



Raspberry Pi für die professionelle Produktentwicklung

Eine gute Idee?

Anna-Lena Marx

Anna-Lena Marx



Embedded Systems Entwicklerin
seit 2019

B.Sc. Informatik

M.Sc. Embedded Systems

B.Eng. Elektrotechnik - in Progress

 @Allegra3141

 Allegra42

Warum überhaupt der Vortrag?



Ziele und Fokus

- › Fokus auf Produktsicht
 - ⇒ Tipps für Plattformauswahl
 - ⇒ Wie kann man ein zuverlässiges, wartbares Embedded-Produkt entwickeln?
- › Kein Raspberry Pi Bashing

Standard Raspberry Pis

Consumer Fokus

- › eigentlich Lernplattform
- › Hardware Design mit Consumer-Fokus
 - RAM und CPU Leistung steigen
 - Schnittstellen
 - schnelle Release-Zyklen
 - ⇒ nicht immer stabil

Langzeitverfügbarkeit



SD Karten

- › Zuverlässigkeit
 - der Karte selbst \Rightarrow Schreibzyklen
 - der mechanischen Verbindung der Kontakte
- › Umgang mit Ausfall der Karte im Feld
- › Schutz der Software gegen Veränderung

Energieversorgung

- › Micro-USB / USB-C mechanisch fragil
- › USB-C (Pi): 5V, 3A
 - Industrie: 12-24V
 - USB-C PD: 20 V, 5A max.

Features

Standard Pis kommen ohne:

- › Echtzeit-Uhr (RTC)
- › Trusted Platform Module (TPM)
- › Flash oder eMMC Speicher
- › Power over Ethernet (PoE)
- › Analog/Digital-Wandler
- › externer Watchdog

Umweltbedingungen

Environment:

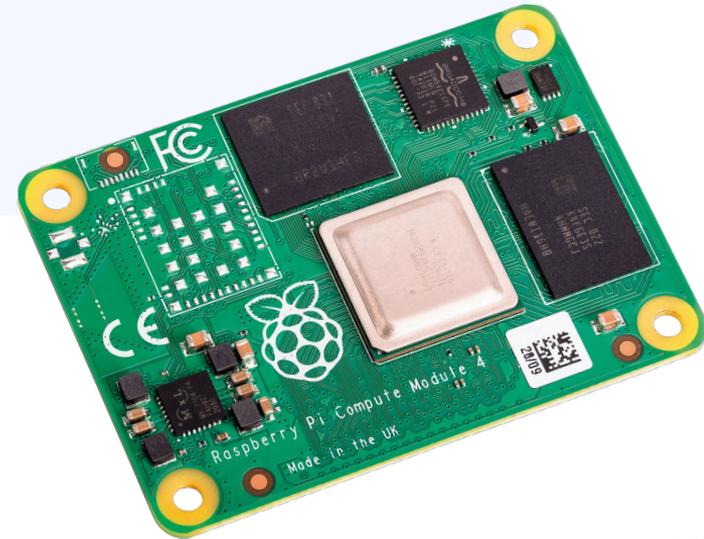
Operating temperature 0–50°C

- › übliche Temperaturbereiche:
 - Commercial: 0°C - 70°C
 - Industrial: -40°C - 85°C
 - Military: -55°C - 125°C

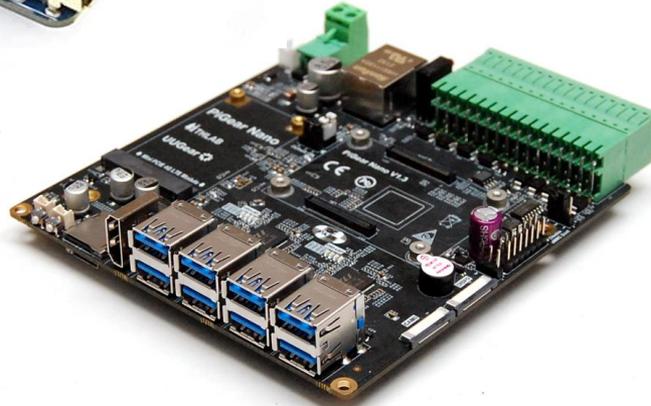
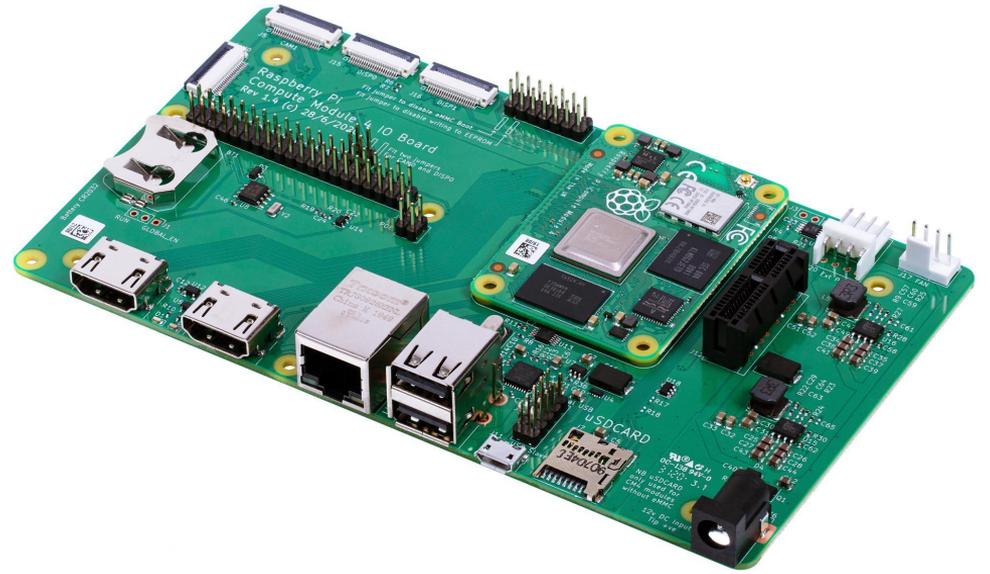
- › Instabil bei Überhitzung



Compute Modules



Base Boards



Industrial Raspberry Pis

Industrial Raspberry Pis

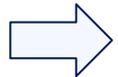
Für Produktentwicklung selten attraktiv.

- › oft Fokus auf Automatisierung (Privat/kleine Betriebe)
- › hoher Preis
- › wichtige Schnittstellen fehlen

Software Welt

Updates

- › deterministische Updates
- › Roll-Back Mechanismus
- › Kontrolle über Versionen



Firmware sollte auf Images statt Paketen basieren

Reproduzierbarkeit

Jede Software Version soll Jahre später erneut und “1:1” nachgebaut werden können.

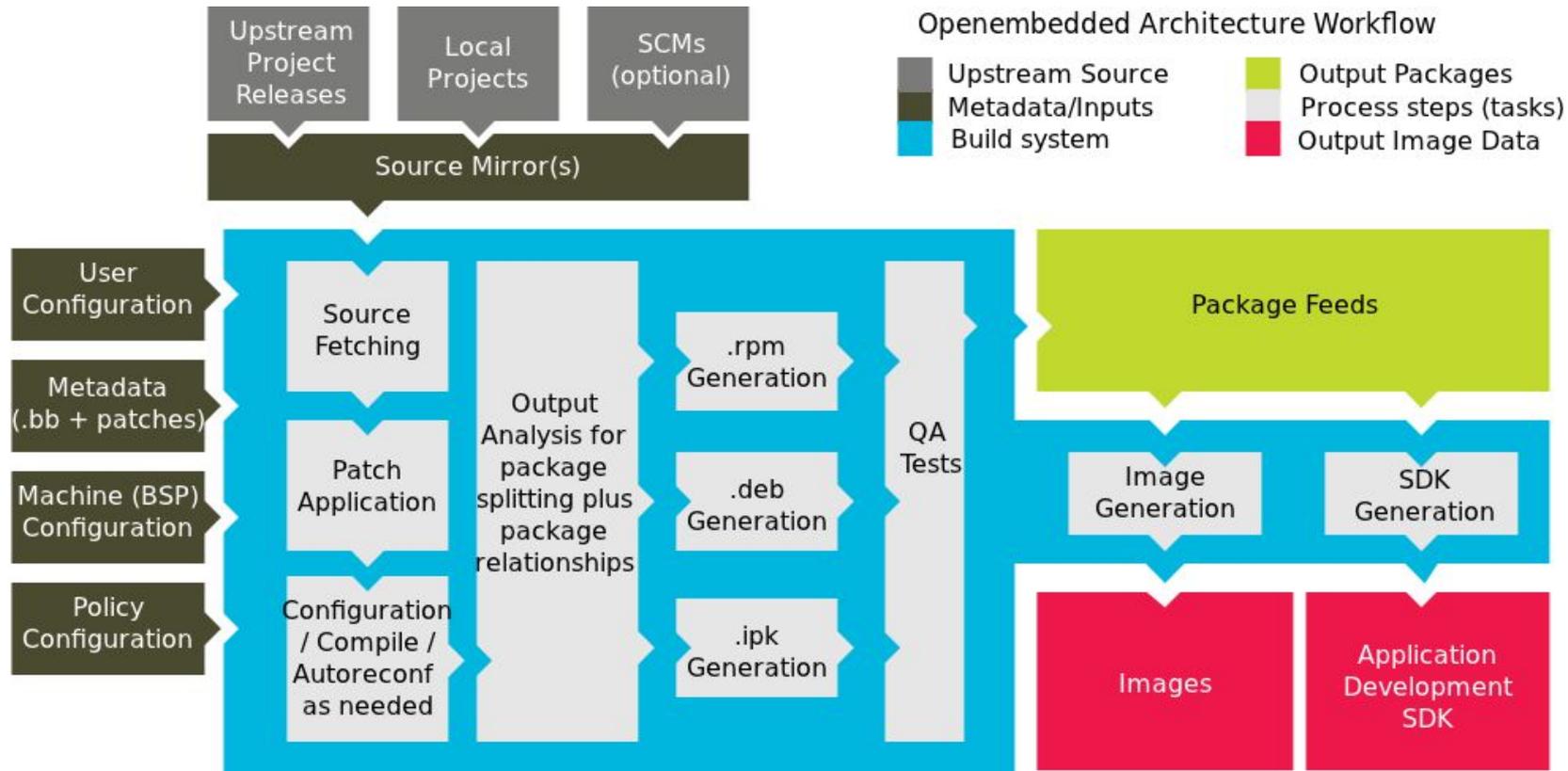
- › Versionierbarkeit
 - › Golden Image Problematik
- ⇒ Build Systeme

Lizenzen und Support

- › problematische Lizenzen erkennen
 - proprietäre Lizenzen
 - GPLv3
- › Software Bill of Materials
 - Lizenzübersicht generieren
- › kein professioneller Support von der Raspberry Pi Foundation

Embedded Build Systems

Yocto



Buildroot

Buildroot is a simple, efficient and easy-to-use tool to generate embedded Linux systems through cross-compilation.



Can handle everything

Cross-compilation toolchain, root filesystem generation, kernel image compilation and bootloader compilation.



Is very easy

Thanks to its kernel-like menuconfig, gconfig and xconfig configuration interfaces, building a basic system with Buildroot is easy and typically takes 15-30 minutes.



Supports several thousand packages

X.org stack, Gtk3, Qt 5, GStreamer, Webkit, Kodi, a large number of network-related and system-related utilities are supported.

Und nun?

Die wichtigsten Fragen

- › passt der Pi tatsächlich zu den Anforderungen?
- › passen garantierte Lebenszeit und Verfügbarkeit?
- › sind SD-Karten, Energieversorgung und Umweltbedingungen wirklich kein Problem?

Und falls es doch der Pi sein soll

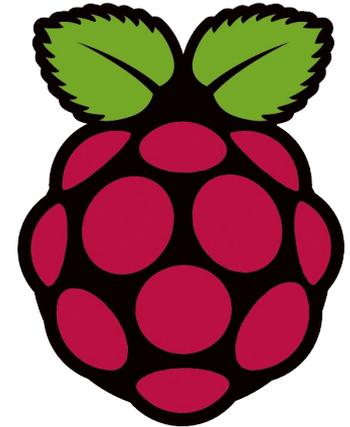


statt



yocto .
PROJECT

statt



Mehr erfahren?

Blog

Raspberry Pi for Products



Blog

Embedded Build Systems



Vielen Dank!

Anna-Lena Marx
Embedded Systems
Entwicklerin

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

anna-lena.marx@inovex.de

