

GNUBLIN, die offene Embedded- Plattform für die Ausbildung

Benedikt Sauter

Linuxtag Berlin

23.05.2013

Zur Person

- Benedikt Sauter
- Informatiker
- Interessen: Mikrocontroller, Linux, Web
- Diverse Open-Source Projekte

GNU Board Linux

Ziele Projekt GNUBLIN

- Idee ca. 2007 / 2008
- Einfachen **Embedded Linux Rechner** bauen
- Hochschule Augsburg
- Lernen, verstehen, Projekte bauen
- Besonders geeignet für Anfänger
- Einfaches Arbeiten mit dem Board
 - Ein USB Kabel reicht
 - Software auf SD-Karte
 - Einfache IO-Schnittstellen

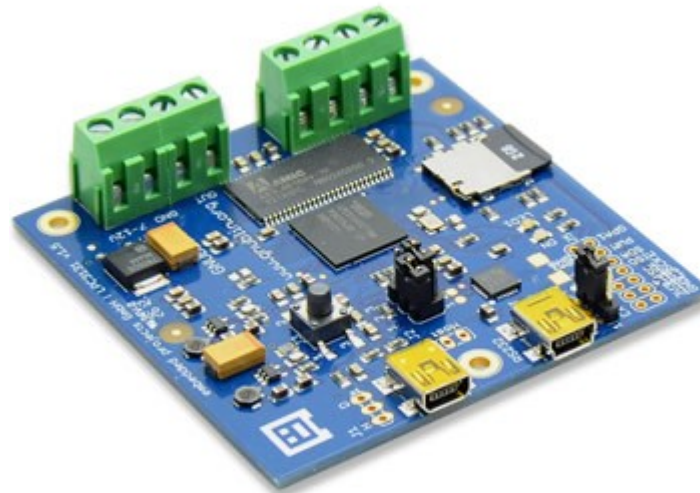


**Hochschule
Augsburg** University of
Applied Sciences

Wunsch

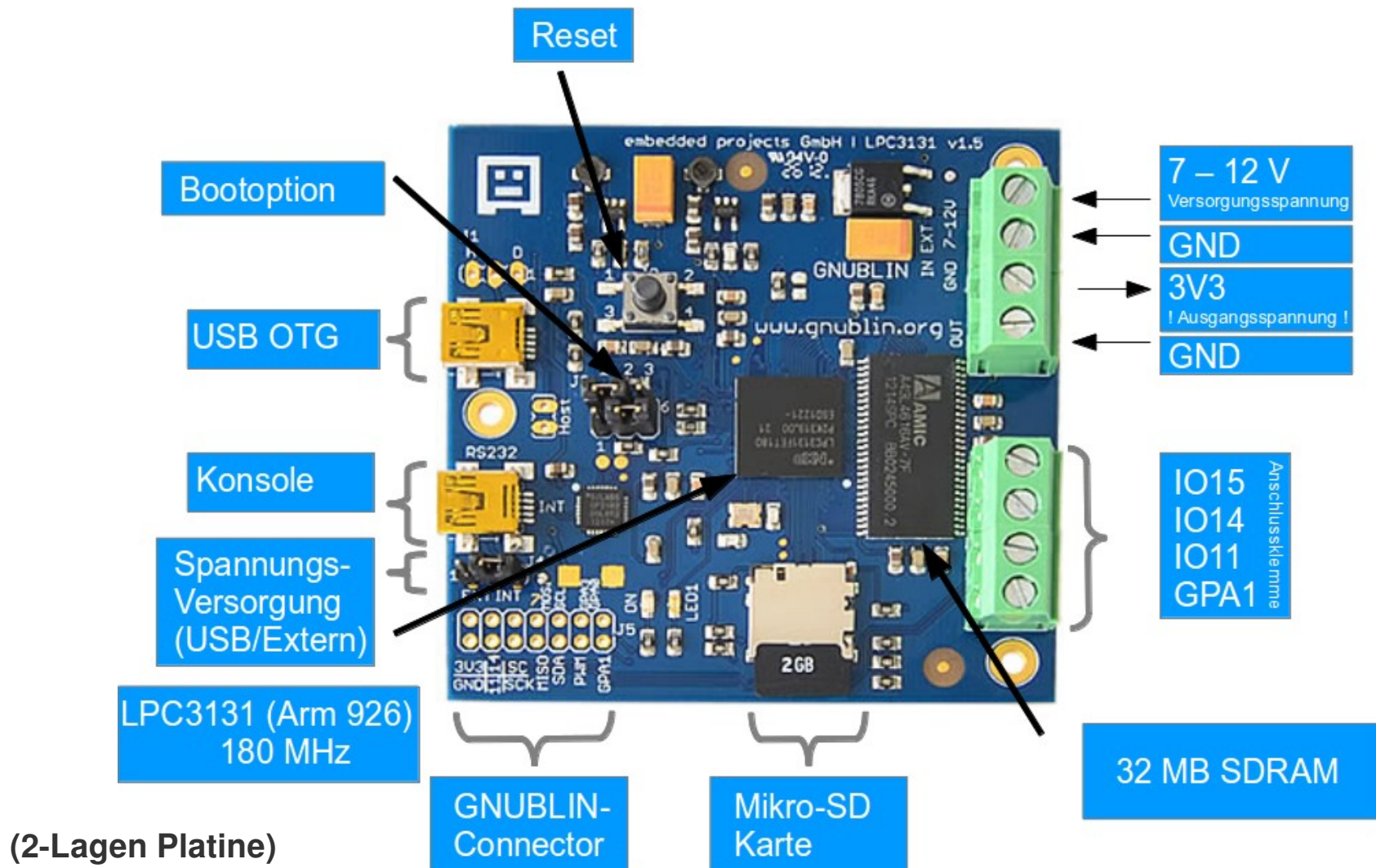
- Komplexität kann später entdeckt werden
- Viele Beispielprogramme
- Alles Open-Source (Board, Software, Tools, Unterlagen, Skripte)
- Messen, Steuern, Regeln

Das Gnublin Board



7 x 7 cm / 2 Lagen

Das Gnublin Board

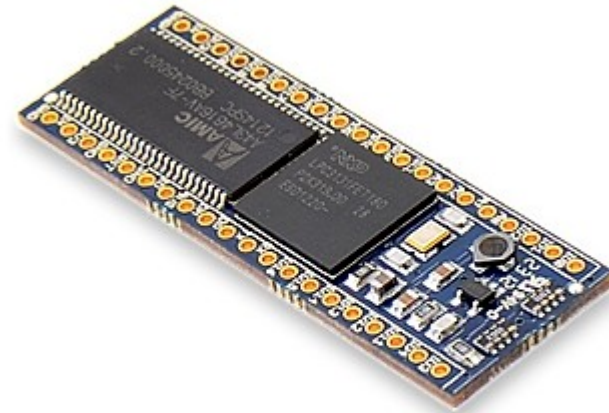


(2-Lagen Platine)

LAN & DIP



Mit Netzwerk, RTC, Relais



Für 40 polige IC-Sockel

Zusammenfassung

- LPC3131 / 180 MHz / 32 MB RAM
- 18 x GPIO
- 4 x AD-Wandler
- I2C / SPI / UART
- PWM
- USB OTG (Host oder Device)
- → **nur 0,5 Watt!!**

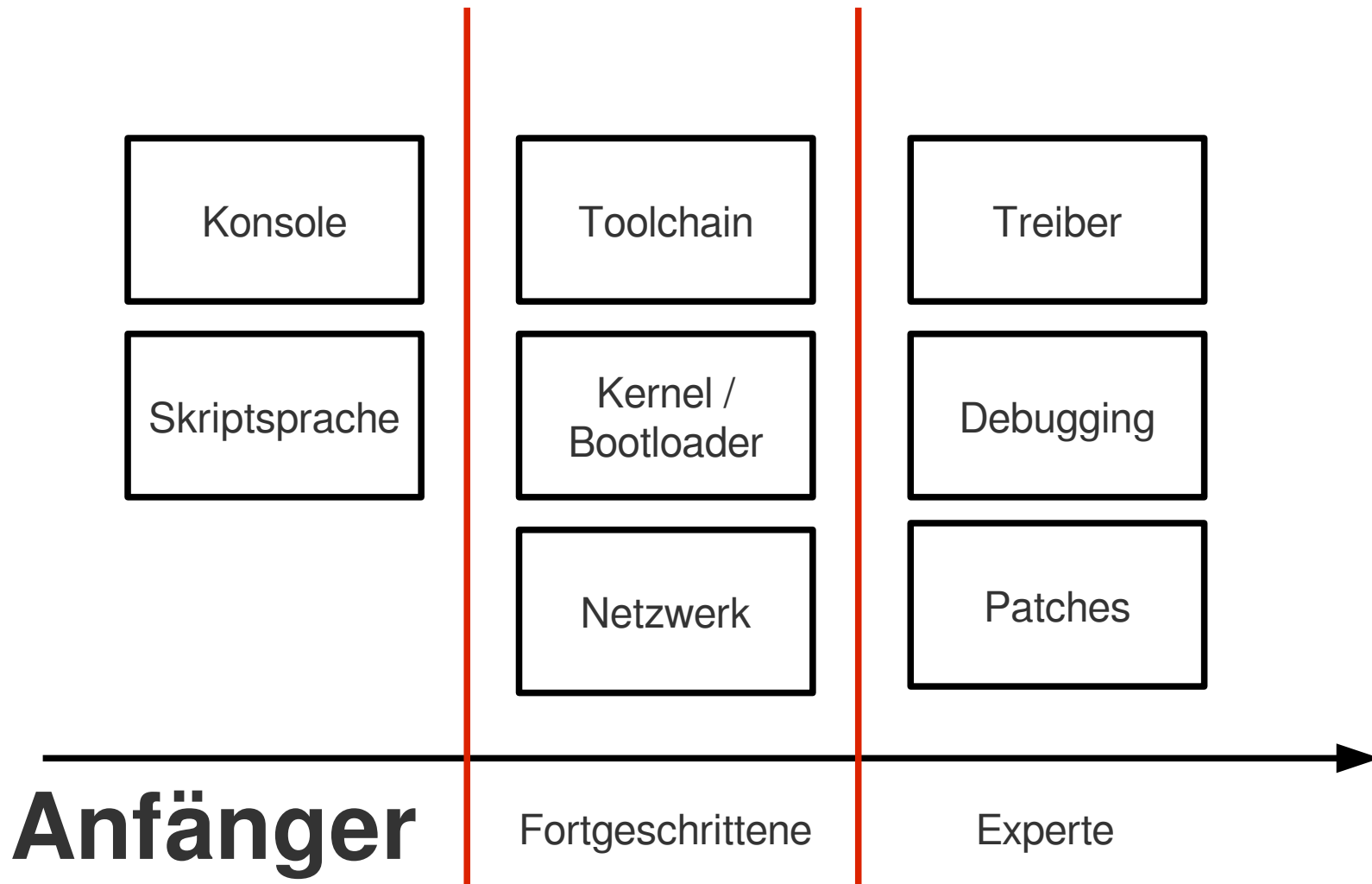
Was kann man damit machen?

- Linux kennenlernen
- Embedded Systeme studieren
- Interpreter Sprachen verwenden
- Einfache Steuerungen bauen
- Datenlogger
- Fahrradcomputer, Kaffeemaschine steuern
- ...

Wie geht man an so ein System ran?

- Vom Anfänger zum Guru werden
- Die richtigen Themen zum richtigen Zeitpunkt
- Für Anfänger muss alles funktionieren!
- Der Anfänger ist kein Linux Guru!

Die Grundregel: STOP!



Projekt Homepage

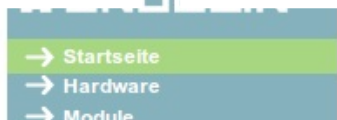
<http://www.gnublin.org>

Wiki, Forum, Tutorials

Wiki

Forum

Tutorials



Viele Erweiterungen

Mehr Informationen

Downloads

Projekte

Ob man mit dem Board Embedded Linux auf einem ARM Mikrocontroller kennenlernen will, ob man Hardware- Erweiterungen, wie z.B. Sensoren, oder Funkmodule anschließen möchte, oder ob man das Board für die Gebäudeautomatisierung für Fahrrad-Computer oder vieles mehr nutzen möchte, das Gemeinschaftsprojekt von embedded projects und der Hochschule Augsburg spricht Anfänger, Fortgeschrittene und Gurus an. Alle Daten, Pläne und Programme gibt es hier frei als Open Source.



Neu: Video Demo-Tour

Das Projekt

Grazer Linuxtage 2013 / Benedikt Sauter, sauter@embedded-projects.net

Das Projekt GNUBLIN entstand gemeinsam mit der Hochschule

Kundenspezifisches GNUBLIN

Die einfachste Möglichkeit an ein eigenes embedded GNU/Linux



Schritt 1: Wiki!

The screenshot shows the top navigation bar of the embedded projects GmbH website. The header is teal with the company logo and name. Below it is a black navigation bar with links to News, Online-Shop, Uni-Shop, Produkte, Journal, Tools, Schulungen, and Unternehmen. A green sidebar contains the GNUBLIN logo and a list of links including Startseite, Hardware, Module, GNUBLIN Designer, Tutorials, Community Wiki / Forum, and zum Online-Shop. A central green area lists various wiki-related links like Wiki Hauptseite, Aktuelle Ereignisse, Datei Hochladen, etc. On the right, there are buttons for Anmelden and Registrieren, and a search box labeled WIKI SUCHE.

[Seite](#) [Diskussion](#) [Quelltext anzeigen](#) [Versionsgeschichte](#)

Gnublin Wiki

Dies ist das Community Wiki für Gnublin. Es dient für das Sammeln von Beschreibungen, Projekten, Links etc. Jeder kann sich an diesem Wiki beteiligen.

Grundlagen

Erste Schritte

Hardware

Messen, Steuern & Regel

Software

Betriebssystem

http://wiki.gnublin.org

GNUBLIN Boards

- GNUBLIN-Standard
- GNUBLIN-LAN
- GNUBLIN-Extended
- GNUBLIN-Elektor

Hardware Wunschliste

On-Board Schnittstellen

- GPIO

Anwendungsentwicklung

- Einfache Entwicklungsumgebung
- C/C++ Toolchain installieren
- Code::Blocks / Eclipse / Anjuta C/C++
- Automatisches Einloggen bei Gnublin
- Anwendung automatisch starten
- Beschreibungen GPIO

Schritt 1: Forum

embedded projects GmbH
HARDWARE FOR PROJECTS

→ News → Online-Shop → Uni-Shop → Produkte → Journal → Tools → Schulungen → Unternehmen

GNUBLIN

- Startseite
- Hardware
- Module
- GNUBLIN Designer
- Tutorials
- **Community Wiki / Forum**
- zum Online-Shop

- Für Anfänger
- Für Fortgeschrittene
- Für Gurus
- Support

- Anmelden/Abmelden
- Registrieren

→ Anmelden
→ Registrieren

FORUM SUCHE

Suche

FAQ Suche

Unbeantwortete Themen | Aktive Themen

Foren-Übersicht Alle Zeiten sind UTC + 1 Stunde [Sommerzeit]

Forum	Themen	Beiträge	Letzter Beitrag
GNUBLIN			
News Moderatoren: J.T., Edlanta, benbrenson, sauterbe	4	20	Di 12. Mär 2013, 09:39 Frank →

<http://forum.gnublin.org>

Moderatoren: J.T., KubuntuFan, Edlanta, benbrenson, sauterbe	77	1062	alexander_ro →
GNUBLIN Community			
Wiki Moderatoren: J.T., KubuntuFan, Edlanta, benbrenson, sauterbe	15	87	Do 28. Mär 2013, 08:53 Kirsch →
Forum Moderatoren: J.T., KubuntuFan, Edlanta, benbrenson, sauterbe	5	10	Mi 10. Apr 2013, 06:04 sauterbe →

Die Mission

3D-Drucker

Fernsteuerung XYZ

Kleingarten

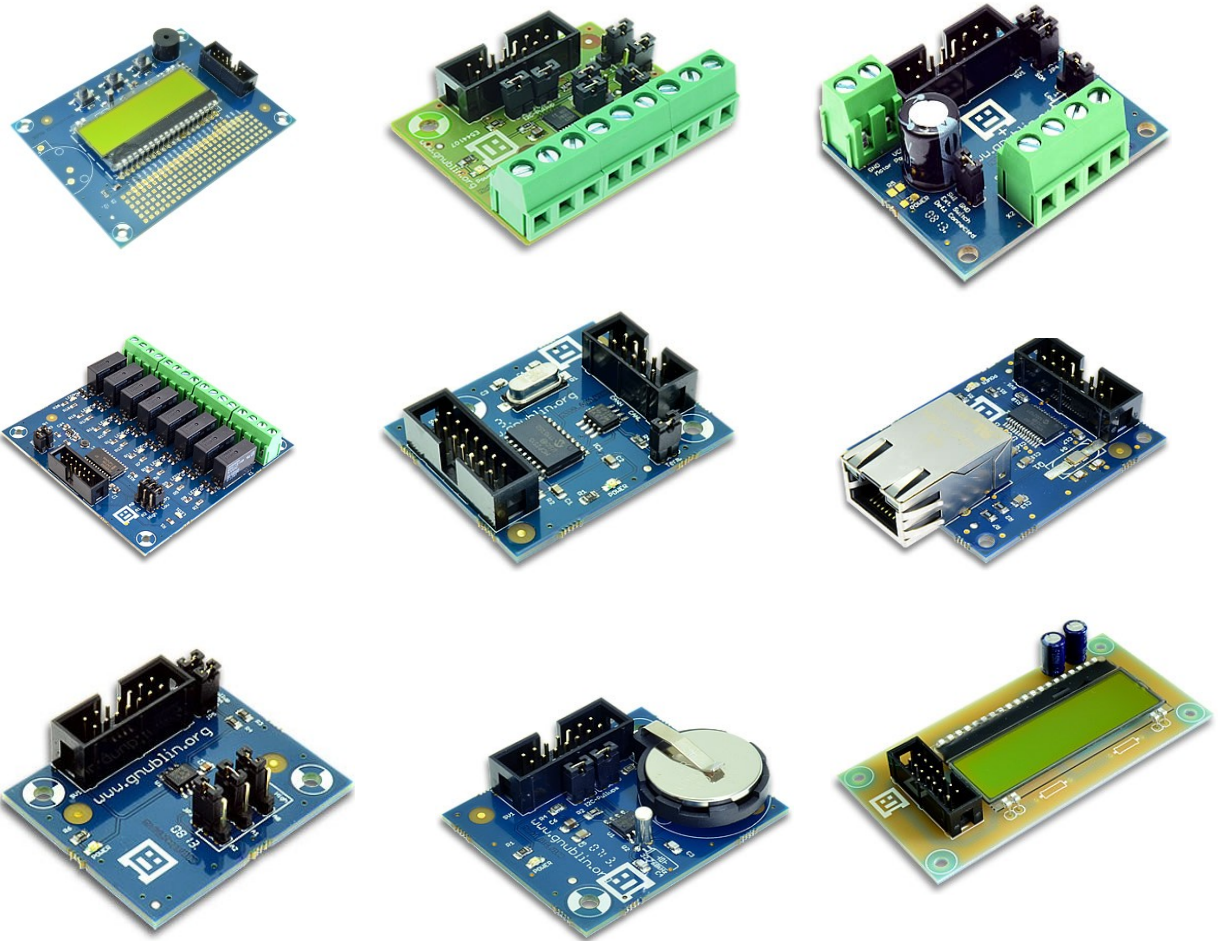
Wetterstation

Haustierüberwachung

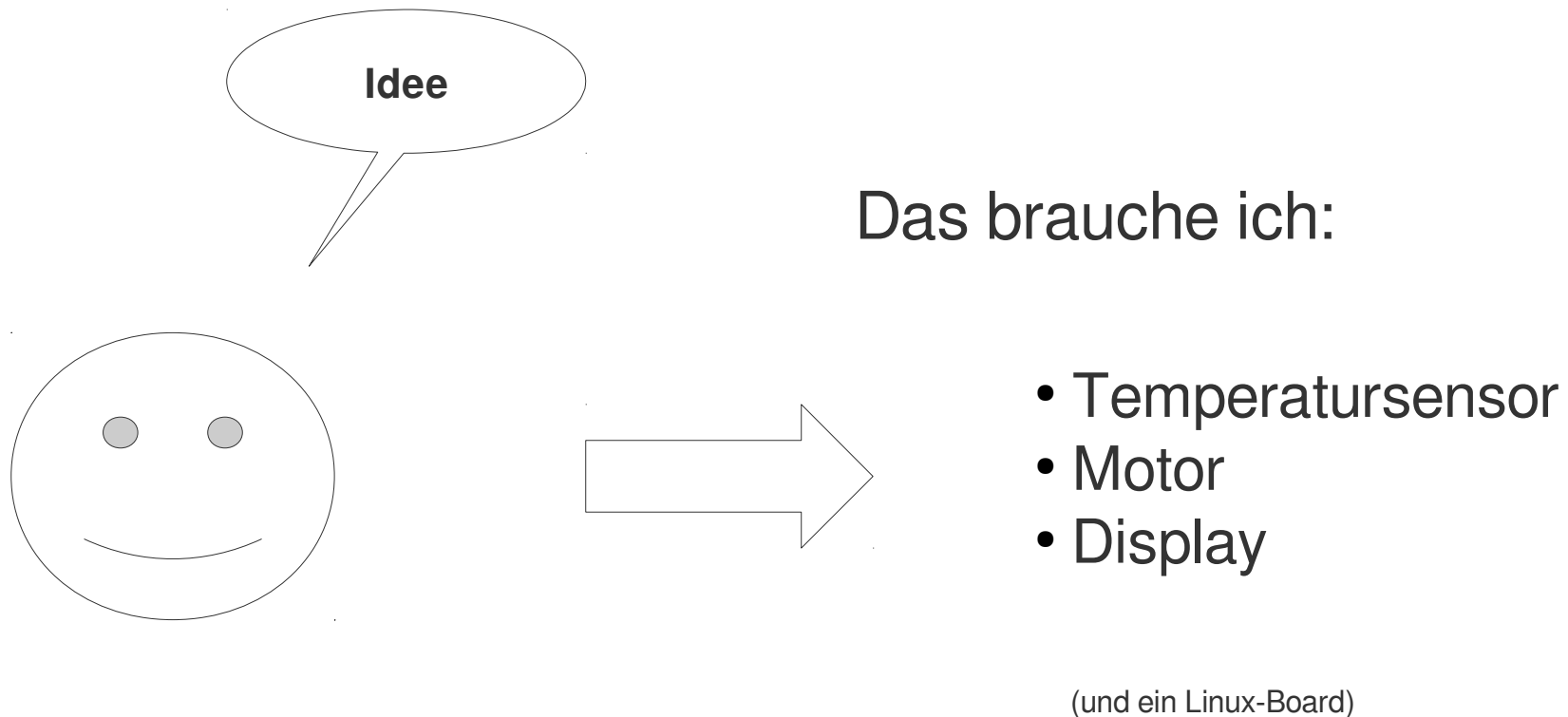
Hausautomatisierung

Die GNUBLIN Module

- Temperatur-Sensor
- Relaiskarte
- GPIO Board
- Schrittmotor
- RTC
- LCD
- CAN
- LAN
- AD-Wandler



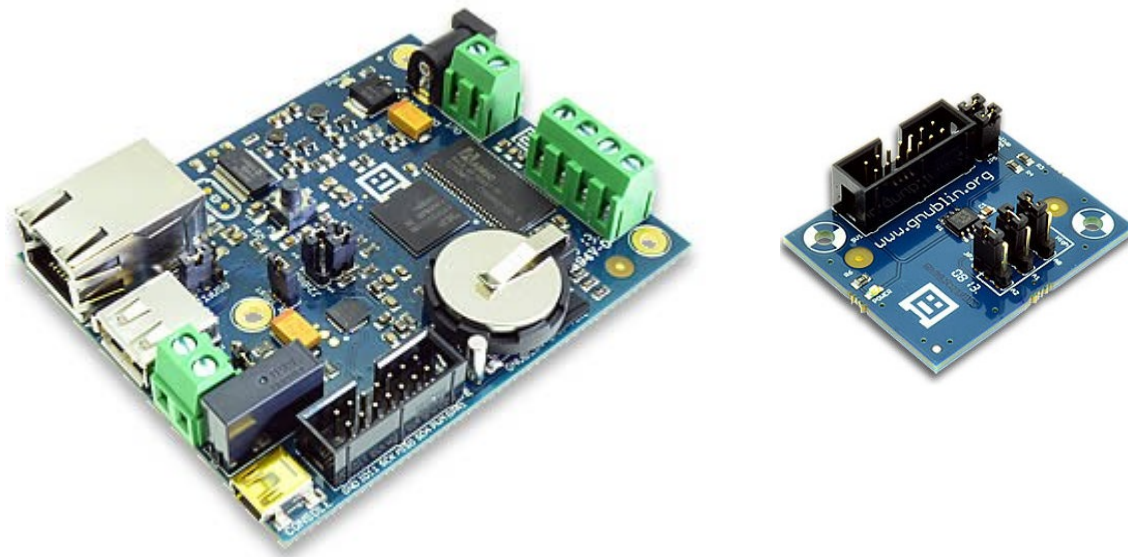
Das muss total einfach gehen!



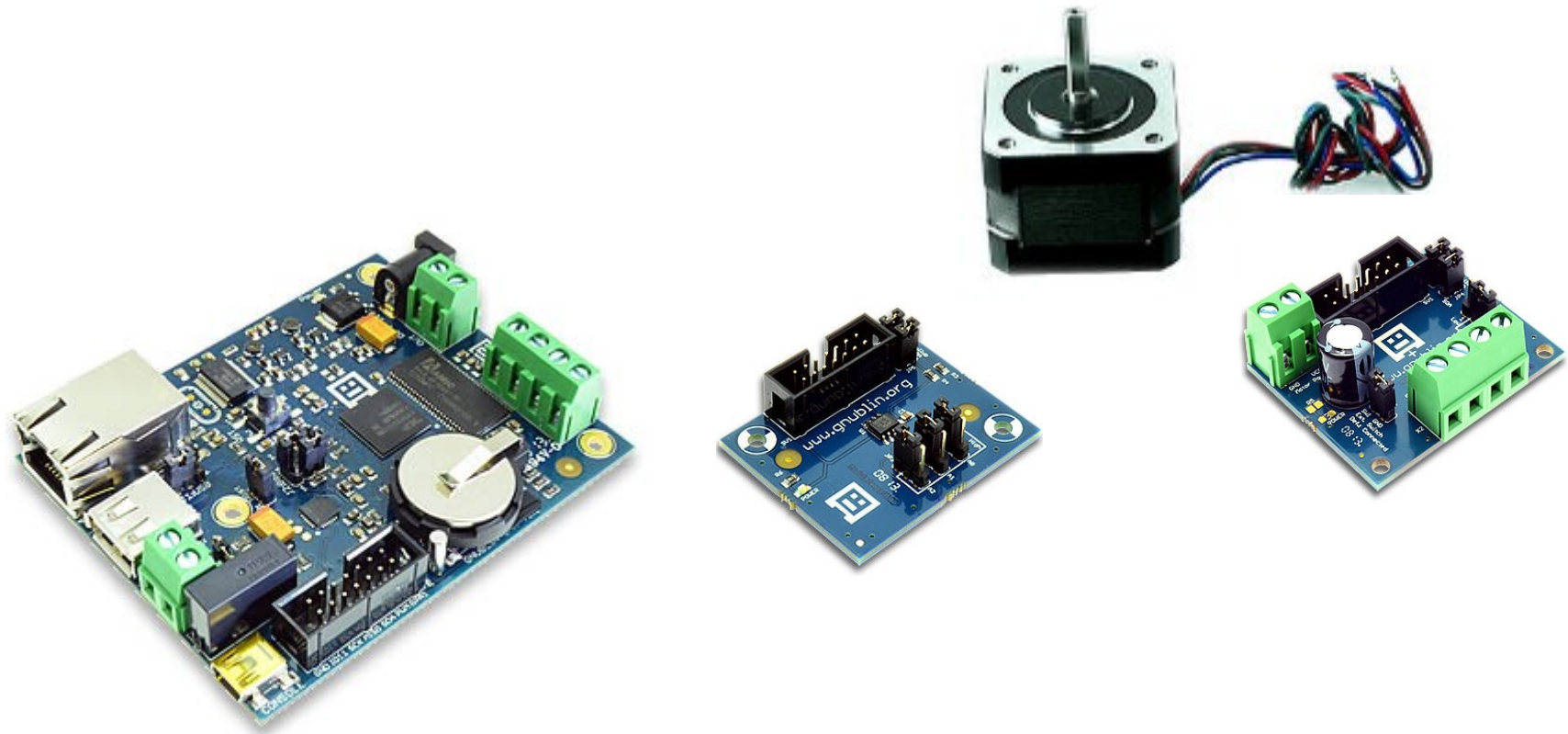
Das Herz: GNUBLIN-LAN



Module-Temperature



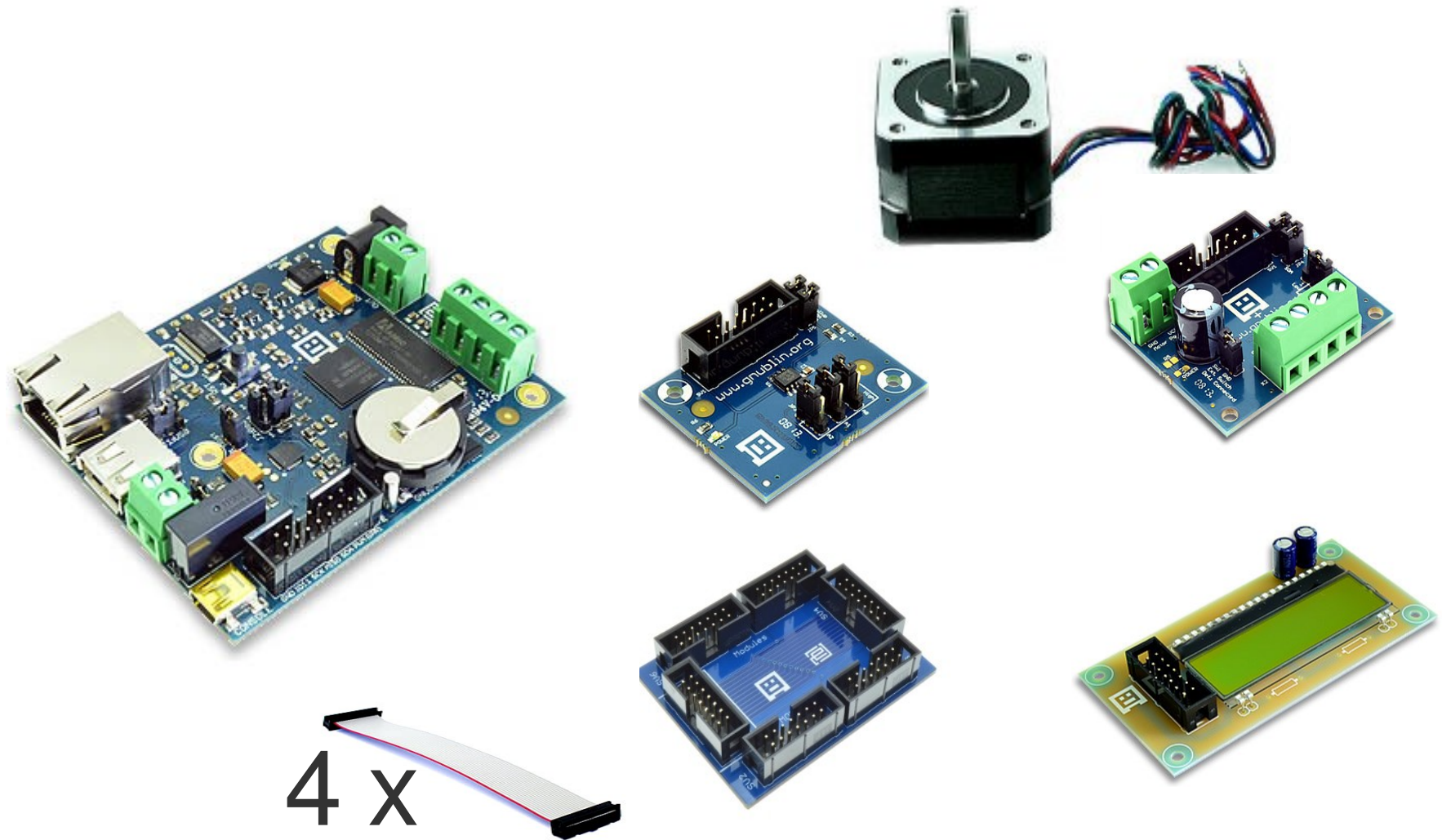
Module-Step



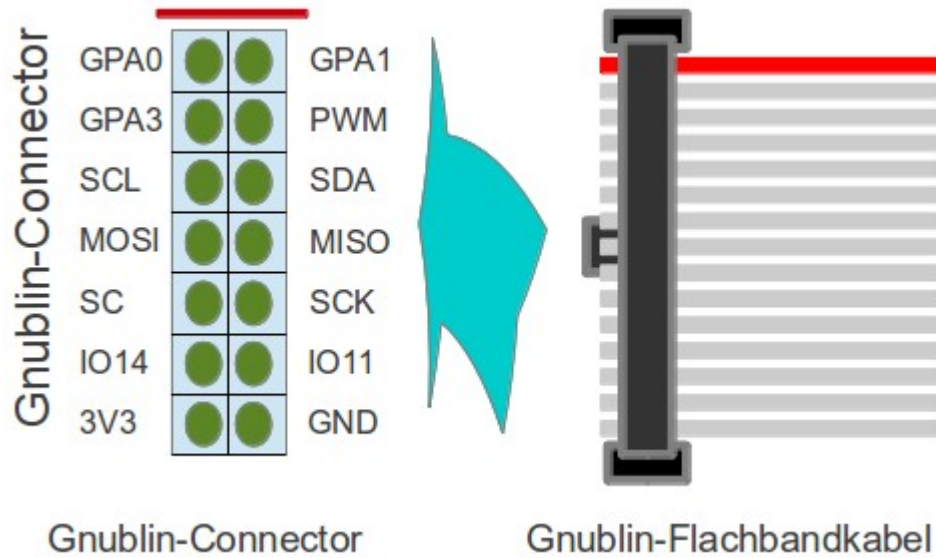
Module-Display 2x16



Module-Bridge



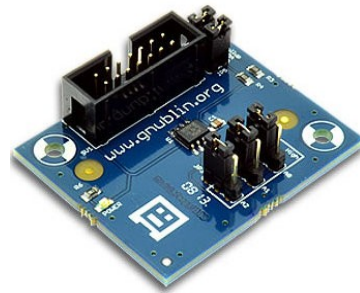
Wie geht das?



Tools

- `gnublin-gpio`
 - `gnublin-adc`
 - `gnublin-temperature`
 - `gnublin-relay`
 - `gnublin-pwm`
 - `gnublin-wlan`
 - `gnublin-lm75`
 - `gnublin-cam`
 - `gnublin-pca9555`
 - `gnublin-step`
 - `gnublin-dogm`
 - `gnublin-adcmod`
 - ...
- Option `-h` (Hilfe)
 - Option `-b` (Bare entspricht Klartext)
 - Option `-j` (JSON)

Demo



```
root@gnublin:~# gnublin-lm75  
26.125 C
```

Open-Source

- Alle Daten verfügbar
- Stücklisten
- Schaltplan
- Eagle Dateien
- ...



Fortgeschritten

Die API

Digitaler Ausgang



LED

Intern GPIO Ausgang

```
#include "gnublin.h"

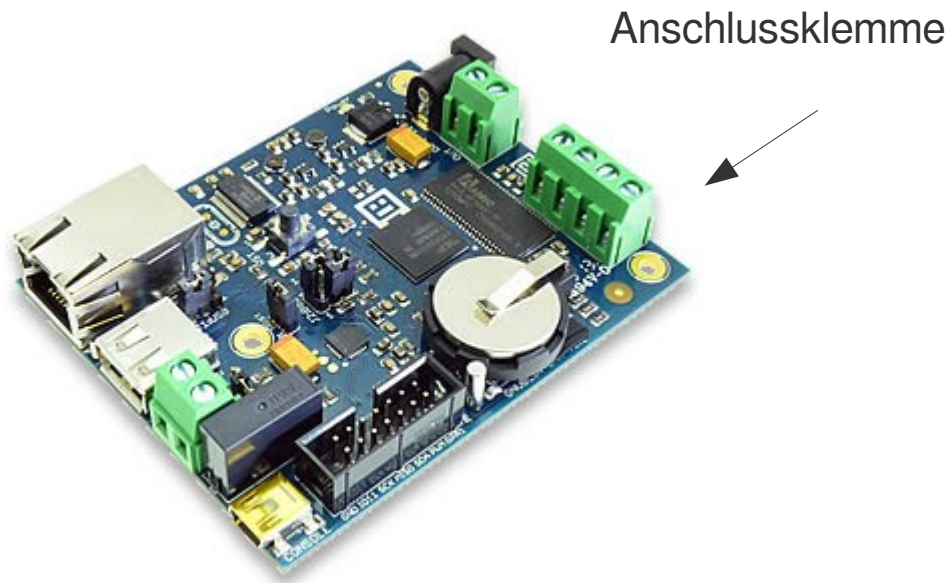
#define BOARD EXTENDED

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

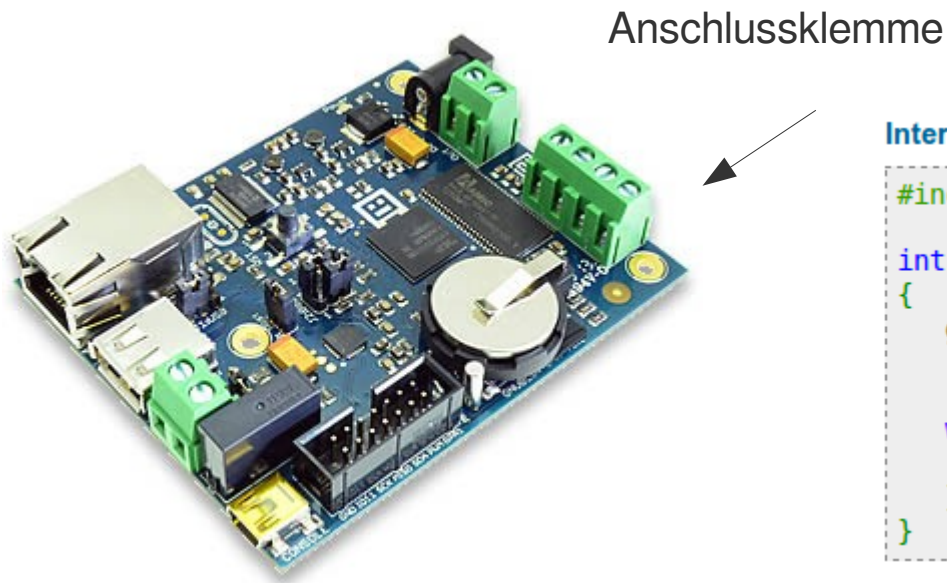
```
#include "gnublin.h"

int main()
{
    gnublin_gpio gpio;

    gpio.pinMode(3,INPUT);

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

Analoger Eingang



Intern Analog Eingang

```
#include "gnublin.h"

int main()
{
    gnublin_adc ad;

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


gnublin.h / gnublin.cpp

→ gnublin.h / gnublin.cpp

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

Anleitung CodeBlocks

The screenshot shows the Code::Blocks IDE interface. The main editor window displays a C++ program with the following code:

```
5  
6 int main()  
7 {  
8     gnuBLIN_gpio gpio;  
9  
10    gpio.pinMode()  
11 }  
12
```

A tooltip for the `pinMode()` call shows the function signature: `int gnuBLIN_gpio::pinMode(int pin, std::string direction)`.

The Project Manager on the left shows a project named "Workspace" with a list of libraries:

- gnuBLIN_adc
- gnuBLIN_gpio
- gnuBLIN_i2c
- gnuBLIN_module_adc
- gnuBLIN_module_dogm
- gnuBLIN_module_lm75
- gnuBLIN_module_pca9555
- gnuBLIN_module_relay

The "Compiler and debugger" menu is open, showing options like "Environment...", "Editor...", "Compiler and debugger...", "Global variables...", "Scripting...", and "Edit startup script".

The "Code::Blocks" dialog box is open, showing the "Recent projects" section with a project named `/tmp/HalloGnuBLIN/HalloGnuBLIN.cbj` and the "Recent files" section which is empty.

The "Logs & others" panel at the bottom shows the "Build log" and "Build messages" tabs.

The status bar at the bottom left says "Welcome to Code::Blocks!" and the bottom right says "default".

Anleitung Makefile

```
user@dev-pc:~$ wget http://www.gnublin.org/downloads/ledblink.cpp
```

Inhalt der Datei *ledblink.cpp*:

```
#include "gnublin.h"

int main(){
  gnublin_gpio gpio;
  gpio.pinMode(3, OUTPUT); //GPIO3 als Ausgang festlegen

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

```
user@dev-pc:~$ wget http://www.gnublin.org/downloads/Makefile
```

Inhalt der Datei *Makefile*:

```
objects = ledblink

all: gnublin.o $(objects)

gnublin.o: gnublin.cpp gnublin.h
           arm-linux-gnueabi-g++ -c gnublin.cpp

$(objects): $(objects).cpp gnublin.o
            arm-linux-gnueabi-g++ -o $(objects) $(objects).cpp gnublin.o

clean:
       rm -Rf *.o $(objects)
```

Möchte man nun das Programm übersetzen muss man einfach in dem Ordner in dem die Dateien und das Makefile liegen in der

```
user@dev-pc:~$ make
```

API Referenz

Referenz

Module

Module-Temperature [\(mehr\)](#)

- `lm75.setAddress()`
- `lm75.getTemp()`
- ...

Module-LCD_2x16 (Dog Display) [\(mehr\)](#)

- `dogm.print()`
- `dogm.controlDisplay`
- `dogm.returnHome()`
- `dogm.offset()`
- `dogm.shift()`
- `dogm.clear()`

Schnittstellen

Digital I/O [\(mehr\)](#)

- `gpio.pinMode()`
- `gpio.digitalRead()`
- `gpio.digitalWrite()`
- ...

Analog I/O [\(mehr\)](#)

- `adc.getVoltage()`
- `adc.getValue()`
- ...

Bibliotheken

Timer [\(mehr\)](#)

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

Events [\(mehr\)](#)

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

API Referenz Teil 2

- ...

Module-Relay (mehr)

- `relay.setAddress()`
- `relay.switchPin()`
- ...

Module-IOExpander (mehr)

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

Module-RTC (mehr)

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

Module-ADC (mehr)

- ...

Module-LCD-4x20 (mehr)

- `lcd.setAddress()`
- ...

I2C (mehr)

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

SPI (mehr)

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

PWM (mehr)

- `pwm.setClock()`
- `pwm.setValue()`
- ...

Serial (mehr)

- `serial.begin()`
- `serial.write()`

CSV Export (mehr)

- `csv.open()`
- `csv.addRow()`
- `csv.close()`
- ...

ini Parser (mehr)

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

SQL Database (mehr)

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

E-Mail (mehr)

- `mail.setSMTPHost()`
- `mail.setSMTPPort()`
- `mail.setSMTPUser()`
- `mail.setSMTPPassword()`



GNUBLIN

- Startseite
- Hardware
- Module
- GNUBLIN Designer
- Tutorials
- **Community Wiki / Forum**
- zum Online-Shop 🛒

- Wiki Hauptseite
- Aktuelle Ereignisse
- Datei Hochladen
- Letzte Änderungen
- Zufällige Seite
- Hilfe
- Links auf diese Seite
- Spezialseiten
- PDF-Version
- Quelltext anzeigen
- Versionen/Autoren
- Abmelden

- Anmelden
- Registrieren

WIKI SUCHE

[Seite](#) [Diskussion](#) [Quelltext anzeigen](#) [Versionsgeschichte](#)

Gnublin Wiki

Dies ist das Community Wiki für Gnublin. Es dient für das Sammeln von Beschreibungen, Projekten, Links etc. Jeder kann sich an diesem Wiki beteiligen.

Grundlagen

Erste Schritte

- Einstieg in die Gnublin Welt
- unter Linux
- unter Windows
- unter Mac OS X
- Erste Versuche mit Gnublin

- **Neu: C++ API für Gnublin**
- **Neu: API Beispiele**

GNUBLIN Boards

- GNUBLIN-Standard
- GNUBLIN-LAN
- GNUBLIN-Extended

Hardware

Messen, Steuern & Regel

- Temperatursensor
- Relais-Karte
- Schrittmotor
- LCD Display
- Uhrzeit
- AD-Wandler Karte
- Portexpander
- GPS Adapter
- Erweiterungskarte
- ...

- [Hardware Wunschliste](#)

On-Board Schnittstellen

Software

Betriebssystem

- **Die Gnublin Distribution**
- Der Gnublin Installer
- Das Gnublin Paket System
- Sicherheitsbackup der SD-Karte
- Wechsel auf größere SD-Karte / Manuell SD Karte partitionieren
- Gnublin Emulator am PC
- ...

Anwendungsentwicklung

- Einfache Entwicklungsumgebung
- C/C++ Toolchain installieren
- Code::Blocks / Eclipse / Anjuta C/C++
- Automatisches Einloggen bei Gnublin
- Anwendung automatisch starten

Ja, wie Arduino



nur mit Linux „inside“ ...



GNUBLIN

- Startseite
- Hardware
- Module
- GNUBLIN Designer
- **Tutorials**
- Community Wiki / Forum
- zum Online-Shop

→ zum Shop



TUTORIAL SUCHE

Suchen

Versuche & Anleitungen

GNUBLIN entdecken, lernen und verstehen. Wir möchten nicht nur Hard- und Softwarelieferant sein sondern wollen mit Wissen und Erfahrung helfen eigene Produkte und Versuche umzusetzen.

In unserer neuen Tutorial Sektion zeigen wir verschiedene Ergebnisse und erklären Themen und Fachgebiete mit einfachen praktischen Beispielen.

Tutorials

C++ API für GnuBLIN

Die GNUBLIN API bietet einen einfachen Zugriff auf alle Schnittstellen an.

```
int main()
{
  gnuBLIN
  (a) gnuBLIN_gpio
  (a) gnuBLIN_i2c
  (a) gnuBLIN_module_adc
  (a) gnuBLIN_module_dogs
  (a) gnuBLIN_module_lm75
  (a) gnuBLIN_module_pca9555
  (a) gnuBLIN_module_relay
}
```

[weiter lesen](#)

Webcam mit GnuBLIN

So einfach verbindet man eine Webcam und nutzt den Video-Stream in der eigenen Anwendung.



[weiter lesen](#)

Uhrzeit mit Pufferbatterie

Benötigt man eine Uhrzeit um Messwerte zu archivieren oder in Prozess zu einem festen Zeitpunkt zu starten.



[weiter lesen](#)

2x16 Zeichen Display

Das 2x16 DOG Display kann für einfache Ausgaben und Menüs problemlos dienen.



IO-Expander (16 Ein- & Ausgabe)

Pro IO-Expander hat man zusätzliche 16 weitere Ein- oder Ausgang Pins.



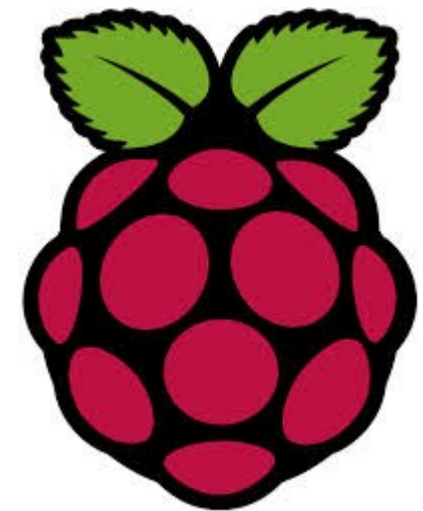
Einfache Menüsteuerung

Mit Hilfe dieser C-Bibliothek erstellt man einfache Menüs.



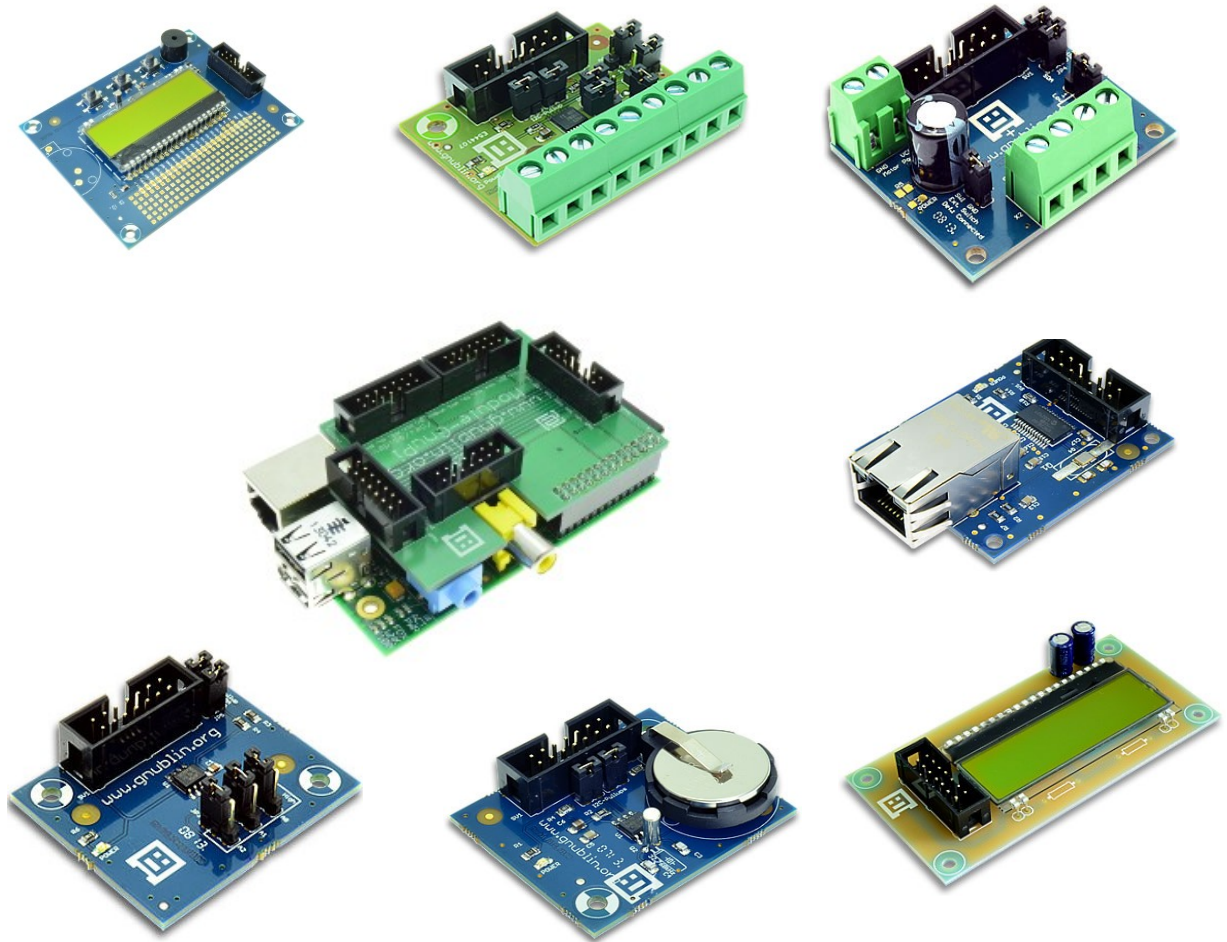
Unterschied RaspberryPi?

- Rechenleistung 700 MHz – ARM11 (Pi)
- Rechenleistung 180 MHz – ARM9 (GNUBLIN)
- Layout / Schaltplan: ????? (Pi)
- Layout / Schaltplan: Creative Commons (GNUBLIN)
- Stromverbrauch: 2,5 – 3,5 W
- Stromverbrauch: 0,25 – 1 W



GNUBLIN meets Pi

Adapter-Platine für
GNUBLIN Module
Mit RaspberryPi nutzen!



Gnublin Distribution

 **GNUBLIN** 



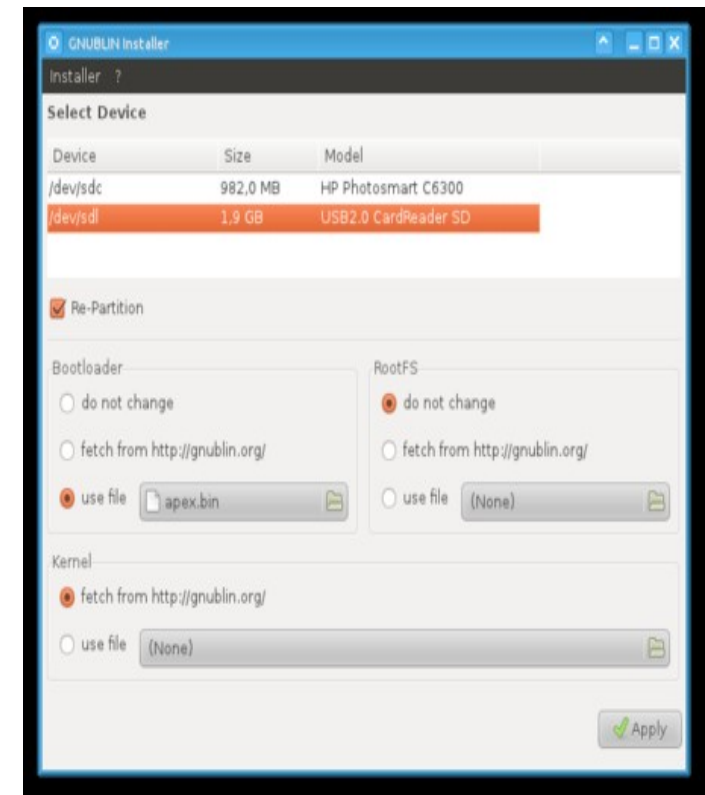
debian

Die Linux-Distribution inkl. Hard- und Software

- Debian 6.0 Grundsystem
- Angepasster Kernel für GNUBLIN
- Bootloader
- Ein Ordner mit vielen Beispielen (Skripte, Programmbeispiele, etc.)
- GNUBLIN Pakete zum Installieren von Zusatzmodulen und Programm Zusammenstellungen

Gnublin Installer

- wxWidget
- C++
- Download / Partitionierung
- SD-Karte beschreiben
- Bootloader / Kernel / RootFS



Projekte

Bewässerung Kleingarten

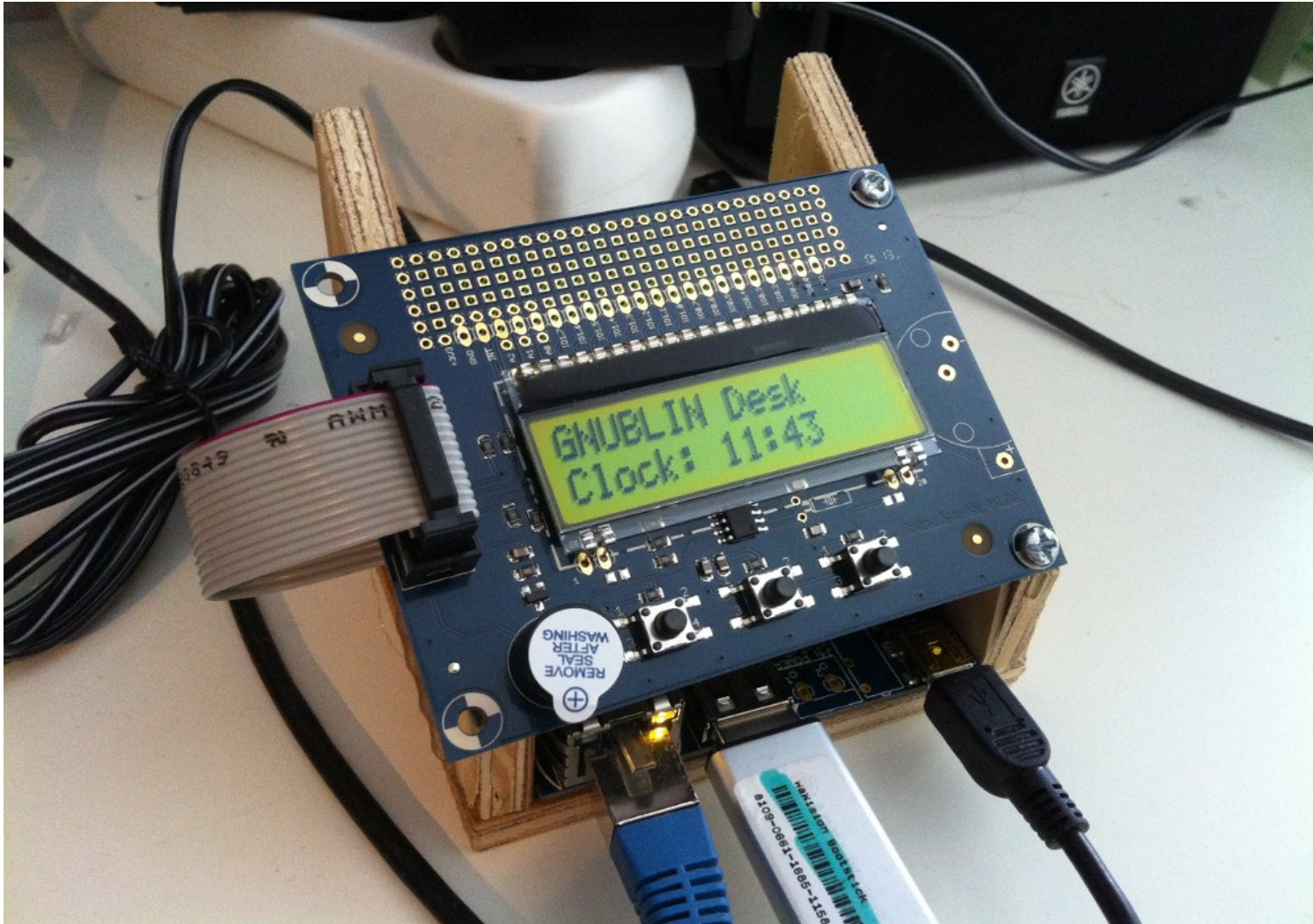




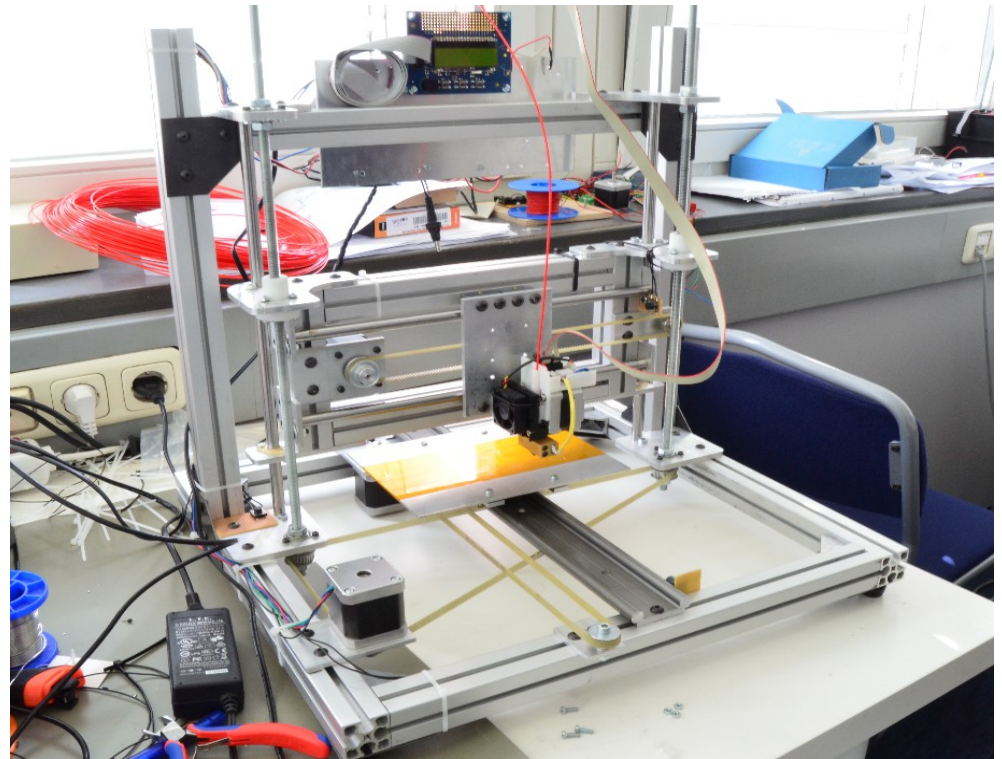
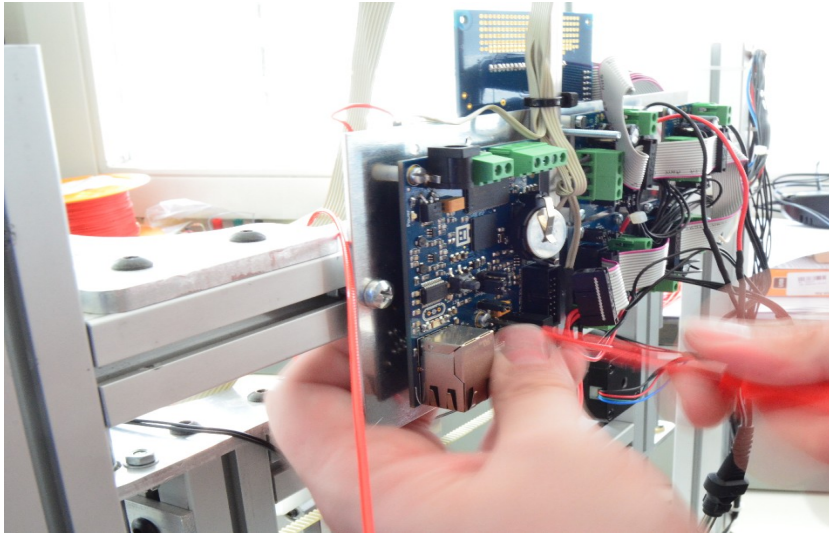
Projekt

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

Mini Pult



3D Drucker



Ende

Fragen?