

# Embedded Linux

## Intro

Hubert Högl

Hochschule Augsburg / Informatik  
<http://hhoegl.informatik.hs-augsburg.de/hhwiki/FrontPage>

2025-03-25 08:06:23

## Worum geht es?

Geräte mit Hilfe von GNU/Linux bauen, die irgendeinen bestimmten Zweck erfüllen.

### Es kann so einfach sein ...

- Raspberry Pi und geeignetes Shield kaufen
- Ein paar Zeilen Python

### Bei genauerer Betrachtung jedoch ...

- 1 Produkt muss jahrelang reproduzierbar sein
- 2 Build Umgebungen vs. aktuelles GNU/Linux
- 3 Individuelles Root-Filesystem
- 4 Ansteuern von spezieller externer Hardware und Feldbussen
- 5 Echtzeit Timing Anforderungen im 1 ms bis 100  $\mu$ s-Bereich
- 6 Neue Sicherheitslöcher flicken, während die Anwendung in Betrieb ist
- 7 "Secure Boot"
- 8 Updates von Betriebssystem und Anwendung im Feld
- 9 Bootzeit optimieren
- 10 Ressourcenschonendes Programmieren, geringer Speicherverbrauch, geringer Stromverbrauch
- 11 Umfangreicher Test der Anwendung

## Worum geht es? (2)

- 12 Applikationsentwicklung kann sehr umfangreich sein; ähnlich wie im Desktop- und Server-Bereich
- 13 Gerätetreiber müssen entwickelt werden
- 14 Hardware muss industriellen Standards genügen
- 15 Tiefere Kenntnisse in der GNU/Linux Administration können nötig sein
- 16 Produkt benötigt "Glitzer-Grafik"
- 17 Prüfung ob Lizenzen der Software-Bestandteile zusammenpassen
- 18 Teile der neu entwickelten Software müssen in die offenen Upstream-Projekte eingespeist werden
- 19 Gerät muss evtl. aus 100% Open-Source Software bestehen
- 20 Intensives Debuggen von Hard- und Software kann nötig sein

Video von Pengutronix: **Was ist eigentlich Embedded?**

→ <https://www.youtube.com/watch?v=M1leRZBC9to>

## Warum Linux? Chancen und Risiken

Cliff Brake, Tips for planning an embedded Linux project, Feb. 2006

→ <https://hhoegl.informatik.hs-augsburg.de/elinux/intro/planning-tips.txt>

- Berechnungssteuerung
  - → <https://hhoegl.informatik.hs-augsburg.de/elinix/kurs/memo-01.pdf>
  - → <https://hhoegl.de/tsgrain.html>
- IO-Box (Semesterprojekt im Winter 2015)
  - XXX to do: iobox/IO-Box
- Eulenfunk (Hochschule Augsburg, Projekt im Jahr 2016)
  - → <https://github.com/studentkittens/eulenfunk>
  - → <https://hhoegl.informatik.hs-augsburg.de/elinix/Berichte/2016/eulenfunk/eulenfunk-doc.pdf>
- Streetpong (Pengutronix 2015)
  - → <https://hhoegl.informatik.hs-augsburg.de/elinix/application/schwebel-streetpong-2015>

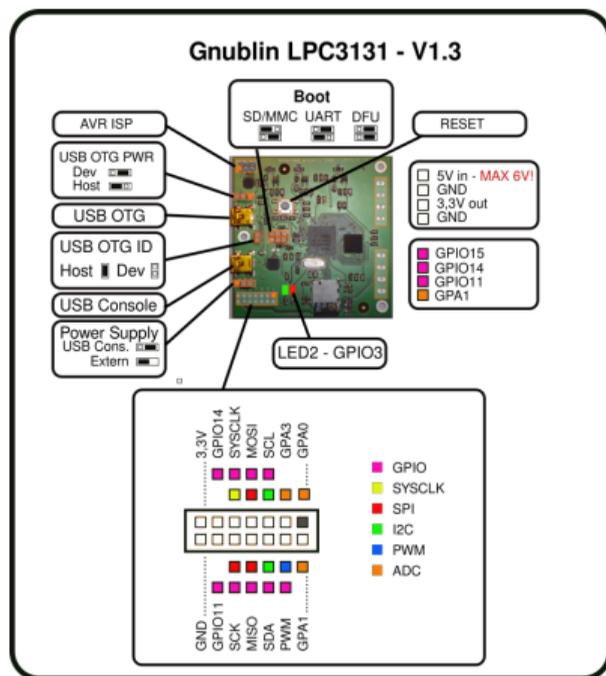


Figure 1: Gnublin Quickref

- Eigenentwicklung Embedded Projects GmbH (Augsburg) und Hochschule Augsburg
- 2012 - 2016
- NXP LPC3131 (ARM926EJ-S / ARMv5), max. 180 MHz
- 4M x 16 SDRAM (8 MByte)
- 5V, 0.1 A,  $P = U \cdot I = 0.5 \text{ W}$  (sehr stromsparend)
- USB 2.0 On-The-Go
- → [Schaltplan](#)
- → [Weiteres Archivmaterial](#)

Auswahl einiger Firmen, die im Embedded Linux Bereich tätig sind

- → <https://bootlin.com> (F)
- → <https://denx.de> (DE)
- → <https://linutronix.de> (DE)
- → <https://pengutronix.de> (DE)

Zusammenschluss von Firmen in einer Genossenschaft

- → <https://www.osadl.org> (DE)

viele weitere ... → <https://elinux.org/Companies>

### Mainstream Community Boards

- → <https://www.raspberrypi.com>
- → <https://beagleboard.org>

### Beispiele für Nischenhersteller

- → <https://www.acmesystems.it> (Microchip Cortex A5 und ARM9)
- → <https://www.olimex.com/Products/OLinuXino/open-source-hardware> (ARM9, Allwinner, i.MX)
- → <https://www.phytec.de> (NXP i.MX, TI AMxxxx, STM32MP1)

### Ein wenig Elinux-Geschichte:

<https://hhoegl.informatik.hs-augsburg.de/elinux/kurs/memo-02.pdf>

- → <https://tha.de/~hhoegl/home/ElinuxVorkenntnisse.html>
- Nehmen Sie dieses Wahlfach nur
  - wenn Sie sich für das Fach interessieren
  - wenn sie schon ein paar Vorkenntnisse haben
  - wenn sie Spaß an den Inhalten haben
  - wenn sie ernsthaft wöchentlich Arbeitszeit hineinstecken wollen

Ansonsten ist es vergeudete Zeit, vor allem wenn Sie nur 7,5 CP brauchen aber ihnen die Inhalte egal sind.



- Frank Vasquez, Chris Simmonds, Mastering Embedded Linux Programming, 3rd edition, 2021, Packt Publishing.

→ <https://learning.oreilly.com/library/view/mastering-embedded-linux/9781789530384>

- Derek Molloy, Exploring the Beagle Bone, 2nd edition, Wiley 2019.

→ <https://derekmolloy.ie/beaglebone>

- Klaus Dembowski, Raspberry Pi – Das technische Handbuch, 3. Auflage, Springer 2019.

→ <https://link-springer-com.ezproxy.hs-augsburg.de>

→ <https://hhoegl.informatik.hs-augsburg.de/nonfree/elinux> (Login/Pwd bekommen Sie von mir)

Alle Experimente, die Sie mit dem Board machen, u.a. die praktischen Übungen, beschreiben Sie knapp in einem Labor-Tagebuch. Das Laborbuch sollte ...

- alle praktischen Arbeiten beschreiben, die sie auf dem Hostrechner, auf dem Beaglebone Black, dem Raspberry Pi etc. machen. Es ist für Sie zur späteren Erinnerung gedacht und belegt auch, dass Sie die Übungen durchgeführt haben.
- eine freie Lizenz haben, z.B. die Creative Commons (<http://creativecommons.org>).
- in einem Markup Dialekt wie ReST oder Markdown geschrieben werden. Man kann z.B. die Werkzeuge *Pandoc* oder *Sphinx*. Der tiefere Sinn davon ist, dass Dokumentation wie Sourcecode in einem einfachen Textformat in Gitlab verwaltet werden kann. *Texte in Word oder LibreOffice akzeptiere ich nicht..*
- auf Gitlab in ihrem Repository liegen.
- Hier sind Vorlagen für ...  
Sphinx → <https://hhoegl.informatik.hs-augsburg.de/dva/sphinxbericht>
- Pandoc-Interessierten empfehle ich ausserdem das freie Buch von Michael Kofler:  
[https://kofler.info/ebooks/markdown\\_pandoc](https://kofler.info/ebooks/markdown_pandoc)

- Durch die Arbeit zu Hause während Corona haben wir weitgehend auf Projekte verzichtet. Wer Material zu Hause hat und ein Projekt machen möchte, kann dies aber gerne machen.
- Verzeichnis vergangener Projekte
  - <https://tha.de/~hhoegl/home/ElinuxProjekte>
- Unabhängig von Projekten gibt es **praktische Übungen**
  - <https://tha.de/~hhoegl/home/PflichtaufgabenBBB>

- → Antratek <https://www.antratek.de>
- → Eckstein <https://eckstein-shop.de>
- → Reichelt <https://www.reichelt.de>
- → Watterott <https://watterott.com>
- → Exp-Tech <https://exp-tech.de>
- → AZ-Delivery <https://www.az-delivery.de>
- → Berrybase <https://www.berrybase.de>

- 7,5 CP → ca. 240 Stunden Arbeit worst case

Bei ca. 14 Wochen bis zur Prüfung sind das ca. 16 Stunden bzw. 2 Arbeitstage die Woche.

**Sechs Stunden pro Woche reichen nicht!**

- Erfolgreiches Praktikum bedeutet Zulassung zur Klausur. Nachweis durch Labor-Tagebuch.
- Abschliessende Klausur in Moodle (Test mit Multiple-Choice und Freitext).

Fragenkatalog (wächst noch!)

→ <https://hhoegl.informatik.hs-augsburg.de/elixir/kurs/Fragen.html>

- Studierende aus Master Studiengängen schreiben zusätzlich eine Studienarbeit.

Jeder Kursteilnehmer hat ein Repository

XXX to do

Entfernter Zugang auf zwei Beaglebone Black im Labor.

→ <https://tha.de/~hhoegl/home/ElinuxRemote>

# Beagle Bone

Poster → <https://hhoegl.informatik.hs-augsburg.de/elix/bbb/BeagleBonePoster.pdf>

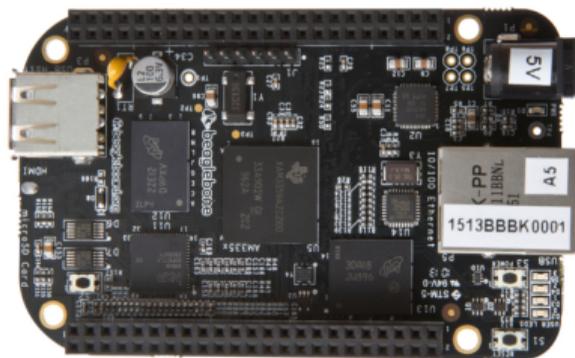


Figure 2: Beagle Bone Black

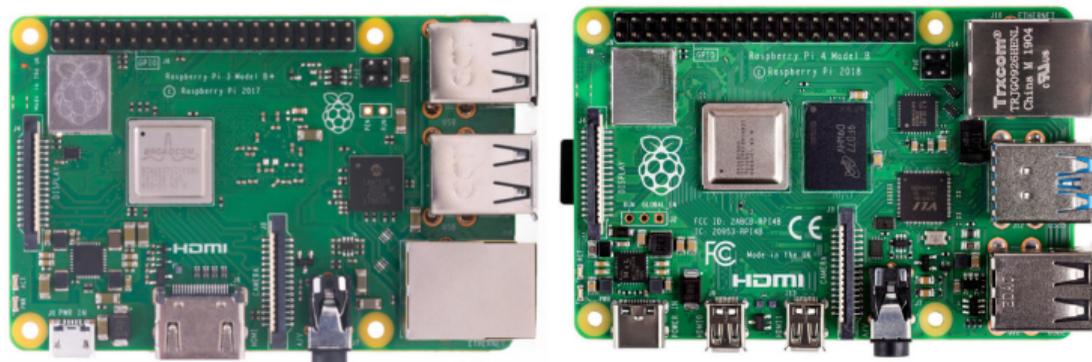
TI AM3358 single core Cortex-A8 32-Bit SoC 1 GHz (ARMv7).

**Open Source!**

→ <https://beagleboard.org/black>

→ <https://derekmolloy.ie/beaglebone>

# Raspberry Pi



Links: RPi 3 | Rechts: RPi 4

**RPi3 Model B+** Broadcom BCM2837B0, Quad core Cortex-A53 64-bit SoC 1.4GHz (ARMv8), 1 GByte RAM, LAN (Gigabit over USB 2.0), WiFi, Bluetooth 4.2.

**RPi4 Model B** Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC 1.5GHz om Quad core Cortex-A72 (ARM v8), 2/4/8 GByte RAM, WiFi, Bluetooth 5.0, Gigabit Ethernet.

→ <https://www.raspberrypi.com>

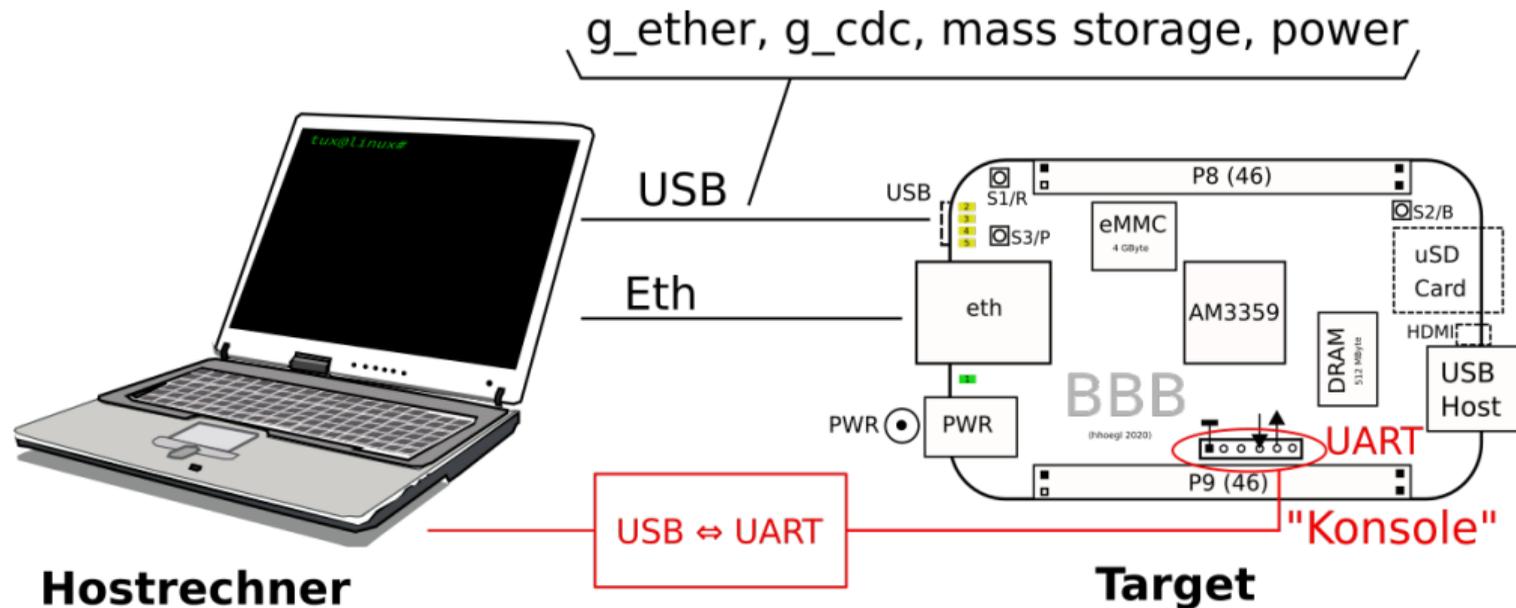


Figure 3: Host und Target