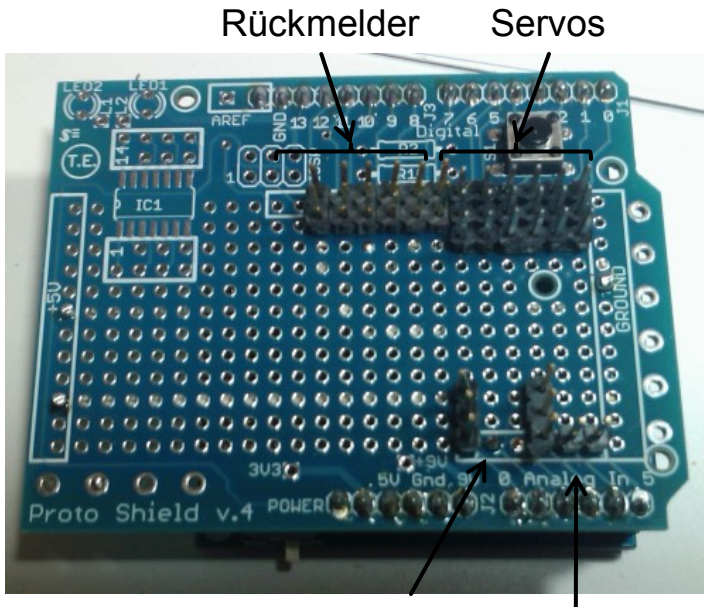
Arduino Proto Shield als Ausgangsplatine



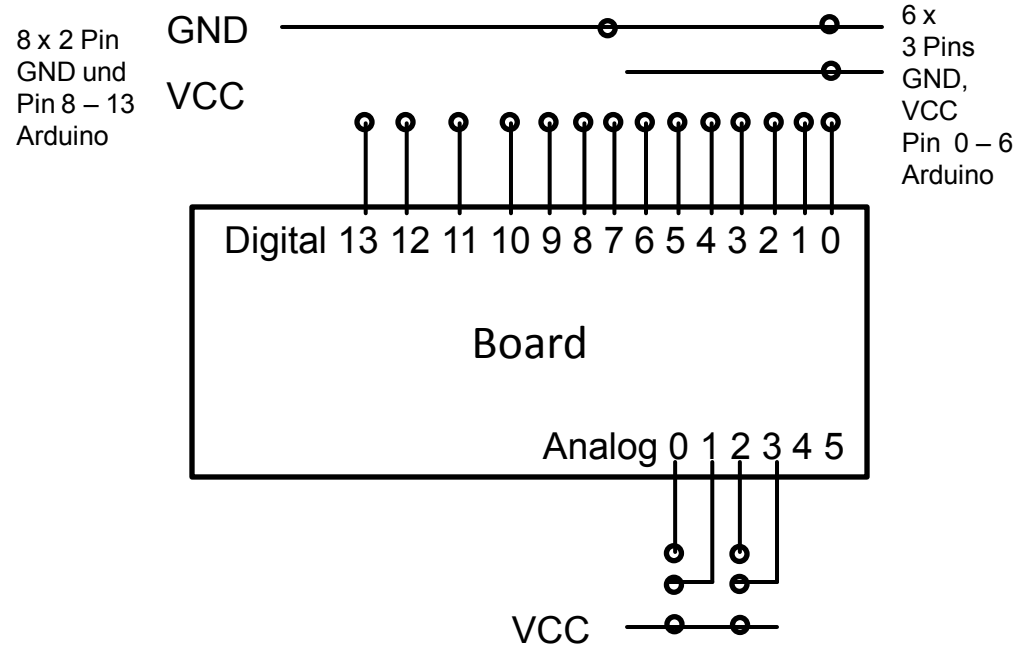
- Hier gibt es leider ein fertiges Board aber mit ein paar Lötarbeiten ist dieses einfach aufzubereiten.
- Der Sketch SRCPGASlave ist ausgelegt auf 6 Servos, 2 Lichtsignale und 7 (8) Rückmelder. Um diese anzuschliessen verwende ich einfache Crimp Stecker. Der Layout ist auf dem folgenden Slide ersichtlich.
- Details zum Board
 - <http://www.arduino.cc/playground/Tutorials/Protoshield>

Steuerung von Servos, Lichtsignalen, Rückmelder (2)

Mein Board



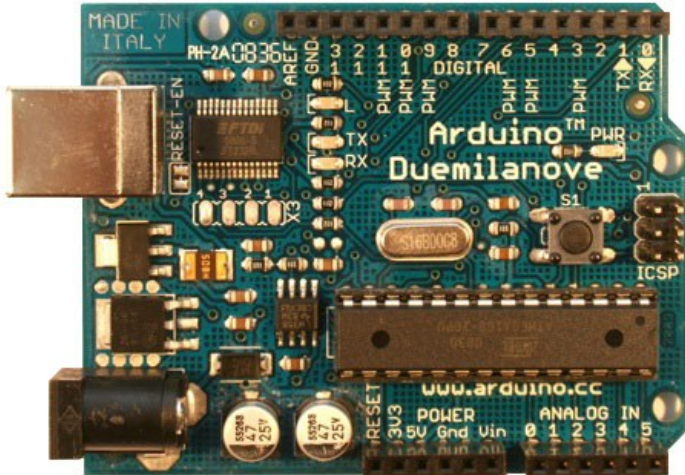
Lichtsignale und I2C Bus



- Die Belegung der Pins ist wie folgt:
 - Digital 0 – 5 für Servos (mit eigener Stromquelle versorgen!)
 - Digital 6 – 12 (13 ist leider von einer LED belegt) für Rückmelder
 - Analog 0 – 4 für Lichtsignale
- **ACHTUNG:** Relais, Lichtsignale mit Glühlampen nicht direkt anschliessen, zerstört den Microcontroller!

Wave Shield (Bahnhofsdurchsagen etc.)

Ein Arduino D. oder kompatibles Board



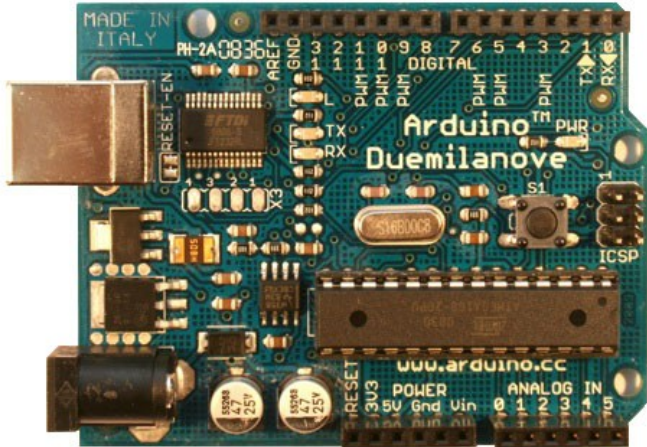
Adafruit Wave Shield



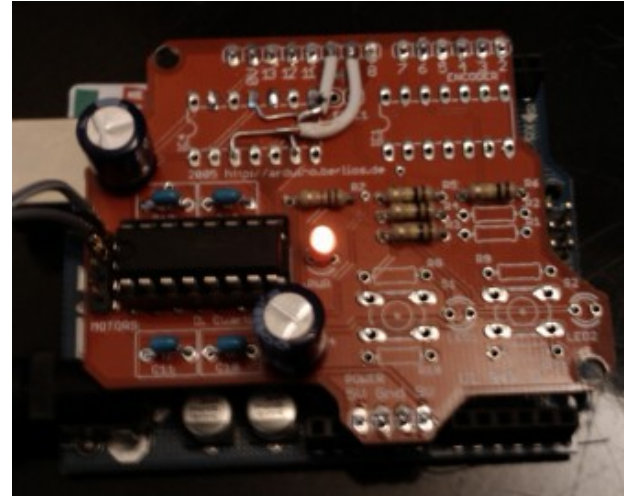
- Um die Anlage lebendig zu gestalten braucht es auch Bahnhofsdurchsagen und andere Geräusche mit dem Sketch SRCPWaveSlave und dem Wave Shield von Adafruit wird das einfach möglich.
- Die Waves werden wie Weichen oder Signale in der Steuerungssoftware eingetragen und können so manuell oder automatisch ausgelöst werden.
- Details zum Board und Link um z.B. Bahnhofsdurchsagen zu erstellen
 - <http://www.ladyada.net/make/waveshield/>
 - <http://www2.research.att.com/~ttsweb/tts/demo.php#top>

Digitale (DCC) Steuerung (1)

Ein Arduino D. oder kompatibles Board

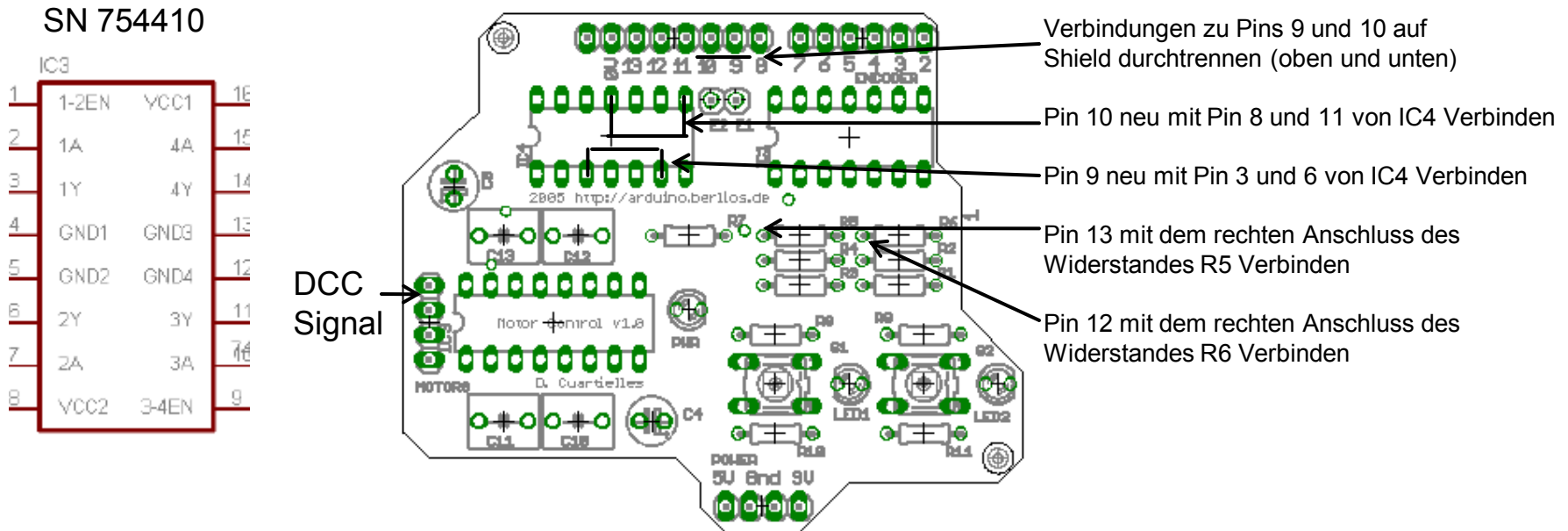


Arduino Motor Shield als DCC Booster



- Mit DCC steht ein standardisiertes Protokoll zur Steuerung von Lokomotiven und Zubehör zur Verfügung. Allerdings braucht es dazu separate Decoder, welche ausserhalb dieses Projektes liegen. Die Ansteuerung der Decoder ist, dank der Vorarbeiten im OpenDCC Projekt, aber einfach.
- Dazu wird der Sketch SRCPDCCSlave, ein Arduino und ein modifizierter Motor Shield benötigt.
- Link zum Board (2. Link = aktuelles und verwendetes Board)
 - <http://arduino.cc/en/Main/ArduinoMotorShield>
 - <http://www.nkcelectronics.com/freduino-arduino-motor-control-shield-kit.html>

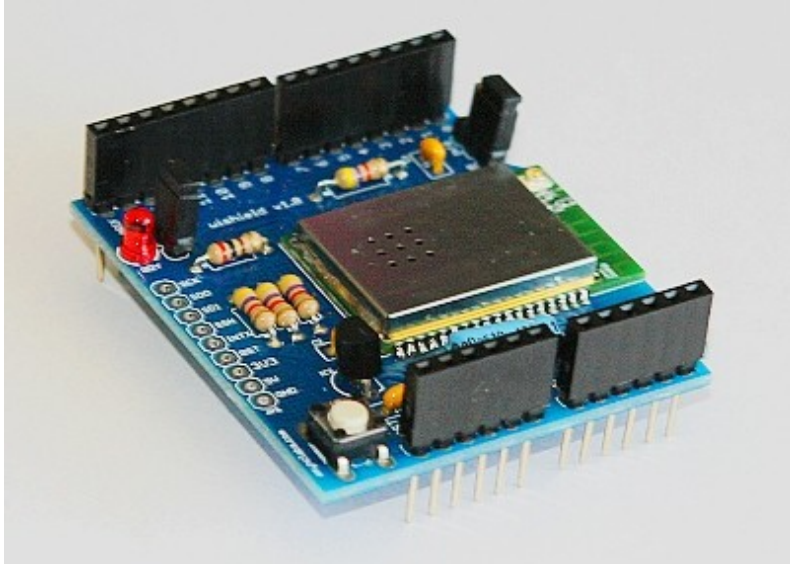
Digitale (DCC) Steuerung (2)



- Um das DCC Signal zu Verstärken braucht es Änderungen am Motor Shield. Hauptsächlich geht es darum, dass das PWM Signal nicht an die Anschlüsse 1-2EN und 3-4EN sondern an 1-4A, des Motortreiberchips (SN754410) geführt werden.
- Dazu muss der IC4 entfernt und ein paar Drahtbrücken gelegt werden, die Detail stehen oben.
- Wurde alles Ordnungsgemäss durchgeführt steht am oberen Motoranschluss das DCC Signal zur Verfügung.
- Der zweite Motoranschluss ist für das Programmgleis reserviert und im Moment ohne Funktion.
- **ACHTUNG:** Externe Stromquelle (=> 12V) anschliessen, bevor der Motoranschluss mit Gleis verbunden wird!

Weitere Boards - in Planung

Asynclabs WiShield

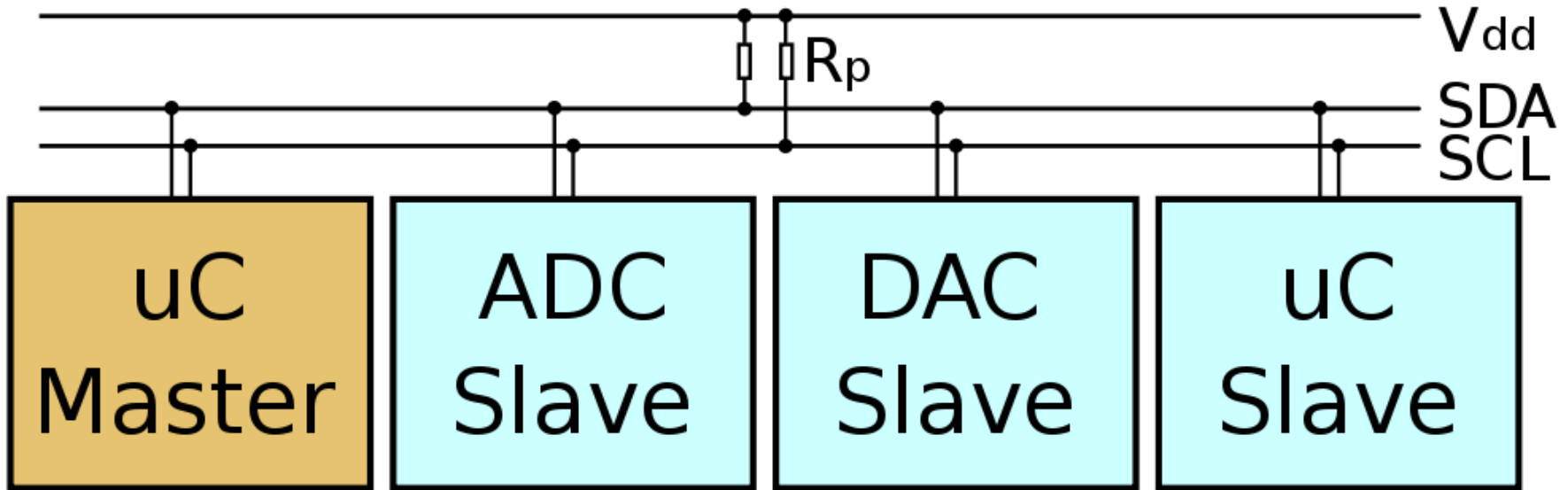


Weitere ...

- Statt dem Ethernet Shield könnte auch direkt das WiShield verwendet werden. Das würde die Verbindung via Computer Netzkabel erübrigen. Ein Wireless Router steht heute in praktisch jedem Haushalt (wie der PC ;-))
- Links zum Board/Software
 - <http://asynclabs.com/wiki/index.php?title=AsyncLabsWiki>

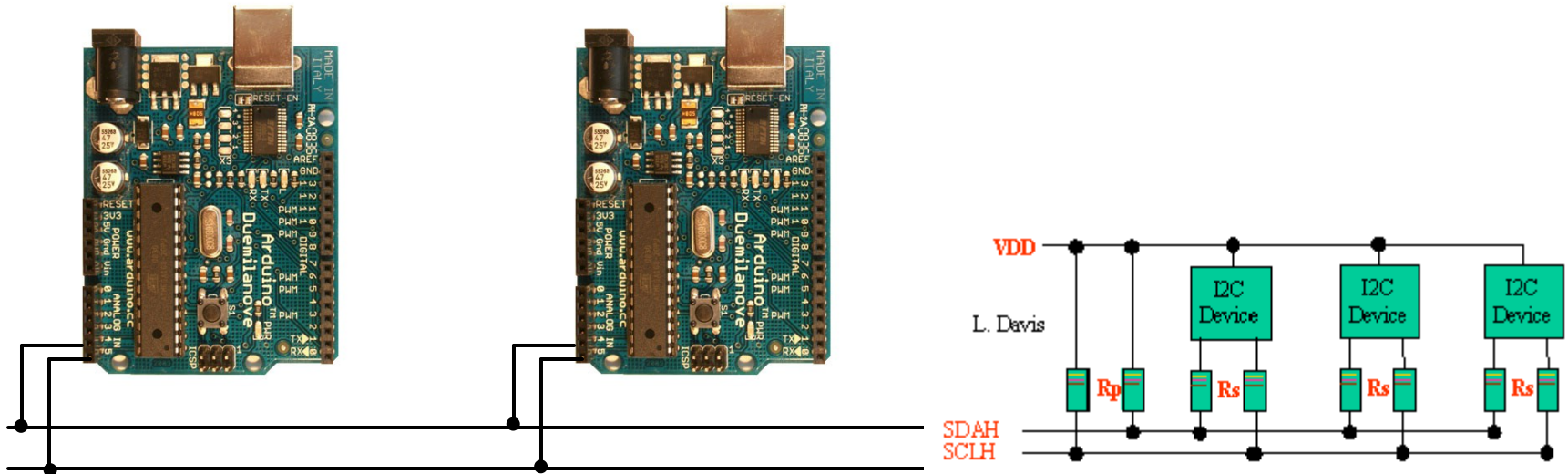
Der I2C Bus

Der I2C Bus



- In Wikipedia steht dazu (<http://de.wikipedia.org/wiki/I%C2%B2C>)
 - **I²C** (für **Inter-Integrated Circuit**, meistens gesprochen als *I-Quadrat-C* oder englisch *I-squared-C*) ist ein von [Philips](#) Semiconductors (heute [NXP Semiconductors](#)) entwickelter [serieller Datenbus](#).
- In Anbetracht der Datenmenge (ca. 6 Bytes pro Befehl) und der Einfachheit ist dieser für die Steuerung ideal geeignet.
- Schwachstelle ist die Länge des Buses (ein paar Meter), aber beim Suchen im Internet stösst man auch auf Längen von 25 Metern und mehr.

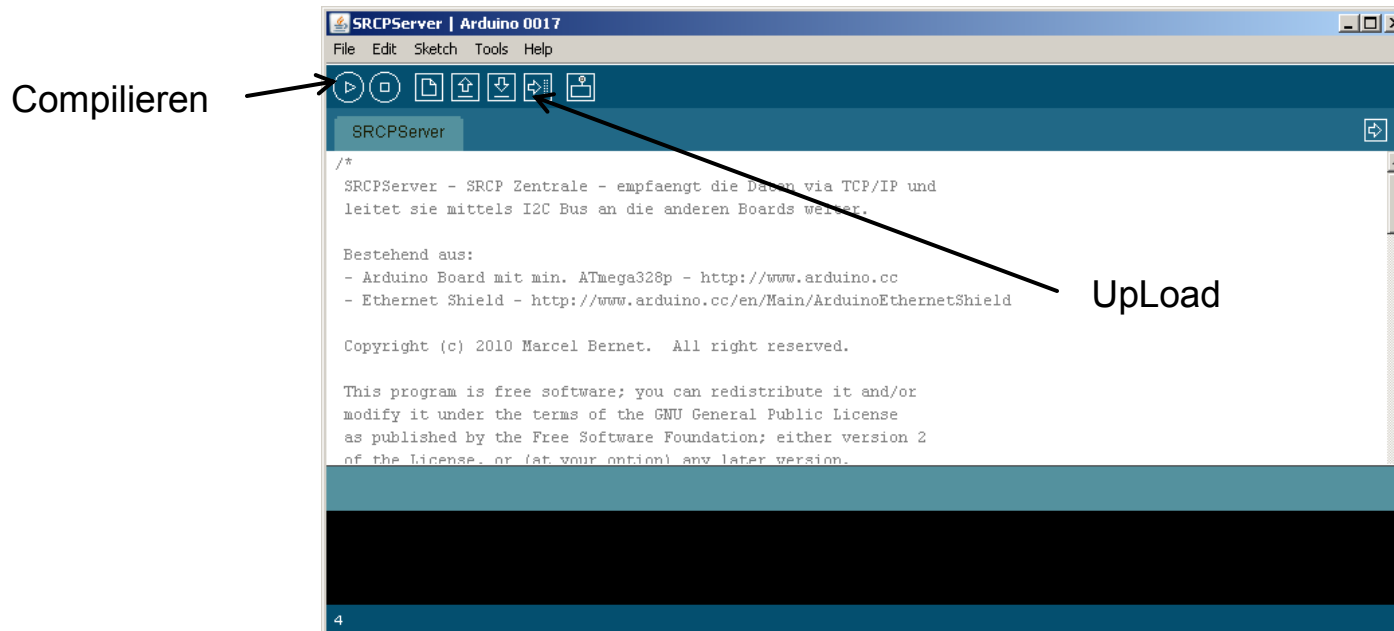
Die Arduino Plattform und der I2C Bus



- Der Arduino stellt das I2C Bus Signal auf den Pin 4 und 5 zur Verfügung. Für eine Versuchsschaltung genügt es diese Pins zu Verbinden. Für den Dauereinsatz sollten entsprechende Vorwiderstände geschaltet werden (siehe Grafik rechts).
- Nützliche Links
 - <http://www.elektronik-magazin.de/page/der-i2c-bus-was-ist-das-21>
 - <http://www.uchobby.com/index.php/2008/09/16/introduction-to-i2c/>
 - <http://www.arduino.cc/playground/Learning/I2C>
 - <http://www.cc2net.de/Tips/tips.html>

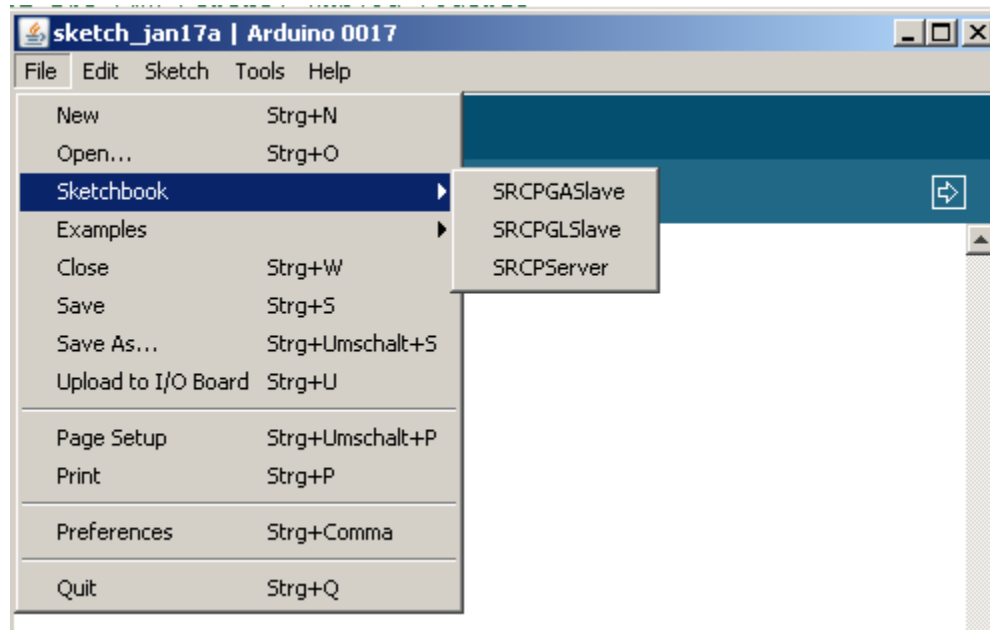
Software

Entwicklungsumgebung Arduino



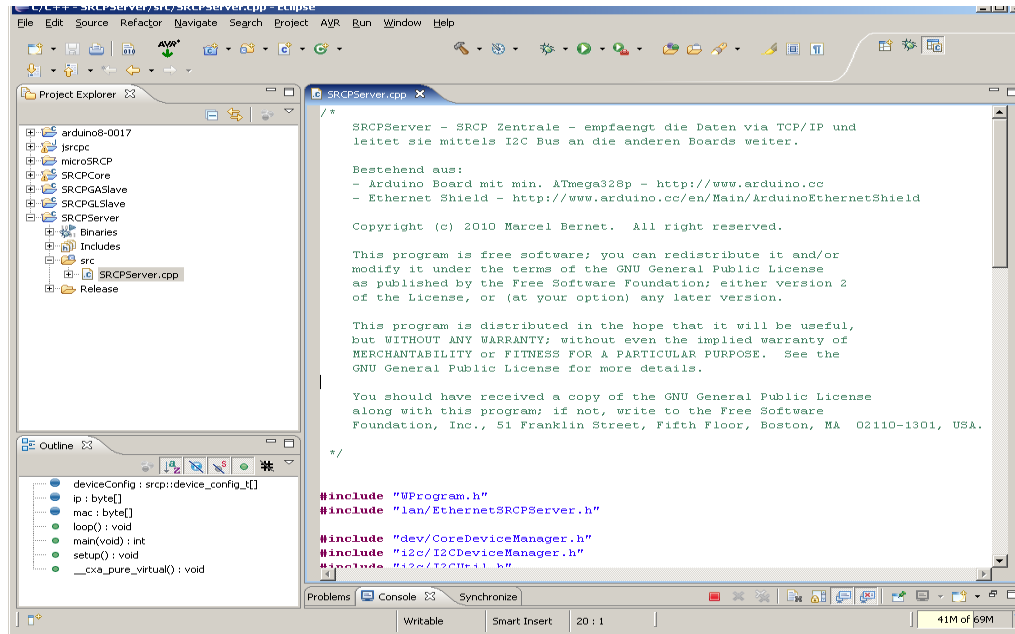
- Die Entwicklungsumgebung wird benötigt um die Steuerungssoftware, sogenannte Sketches, zu compilieren und in die Boards zu laden (= Upload)
- Die Installation ist Ausführlich unter den folgenden Links beschrieben:
 - <http://arduino.cc/en/Main/Software>, <http://www.freeduino.de/books>
- Die microSRCP Erweiterungen kann als ZIP-Datei (microSRCP_4_arduino.zip) von der Projektseite geladen werden.
- Deren Installation ist im ReadMe.txt beschrieben.

Die Sketches



- Wurde alles richtig installiert sollten vier Sketches zur Verfügung stehen:
 - SRCPGASlave – Steuerung für Servos, Lichtsignale, Rückmelder
 - SRCPGLSlave – Steuerung für 2 analoge Stromkreise
 - SRCPWaveSlave – Wiedergabe von Sounds
 - SRCPServer – SRCP Zentrale
- Weitere Informationen stehen im Sourcecode.
- Ein Sketch für DCC steht leider nicht zur Verfügung, wegen Problemen mit der Version des gcc Compiler in der Arduino IDE.

Entwicklungsumgebung Eclipse



- Alternativ kann Eclipse (www.eclipse.org) verwendet werden.
- Die Installation ist nicht ganz einfach und hier (<http://www.arduino.cc/playground/Code/Eclipse>) beschrieben.
- Auch sind die Arduino Sourcen als Eclipse Projekt abzulegen (bei mir arduino8-0017, mit der Library libarduino8.a)
- Die benötigten microSRCP Eclipse Projekte können als Sourcen-[Version].zip downgeloadet werden.

Ein Wort zum Schluss

- Es handelt sich hier um ein sehr junges Projekt, dass von einer Person (mir) erstellt wurde.
- Bitte nur Fragen zum microSRCP Projekt im Forum stellen, meine Zeit ist sehr begrenzt und die Antwort kann dauern!
- Für alle anderen Fragen rund um Arduino gibt ein ausgezeichnetes Mehrsprachiges Forum unter:
 - <http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl>