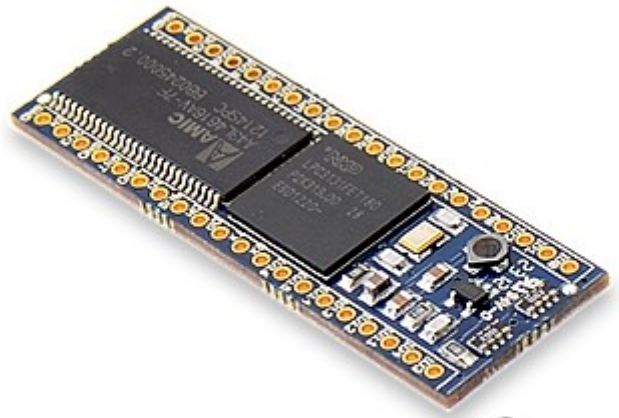
Anwendungen mit RaspberryPi und dem Elektor Linux Board (GNUBLIN)

Benedikt Sauter, sauter@embedded-projects.net

Gliederung der Folien

Inhalt:

- Ursprung / Hintergrund
- Komponenten für Anwendungsentwicklung
- Evaluationsboards
- Module / Erweiterungen
- Schritt für Schritt
- Beispielanwendungen
- Demo



Zur Person:

- Benedikt Sauter
- Informatiker mit Hardware
- Autor / Entwickler Elektor Linux Board / Gnublin
- Embedded Linux
- Embedded Systeme

elektor

GNUBLIN

Embedded Linux Anwendungen

Warum Embedded Linux für Mikrocontroller?

- Früher Rechner Technik
- Einzug von ARM Prozessoren im Labor
- Erste Embedded Linux Versuche
- Problem: Teure geschlossene Boards

Das Projekt GNUBLIN:

Embedded Linux

Los ging es bei GNUBLIN:

- 2-Lagen Board
- Hard- und Software lernen und verstehen
- Bootloader, Kernel
- Erste Anwendungen
- Für Studenten
- Elektor Linux Board

  (Einführung von Geschichte bis zur Anwendung)

GNUBLIN

Embedded Linux Hype

- Embedded Linux Markt nimmt fahrt auf
- Ankündigung RaspberryPi
- Viele neue Interessenten
- Markt öffnet sich
- Nicht nur Techniker / Nerds
- Wunsch: Interessantes Projekt für Einsteiger / Mikrocontroller

GNUBLIN

In der Zwischenzeit

- Mehrere Boards
- Wiki mit vielen Infos
- Erweiterungsmodule / Platinen
- Diverse Programme
- Anbindungen an RaspberryPi

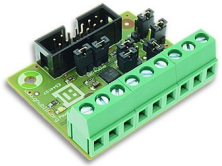
  (da Studenten in der Vorlesung damit arbeiteten)

GNUBLIN

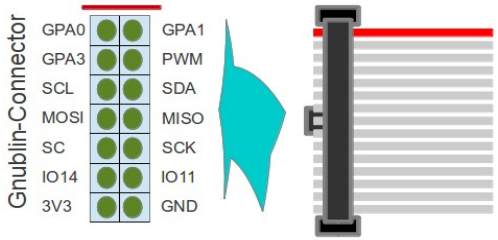
Was ist das GNUBLIN Toolkit?

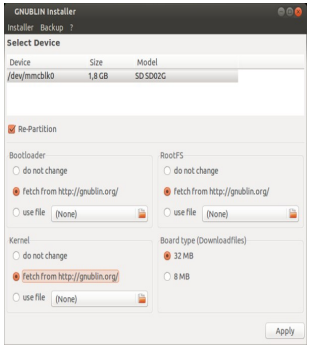
- Embedded Linux Entwicklungsumgebung
  für verschiedene Embedded Systeme
- Kernel, Bootloader, Distribution
- Gnublin Installer zum Erstellen von SD-Karten
- Viele Module und Erweiterungsplatinen (für Prototypen)
- Schnittstellen zu RaspberryPi, BeagleBone, …
- API: C++ und Python (weitere folgen)
- Gnublin Tools (Kommandozeilen Tools)

Embedded Linux Boards



Erweiterungsmodule





Installer

Kommandozeilen Tools

Intern GPIO Ausgang

```
#include "gnublin.h"

#define BOARD EXTENDED

int main()
{
   gnublin_gpio gpio;

   gpio.pinMode(3,OUTPUT);

   while(1){
     gpio.digitalWrite(3,HIGH);
     sleep(2);
     gpio.digitalWrite(3,LOW);
     sleep(2);
   }
}
```

Programmierung

http://wiki.embedded-projects.net

„Open-Source“ darf kommerziell verwendet werden!

→ **Lizenzbedingungen beachten!**

GNUBLIN
Los geht's

elektor

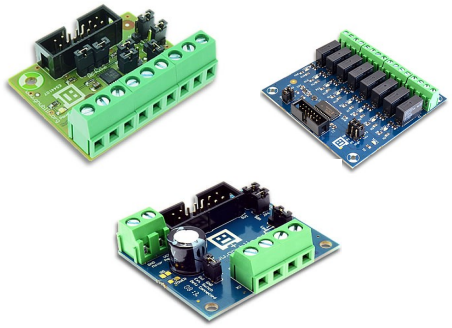
GNUBLIN

Board

Module

Tools

API



Intern GPIO Ausgang

```
#include "gnublin.h"

#define BOARD EXTENDED

int main()
{
   gnublin_gpio gpio;

   gpio.pinMode(3,OUTPUT);

   while(1){
     gpio.digitalWrite(3,HIGH);
     sleep(2);
     gpio.digitalWrite(3,LOW);
     sleep(2);
   }
}
```

Vorgehensweise:

- Auswahl der Komponenten
- Verbinden per Flachbandkabel
- Funktionstests mit Gnublin Tools
- Programmierung mit der API

→ Schritt für Schritt

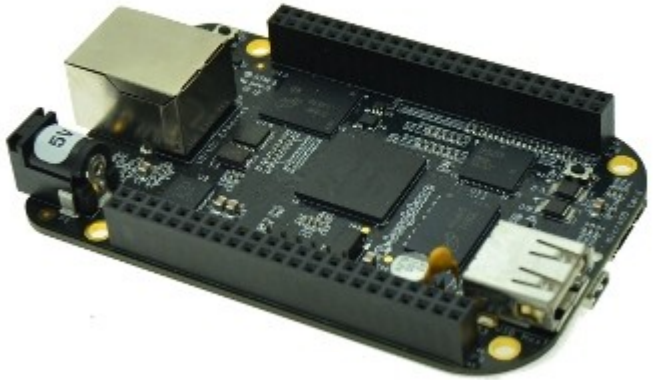
Evaluationsboards / Prozessoren

elektor

GNUBLIN

Boards:

RaspberryPi Boards
ARM1176JZF-S



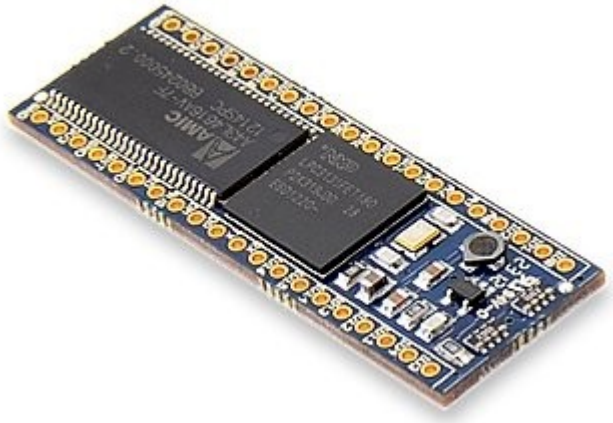
BeagleBone Black
Cortex-A8

GNUBLIN

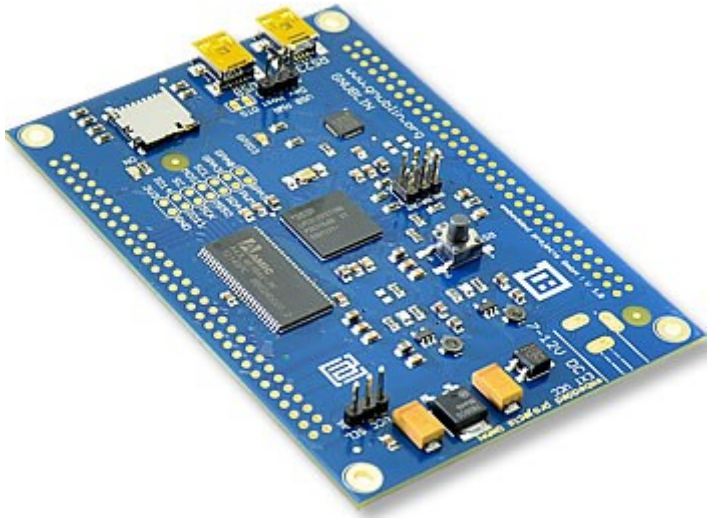
GNUBLIN / Elektor
ARM926EJ-S

Elektor Linux Board

Gnublin Standard

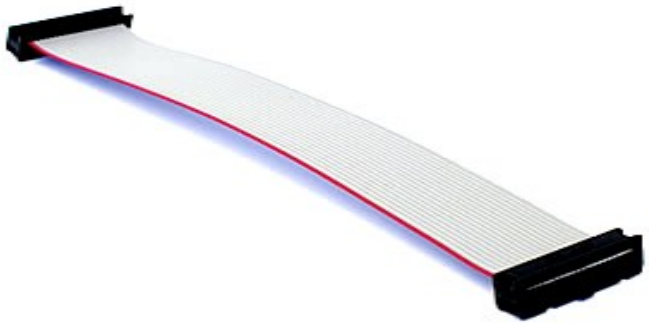


Gnublin DIP

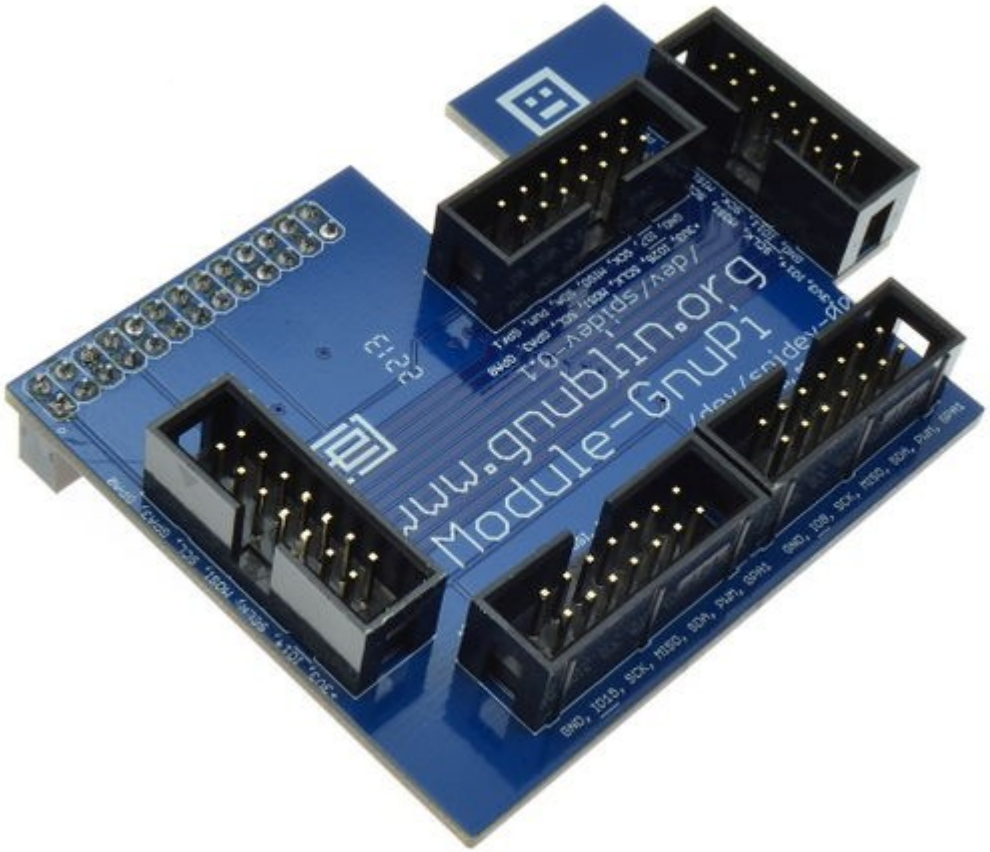


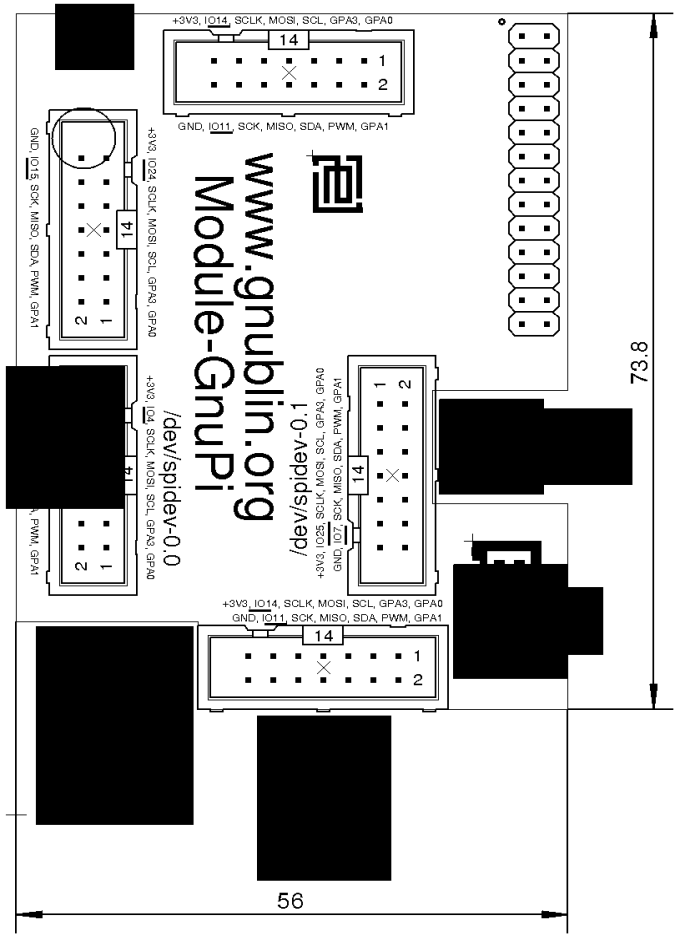
Gnublin Extended

Anbindung an GNUBLIN

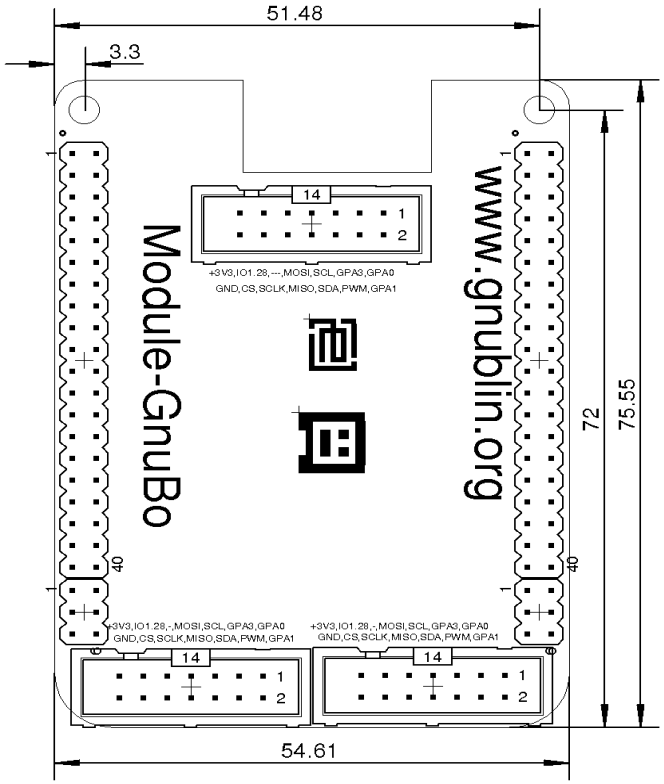


Anbindung an RaspberryPi





Anbindung an BeagleBoneBlack



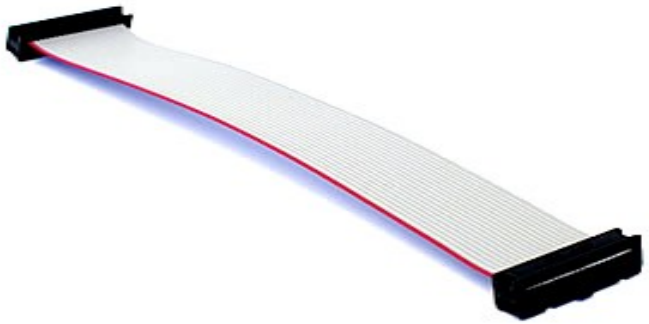
Baukasten mit Module

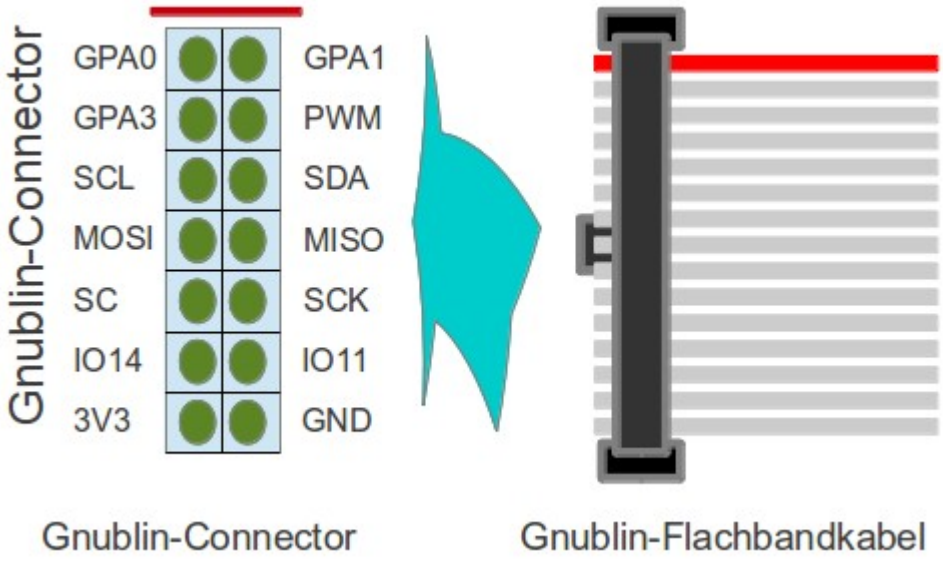
Embedded System = Schnittstellen

- Embedded bedeutet zu 99% Schnittstellen
- Module mit Standardfunktionen
- Für jedes Modul gibt es ein Gnublin Tool (Kommandozeilen Tool)
- Gnublin API für C++ und Python Entwicklung

Verbindung zu den Boards

- Standard Stecker
- Flachbandkabel





Temperatur Sensor:

- -55 bis 125 Grad Celsius
- max 8 Sensoren
- Baustein: LM75

Gnublin Tool:

```
gnublin-lm75 -h
```

GPIO Expander

- 16 zusätzliche Ports
- Ein- oder Ausgang
- Baustein: PCA9555

Gnublin Tool:

```
gnublin-pca9555 -h
```

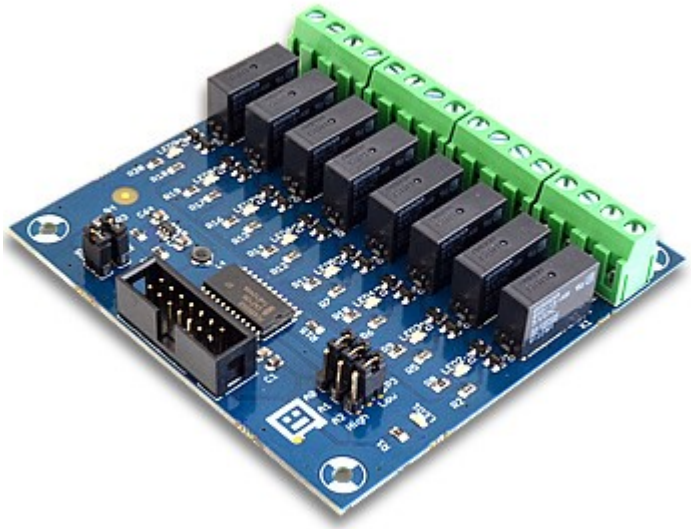
Display (4x20):

- 4x20 Zeichen
- 5 x Taster
- Max. 8 Stück pro Board
- Baustein: PCA9555

Gnublin Tool:

```
gnublin-lcd -h
```

Relais-Karte (8 Fach)



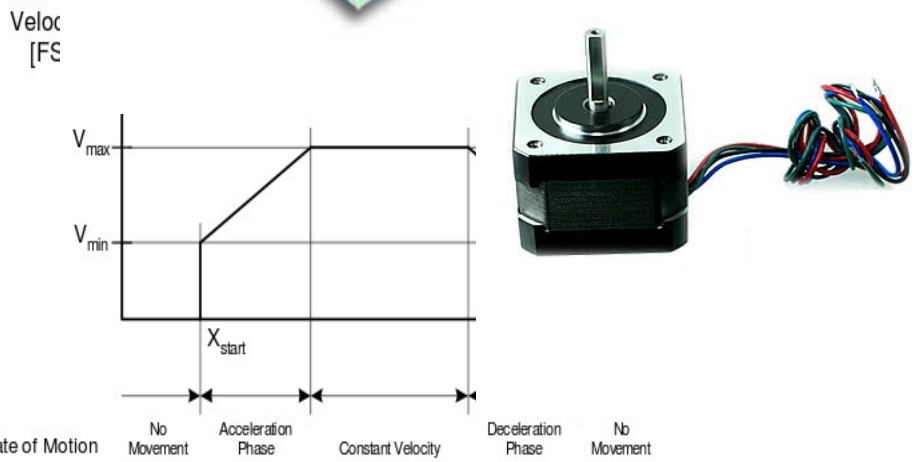
- 8 x Relais
- 230V AC / 5A bzw. 30V DC *
- Baustein: PCA9555

Gnublin Tool:

```
gnublin-relay -h
```

* Achtung nur für Fachkräfte bei 230V!

Schrittmotor:



- Schrittmotor
- Geschwindigkeit, Beschleunigung
- Eine Adresse per Jumper

  andere per Programmierung
- Baustein: TMC222

Gnublin Tool:

```
gnublin-step -h
```

WLAN

- WEP / WPA / WPA2
- Verbindung zu Access Point
- Integriert in Kernel

Gnublin Tool:

```
gnublin-wlan -h
```

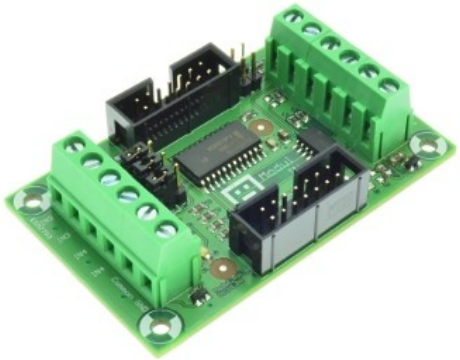
Webcam:

- Live-Stream
- Snapshot
- Standard Video Klasse (USB)

Gnublin Tool:

```
gnublin-webcam -h
```

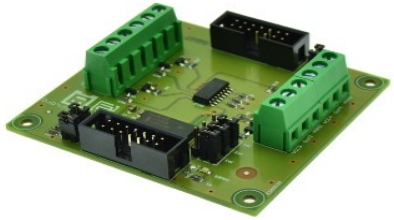
Hutschiene:



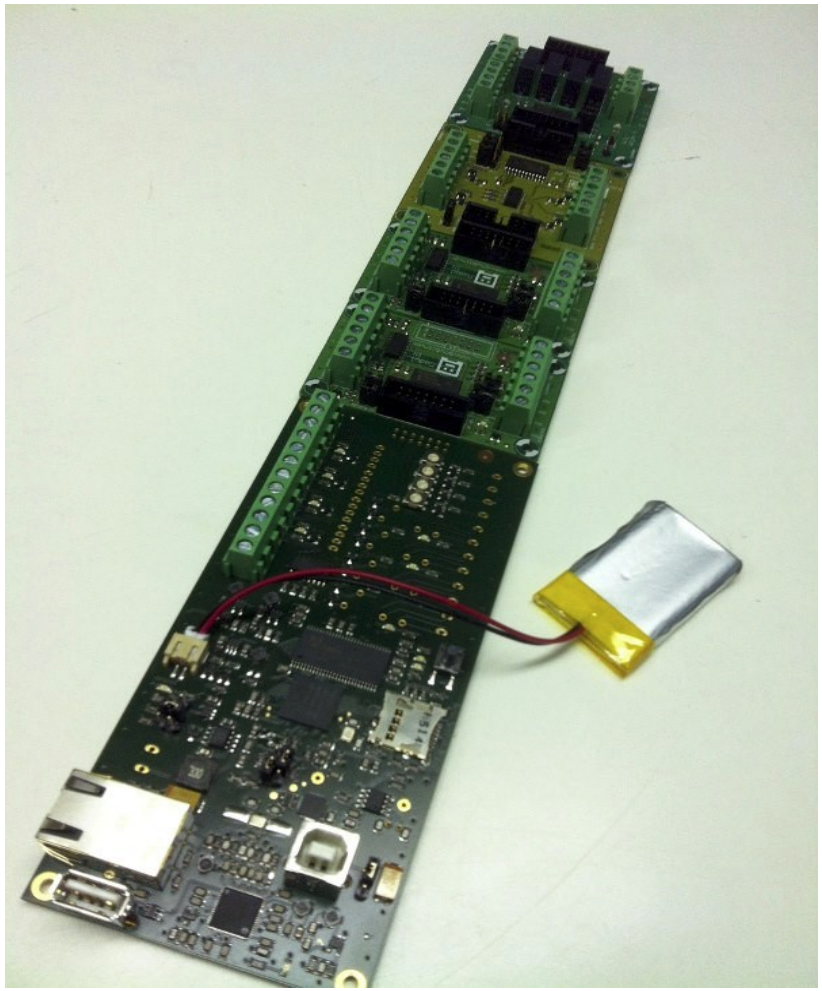
FET Eingangskarte
(bis 24V mit Optokoppler)

Relais Karte
(4 Kanal)

Digital zu Analog
Wandler



FET Ausgang
(mit Optokoppler)



Für Phoenix Hutschienengehäuse

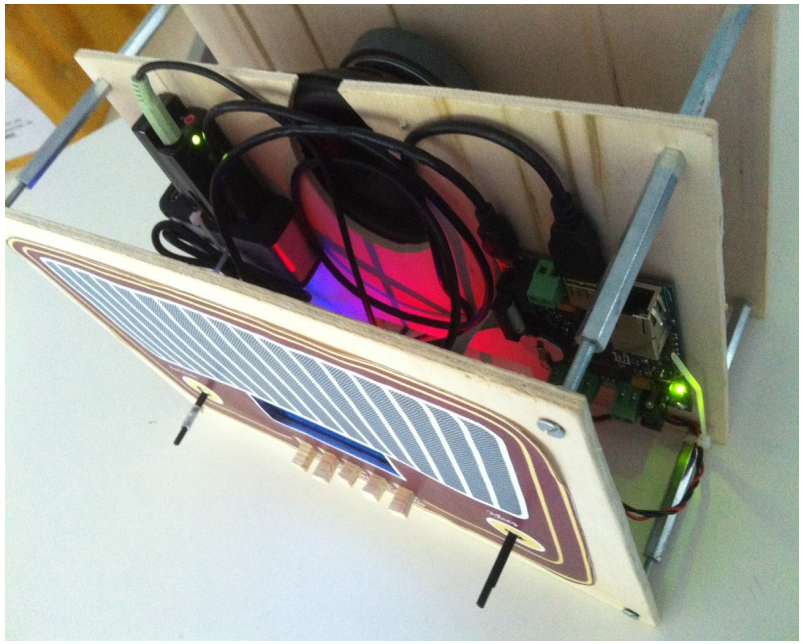
Schaltpläne und Platinendaten:

- Eagle Schaltplan + Layout
- Alles Open-Source

https://github.com/embeddedprojects/gnublin-schematics

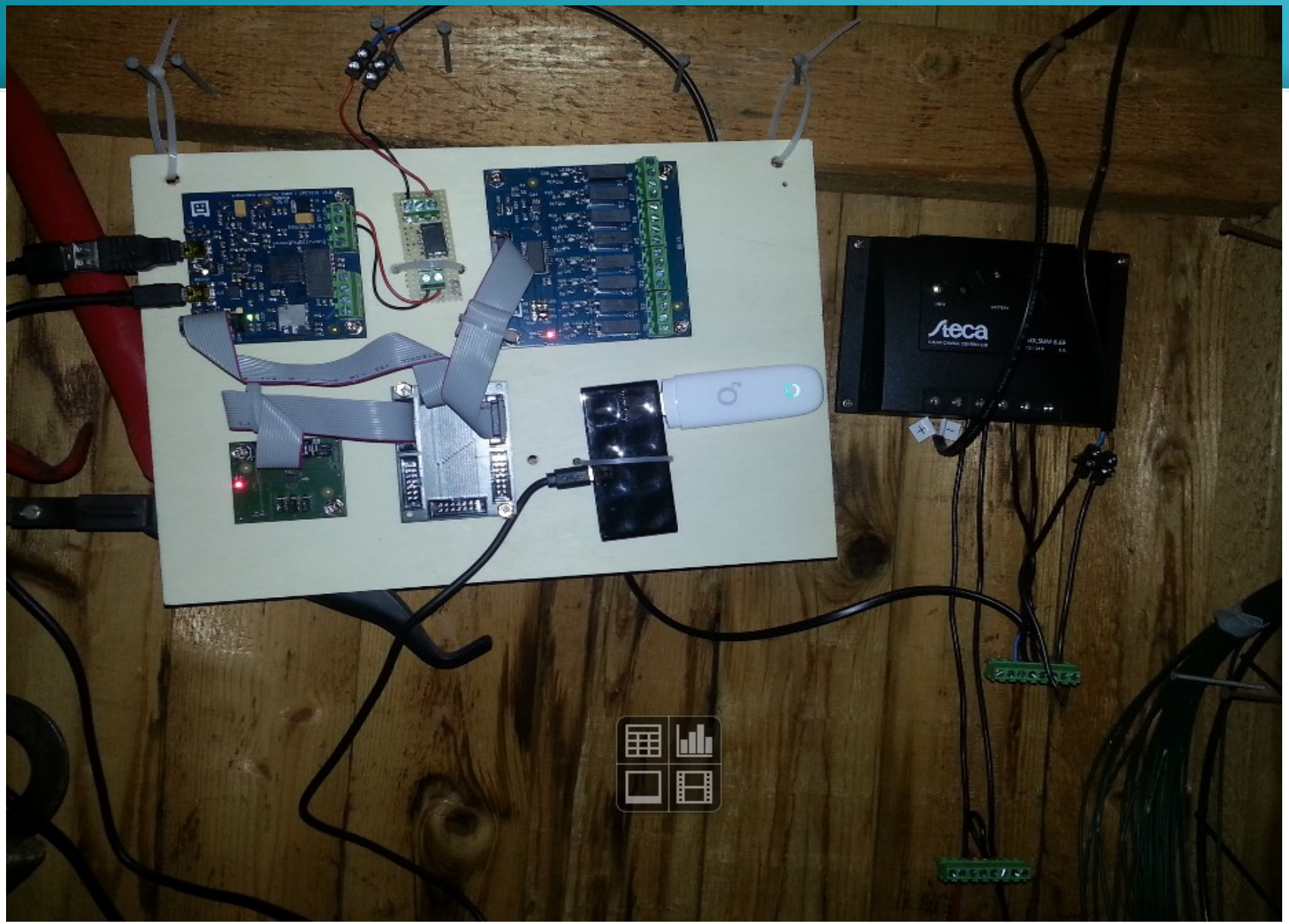
Anwendungen

Radio

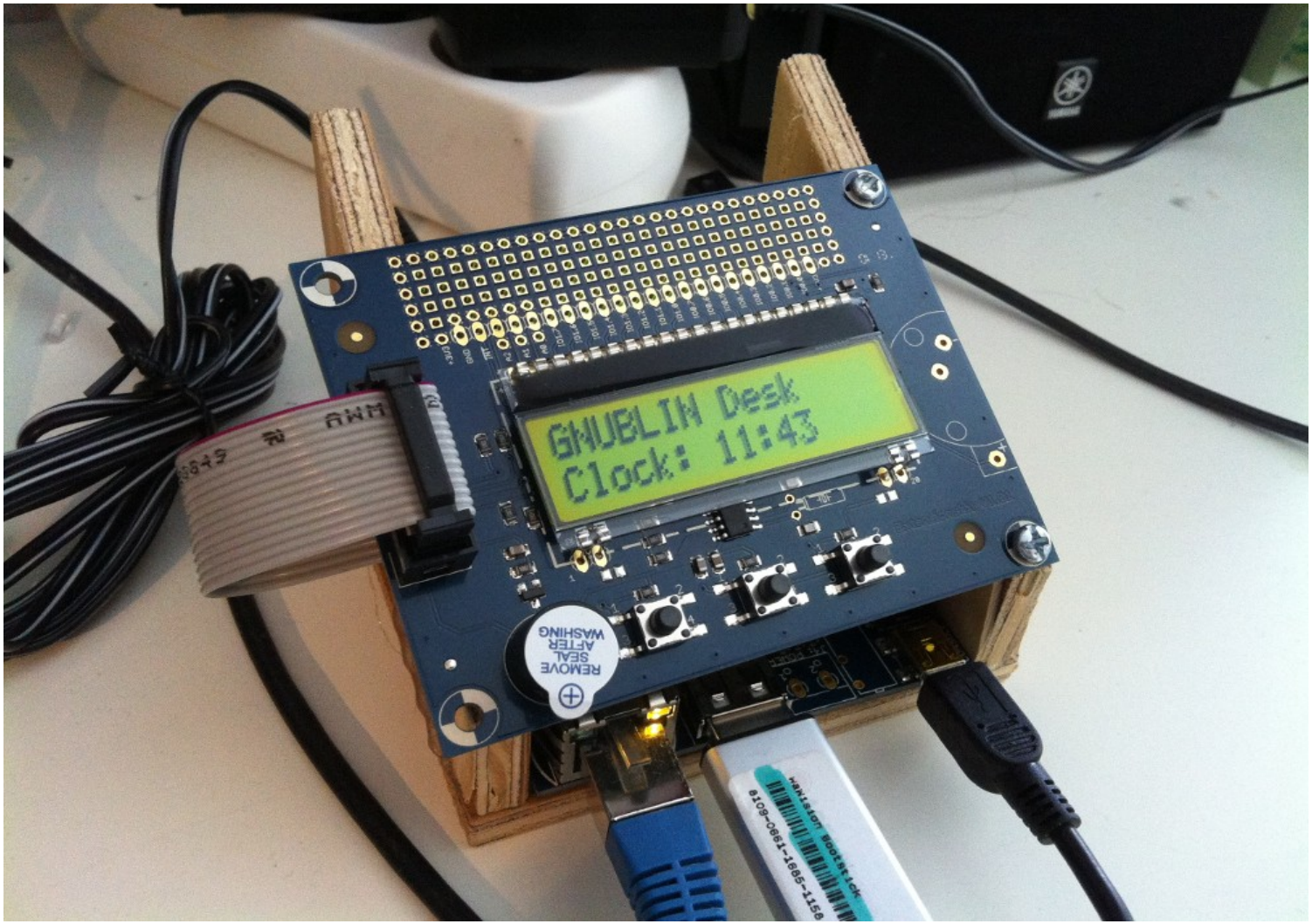


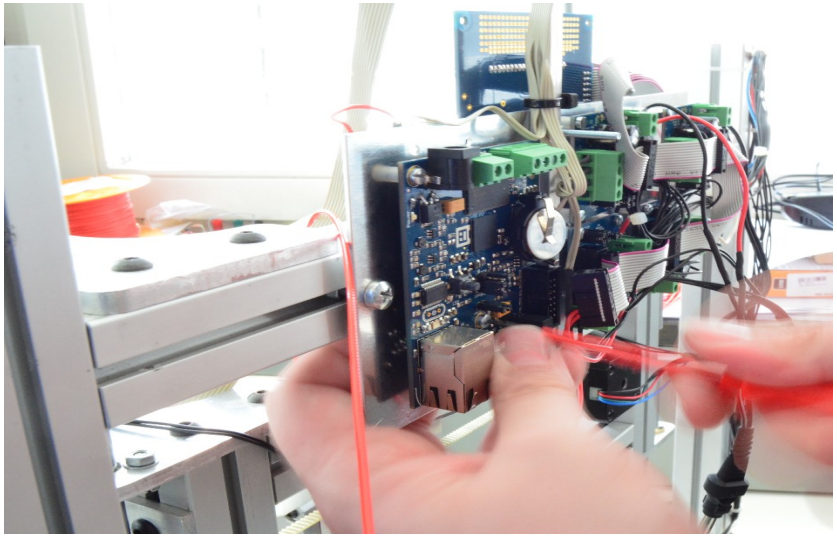
- **Elektor Linux Board LAN bzw. GNUBLIN LAN**
- **Display 4x20, USB Sound, USB Hub, Verstärker**

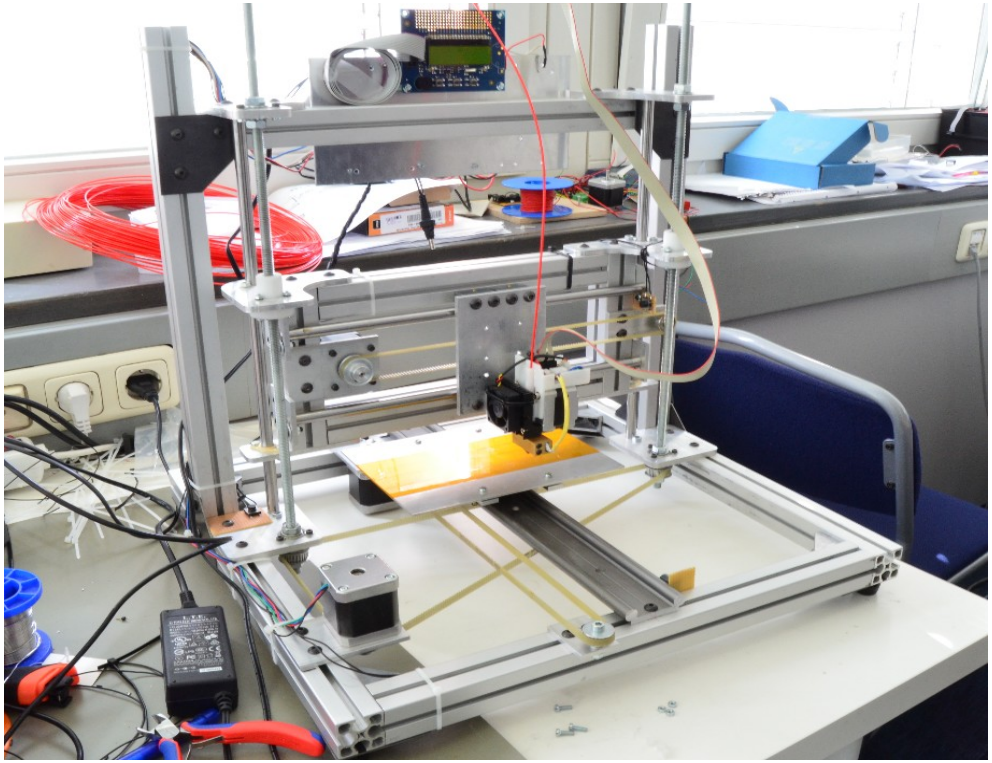
Kleingarten


steca

- UMTS Stick
- Relais-Board für Pumpe
- Temperatursensor
- Solarpanel / Bleiakku
- Webcam?
- ca. 100 mA / 24h? / 365 Tage?
- embedded projects Journal


GNUBLIN Desk
Clock: 11:43
REMOVE SEAL AFTER WASHING





elektor

GNUBLIN

Entwicklungsumgebung für PC

Anwendungen für Entwicklung:

- Gnublin Installer (für neue SD-Karten)
- Serielles Terminal
- SSH / SCP (Konsole per Netzwerk + Dateiübertragung)
- CrossCompiler (Toolchain)
- Entwicklungsumgebung (Eclipse)

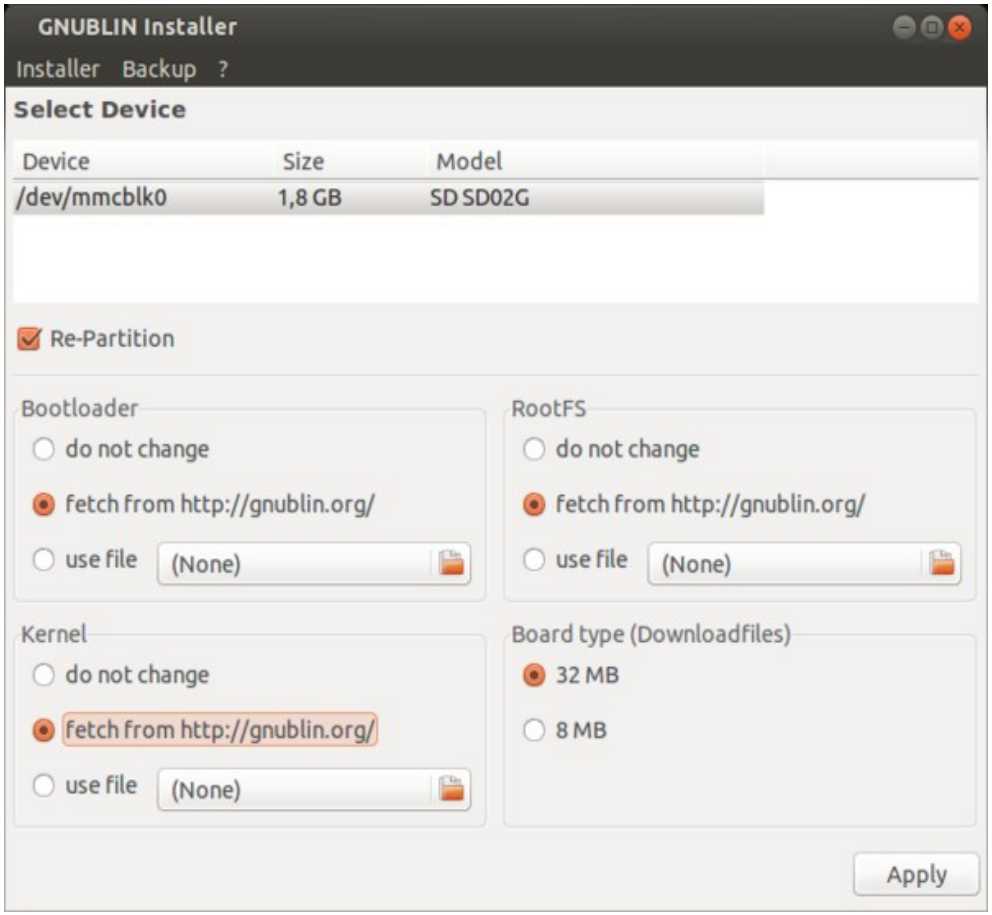
Schritt 1: Live-CD oder USB-Stick

- Linux Entwicklungsumgebung mit notwendigen Tools
- Boot per CD oder USB-Stick
- Alternativ installation auf Festplatte (Anleitung)

Schritt 1: SD-Karte erstellen mit Gnublin Installer

- Download Images
- Partitionierung
- Formatierung
- Kopieren:

  Bootloader, Kernel, RootFS



Schritt 2: Embedded Linux starten

- Serielle Schnittstelle /dev/ttyUSB0
- Bootmeldungen
- Anmelden als Benutzer „root“
- Passwort vergeben

```
picocom -b 115200 /dev/ttyUSB0
```

Schritt 3: Netzwerk Verbindung einrichten

- LAN (Anleitung) oder WLAN
- SSH Verbindung für Konsole
- SCP oder FTP für Daten und eigene Anwendungen

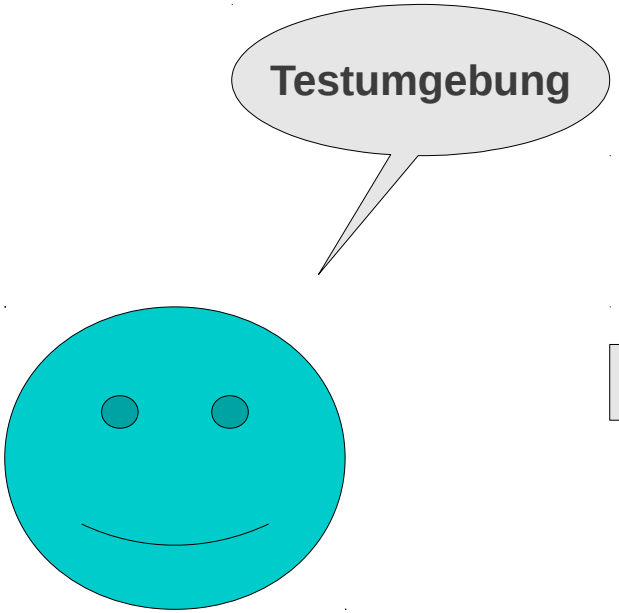
WLAN Verbindung:

```
gnublin-wlan -s „ssid“ -k „schluessel“ -t dhcp
```

Verbindung prüfen:

```
ping 8.8.8.8
```

Fliegenden Prototypen:



Aufgabe:

„Temperatur Regelung“

< 20 ° C → Fenster zu
> 20 ° C → Fenster offen

Display: Zustandsanzeige +
IST Temperatur

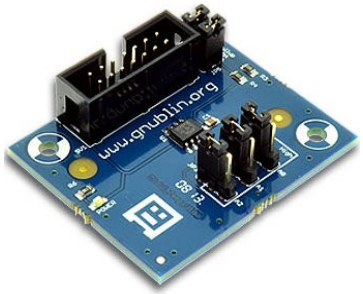
Material:

- Temperatursensor
- Motor
- Display

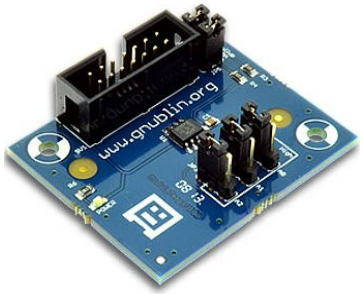
(und ein Linux-Board)

Board

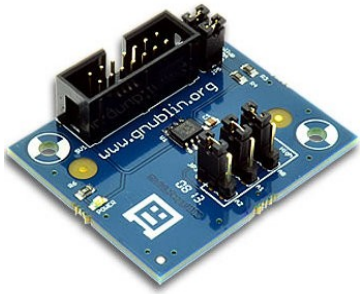
Temperatur Sensor

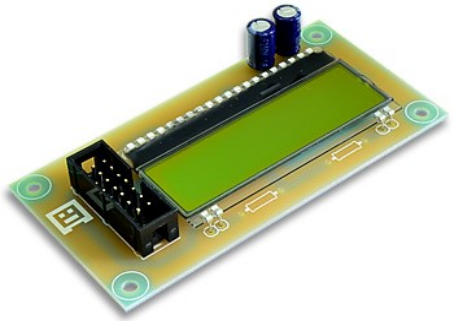


Schrittmotor

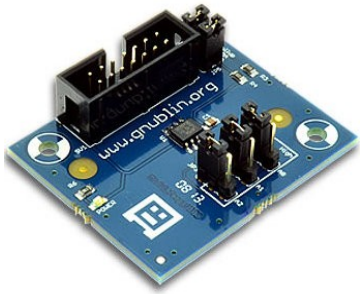


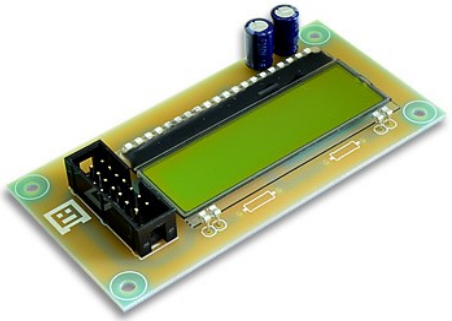
Display





Verkabelung





4 x

Gnublin Tools

Vorteile:

- Per Kommandozeile Funktion testen
- Interaktive Ansteuerung
- Einbau in eigene Software Dank JSON oder Plaintext Ausgabe

Für jedes Module ein Gnublin Tool

- `gnublin-gpio`
- `gnublin-adc`
- `gnublin-temperature`
- `gnublin-relay`
- `gnublin-pwm`
- `gnublin-wlan`
- `gnublin-lm75`
- `gnublin-cam`
- `gnublin-pca9555`
- `gnublin-step`
- `gnublin-dogm`
- `...`

- Option `-h` (Hilfe)
- Option `-b` (Bare entspricht Klartext)
- Option `-j` (JSON)

Gnublin API

Digitaler Ausgang

LED

Intern GPIO Ausgang

```
#include "gnublin.h"

#define BOARD EXTENDED

int main()
{
   gnublin_gpio gpio;

   gpio.pinMode(3,OUTPUT);

   while(1){
     gpio.digitalWrite(3,HIGH);
     sleep(2);
     gpio.digitalWrite(3,LOW);
     sleep(2);
   }
}
```

Digitaler Eingang

Intern GPIO Eingang

```
#include "gnublin.h"

int main()
{
   gnublin_gpio gpio;

   gpio.pinMode(3,INPUT);

   while(1){
     if(gpio.digitalRead(3))
     {
        printf("GPIO is set \n");
     }
     sleep(2);
   }
}
```

Analoger Ausgang

Anschlussklemme

Intern Analog Eingang

```
#include "gnublin.h"

int main()
{
   gnublin_adc ad;


   while(1){
        printf("AD value %i \n",ad.getValue(1));
   }
}
```

Module

Module-Temperature (mehr)

- lm75.setAddress()
- lm75.getTemp()
- Beispiel

Module-LCD_2x16 (Dog Display) (mehr)

- dogm.init()
- dogm.print()
- dogm.controlDisplay
- dogm.returnHome()
- dogm.offset()
- dogm.shift()
- dogm.clear()
- Beispiel

Module-Relay (mehr)

- relay.setAddress()
- relay.switchPin()
- Beispiel

Module-IOExpander (mehr)

- pca.setAddress()
- pca.pinMode()
- pca.digitalRead()
- pca.digitalWrite()
- Beispiel

Module-Step (mehr)

- step.setAddress()
- step.drive()
- step.getActualPosition()
- Beispiel

Module-RTC (mehr)

- rtc.setAddress()
- rtc.getTimestamp()
- ...

Module-ADC (mehr)

- adc.getVoltage()
- adc.getValue()
- Beispiel

Module-LCD-4x20 (mehr)

Schnittstellen

Digital I/O (mehr)

- gpio.pinMode()
- gpio.digitalRead()
- gpio.digitalWrite()
- Beispiel input
- Beispiel output

Analog I/O (mehr)

- adc.getVoltage()
- adc.getValue()
- Beispiel

I2C (mehr)

- i2c.setAddress()
- i2c.receive()
- i2c.send()
- Beispiel

SPI (mehr)

- spi.setCS()
- spi.setSpeed()
- spi.receive()
- spi.send()
- spi.message()
- Beispiel

PWM (mehr)

- pwm.setClock()
- pwm.setValue()
- Beispiel

Serial (mehr)

- serial.send()
- serial.fail()
- serial.setBaudrate()

Bibliotheken

Timer (mehr)

- timer.setTimer()
- time.setEvent()
- time.startTimer()
- time.stopTimer()
- ...

Events (mehr)

- event.registerGPIO()
- event.enableEvents()
- event.disableEvents()

CSV Export (mehr)

- csv.addRow()
- csv.open()
- csv.close()
- csv.delimiterColumn()
- csv.delimiterField()
- csv.delimiterRow()
- csv.NumberToString()
- Beispiel

ini Parser (mehr)

- ini.open()
- ini.setValue()
- ini.getValue()
- ini.close()
- ...

SQL Database (mehr)

- sql.open()
- sql.query()
- sql.close()
- ...

E-Mail (mehr)

- mail.SetSMTPServer()
- mail.SetLogin()
- mail.SetSubject()
- mail.AddRecipient()
- mail.AddMsgLine()

→ gnublin.h / gnublin.cpp

zum Download auf
http://wiki.gnublin.org

Programmieren

Schritt 5: Entwicklungsumgebung starten

- Eclipse für eigenes Projekt
- CrossCompiler am PC nutzen
- Übertragen per Netzwerk

elektor

GNUBLIN

eclipse

Visit other Eclipse Sites

Home Downloads Users Members Committers Resources Projects About Us

Google™ Custom Search Search

Downloads Home

- Juno Packages
- Indigo Packages
- Helios Packages
- Galileo Packages
- Ganymede Packages
- Europa Packages

Eclipse IDE for C/C++ Developers

Package Description

An IDE for C/C++ developers with Mylyn integration.

This package includes:

- **C/C++ Development Tools**
- **Eclipse EGit**
- **Mylyn Task List**
- **Remote System Explorer**

Detailed features list

Maintained by: Eclipse Packaging Project

Download Links

Windows 32-bit
Windows 64-bit
Mac OS X(Cocoa 32)
Mac OS X(Cocoa 64)
Linux 32-bit
Linux 64-bit

Downloaded 540,221 Times

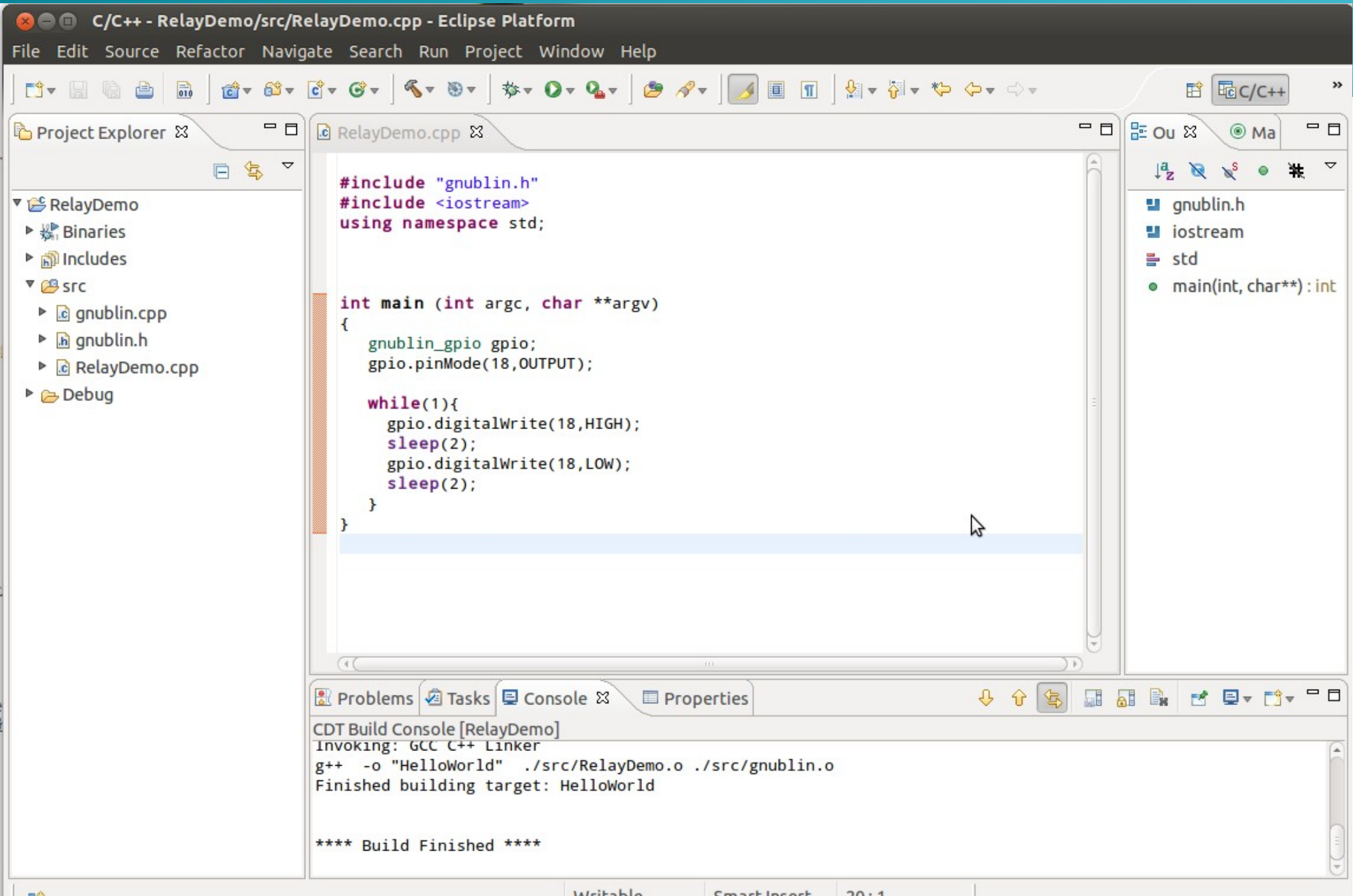
Checksums...

Bugzilla

Open Bugs: 21

Resolved Bugs: 27

File a Bug on this Package


C/C++ - RelayDemo/src/RelayDemo.cpp - Eclipse Platform
File Edit Source Refactor Navigate Search Run Project Window Help
Project Explorer
RelayDemo
Binaries
Includes
src
gnublin.cpp
gnublin.h
RelayDemo.cpp
Debug
RelayDemo.cpp
#include "gnublin.h"
#include <iostream>
using namespace std;
int main (int argc, char **argv)
{
gnublin_gpio gpio;
gpio.pinMode(18,OUTPUT);
while(1){
gpio.digitalWrite(18,HIGH);
sleep(2);
gpio.digitalWrite(18,LOW);
sleep(2);
}
}
gnublin.h
iostream
std
main(int, char**) : int
Problems Tasks Console Properties
CDT Build Console [RelayDemo]
Invoking: GCC C++ Linker
g++ -o "HelloWorld" ./src/RelayDemo.o ./src/gnublin.o
Finished building target: HelloWorld
**** Build Finished ****

Programm übertragen & starten

Schritt 6: Per Netzwerk übertragen

- scp ledblink root@192.168.0.172:/root
- Kein Passwort mehr:
- ssh-keygen
- ssh-copy-id root@192.168.0.172