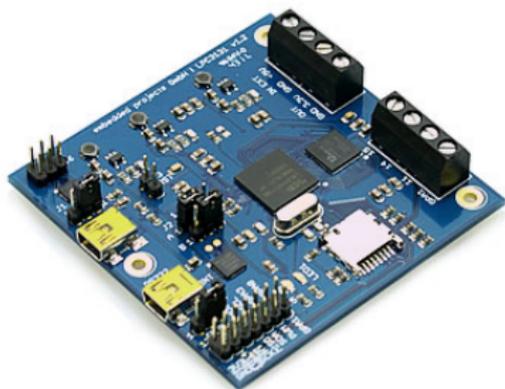


Embedded Linux am Beispiel des GnuBLin-Boards



Was ist Embedded Linux?

Wikipedia

Als Embedded Linux bezeichnet man ein eingebettetes System mit einem auf dem Linux-Kernel basierenden Betriebssystem.

In der Praxis:

- Board mit Prozessor (z.B. ARM, MIPS), RAM und externen Schnittstellen
- oftmals Graphik-Prozessor und Display integriert
- Sensoren, Akku als Spannungsversorgung
- stromsparend, klein, leise, günstig
- Einsatzgebiete:
 - kleiner Server (Router)
 - für Steuer- und Regelaufgaben (Drohnen, Segways)
 - Automatisierung (Rolladen, Beleuchtung, Kaffeemaschine)
 - Unterhaltungselektronik (Media-Player)
 - Handys (Android)

Embedded Linux

Was braucht man für die Hardware?

- Prozessor auf dem Linux läuft
- RAM
- Speicherplatz (z.B. NAND, SD-Karte)
- Schnittstellen / IO-Ports / Display ...

Was braucht man an Software?

- auf dem PC:
 - Cross-Compiler (gcc)
 - Tool zum Übertragen der Software / Firmware
- auf dem Embedded Linux System:
 - Bootloader
 - Linux-Kernel
 - Linux-System (initramfs, root-Dateisystem)

Motivation

- entwickelt von Embedded Projects GmbH in Zusammenarbeit mit der Hochschule Augsburg
- Board mit zwei Lagen

Ziele

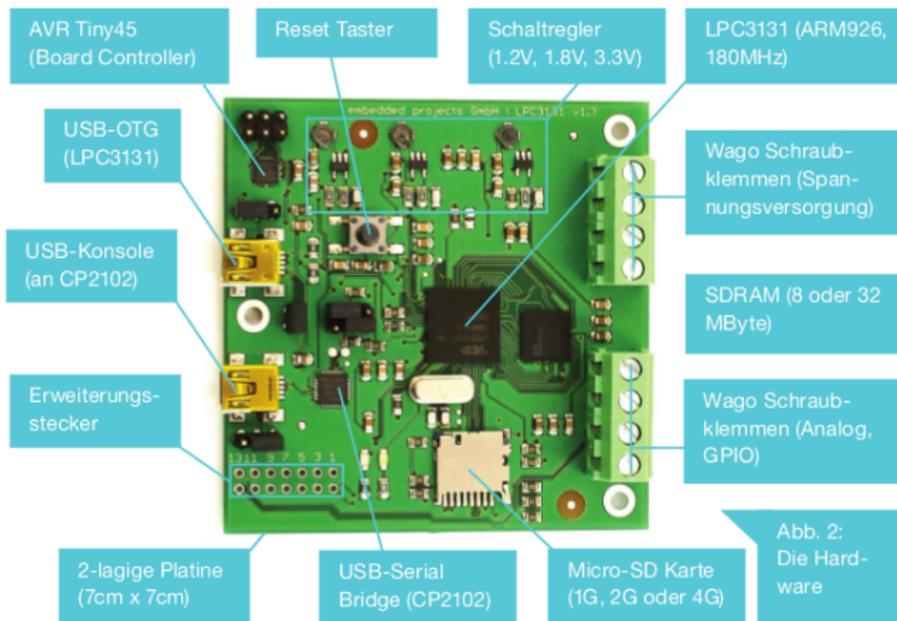
- günstiges Embedded Linux Board für Ausbildung
- einfacher Einstieg ohne große Vorkenntnisse
- ein einfaches USB Kabel soll ausreichen
- freie Lizenz für Hard- und Software
- Preis nicht höher als 50 Euro
- Unterstützung vieler Programmiersprachen
- Einfaches soll einfach, komplexe Aufgaben möglich sein
- gute Dokumentation

Hardware

- 180 MHz ARM9 / 32 MB RAM
- USB serielle Konsole On-Board (CP2102)
- Stromversorgung per USB oder Netzteil 7-12V
- microSD Karte
- 3x IO, 3x AD extern an Anschlussklemme
- I2C, SPI und UART
- USB Device oder Host (OTG)
- drei Versionen: Standard, Extended, DIP

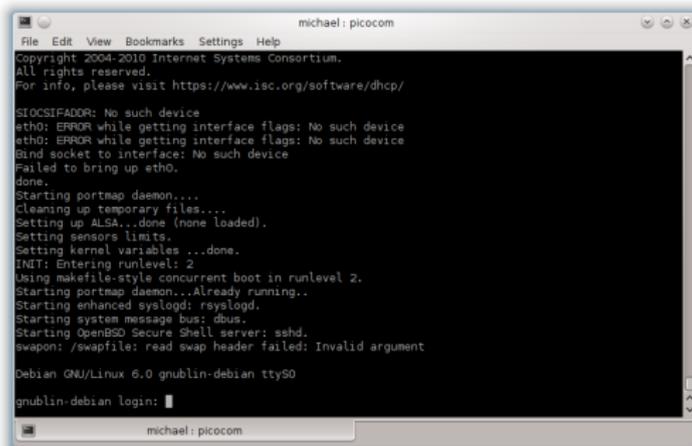


Hardware – Komponenten



Schnellstart

- 1 SD-Karte mit root-Dateisystem und Kernel in den Halter stecken
- 2 GnuBLIN über USB-Kabel mit Strom versorgen (RS232 Buchse)
- 3 Terminalemulator starten (z.B. picocom)
- 4 Linux-System benutzen



```
michael: picocom
File Edit View Bookmarks Settings Help
Copyright 2004-2010 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

STOCSIFADDR: No such device
eth0: ERROR while getting interface flags: No such device
eth0: ERROR while getting interface flags: No such device
Bind socket to interface: No such device
Failed to bring up eth0.
done.
Starting portmap daemon...
Cleaning up temporary files...
Setting up ALSA...done (none loaded).
Setting sensors limits.
Setting kernel variables ..done.
INIT: Entering runlevel: 2
Using makefile-style concurrent boot in runlevel 2.
Starting portmap daemon..Already running..
Starting enhanced syslogd: rsyslogd.
Starting system message bus: dbus.
Starting OpenBSD Secure Shell server: sshd.
swapon: /swapfile: read swap header failed: Invalid argument

Debian GNU/Linux 6.0 gnuBLIN-debian ttyS0
gnuBLIN-debian login: █

michael: picocom
```

Software-Komponenten

- Bootloader: apex
 - viele integrierte Kommandos
 - Erweiterung über neue Kommandos einfach möglich

- Kernel: Linux 2.6.33
 - neuere Kernel sollten ohne große Anpassungen nutzbar sein
 - Kernel mit Real Time Patches verfügbar
 - Kernel durch eigene Kernel Module erweiterbar
 - initramfs kann ebenfalls benutzt werden

- root-Dateisystem
 - Debian mit vielen Paketen
 - Pakete nachinstallierbar

USB

gnuBLIN als USB-Device

- gnuBLIN kann als USB-Device verwendet werden
- viele USB-Gadgets verfügbar unter Linux:
 - `g_ether`: Ethernet emulation on USB
 - `g_file_storage`: Mass storage
 - `g_serial`: Serial emulation on USB

gnuBLIN als USB-Host:

- an gnuBLIN können beliebige USB-Geräte angeschlossen werden
- allerdings USB-Host Adapter notwendig (ca. 2 Euro)
- z.B.: Webcam, WLAN, LAN, Massenspeicher, Audio



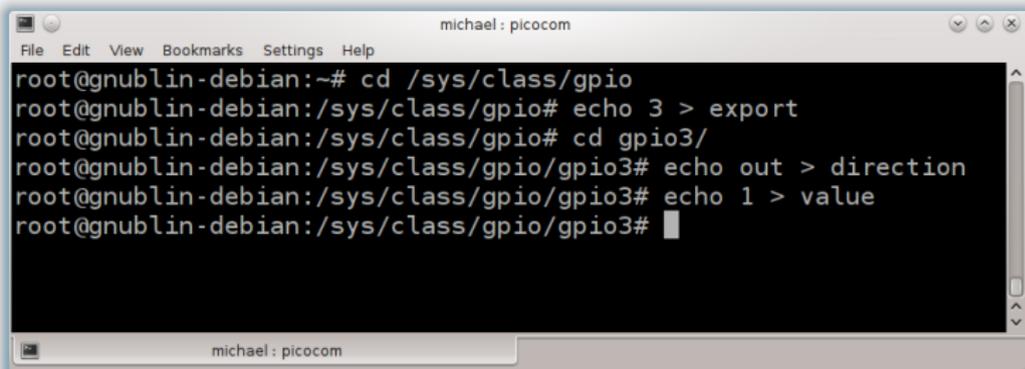
Programmiersprachen

Alle unter Linux verfügbaren Programmiersprachen:

- C, C++
 - Crosskompilierung auf dem PC (arm-linux-gnueabi-gcc)
 - Cross-Compiler kann durch einfaches Entpacken installiert werden
 - natives Kompilieren direkt auf dem Board (langsam, bei kleinen Programmen aber ohne Probleme möglich)
- Skriptsprachen
 - Python
 - Perl
 - Lua (spartanisch und sehr schnell)
 - PHP
 - Ruby
 - Bash, Shell
- Go
- theoretisch auch Java

Zugriff auf Hardware

- Alles ist eine Datei!
- z.B.: Ansteuern der GPIOs über sysfs:



```
michael : picocom
File Edit View Bookmarks Settings Help
root@gnublin-debian:~# cd /sys/class/gpio
root@gnublin-debian:/sys/class/gpio# echo 3 > export
root@gnublin-debian:/sys/class/gpio# cd gpio3/
root@gnublin-debian:/sys/class/gpio/gpio3# echo out > direction
root@gnublin-debian:/sys/class/gpio/gpio3# echo 1 > value
root@gnublin-debian:/sys/class/gpio/gpio3#
```

- I2C, SPI, ADC, UART ähnlich
- Zugriff von allen Programmiersprachen aus möglich

Anwendungen

Was kann man jetzt damit machen?

- gnuBLIN über WLAN-Stick mit dem eigenen Netzwerk verbinden
- Steuerung von Rolläden, Beleuchtung, Kaffeemaschine über Relais zu bestimmten Zeiten bzw. über Webinterface
- auch denkbar als Steuerung für Roboter oder Drohne
- gnuBLIN steuert Motoren, überwacht Sensoren
- Webcam zeigt Bilder, GPS liefert aktuelle Position
- Roboter/Drohne lässt sich über WLAN steuern



Vielen Dank!

Vielen Dank für Eure
Aufmerksamkeit!

Weitere Informationen finden sich unter wiki.gnuBLIN.org.