



Microkontroller- & Linux-Boards Ein Überblick

Jochen Krapf

KNF-Kongress 2014

Agenda

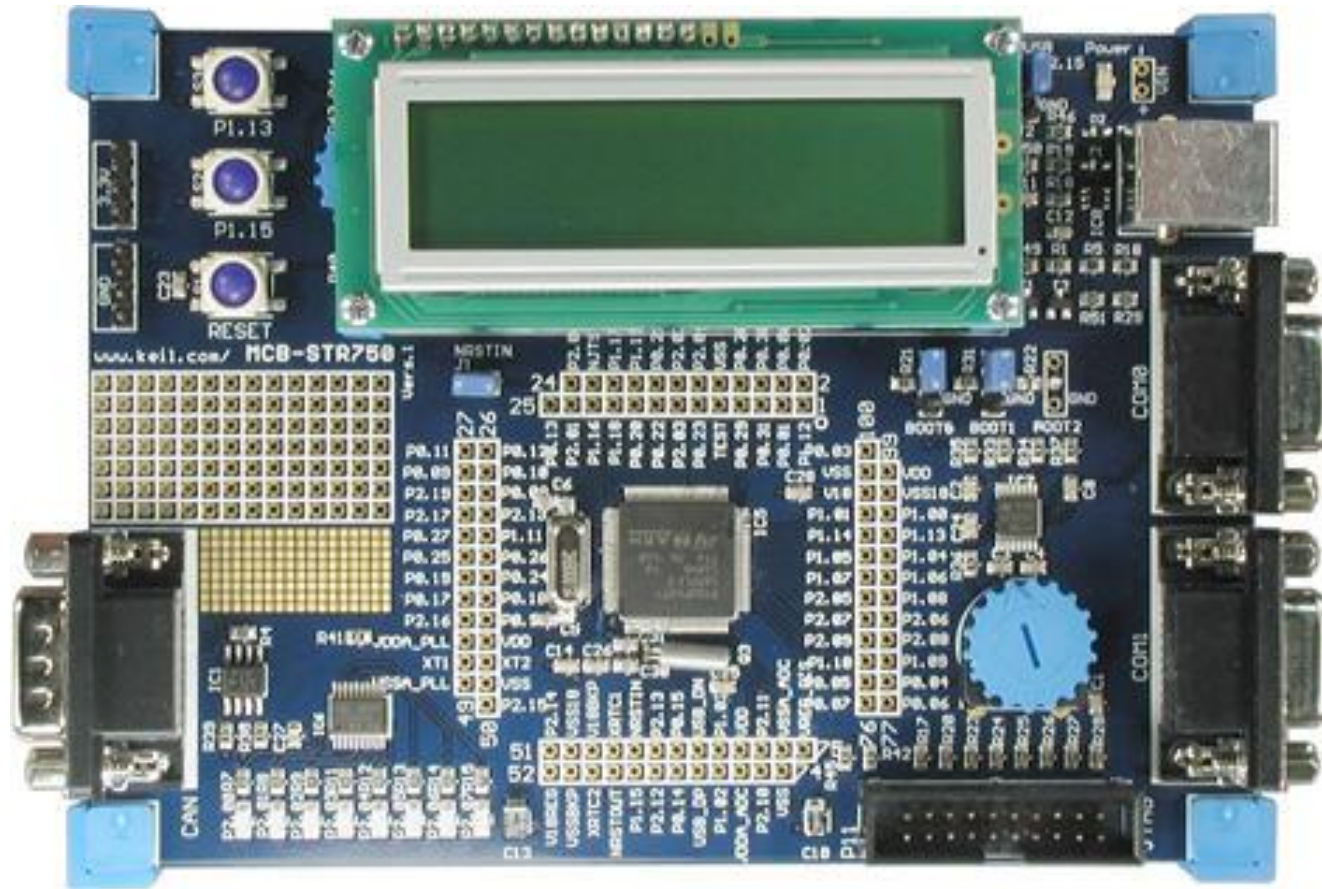
- **Boards**
 - **Arduino-Boards**
 - **ARM-Boards**
 - **Linux für Multimedia**
 - **Linux für Steuerungen**
- **Wann Was Wo**
- **Programme für eigene HW**

Boards:



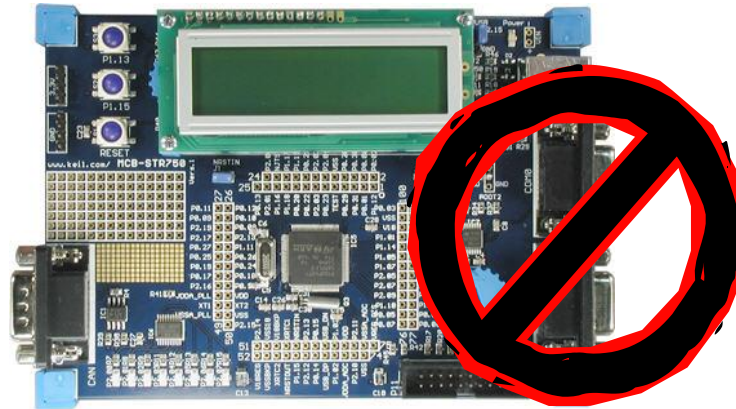
Boards

Boards: viele...



Quelle: keil.com

Boards: Gruppen



Development-Boards



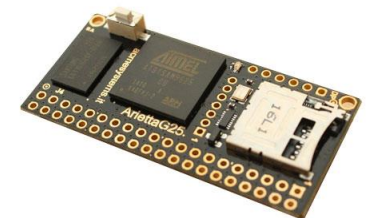
Arduino-Boards



ARM-Boards



Linux
für Multimedia

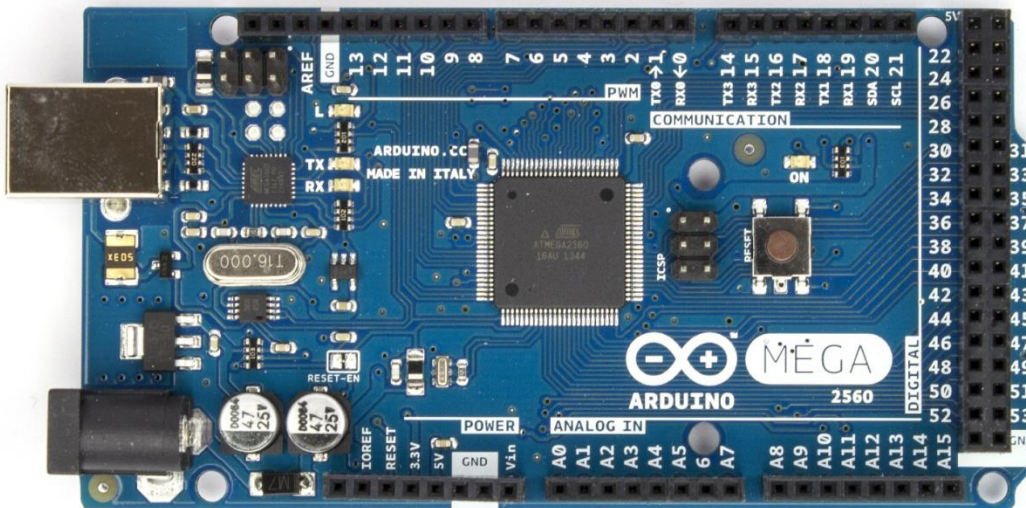
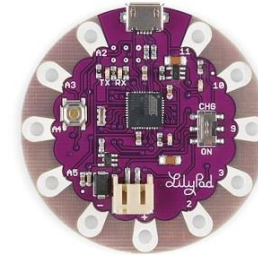


Linux
für Steuerungen



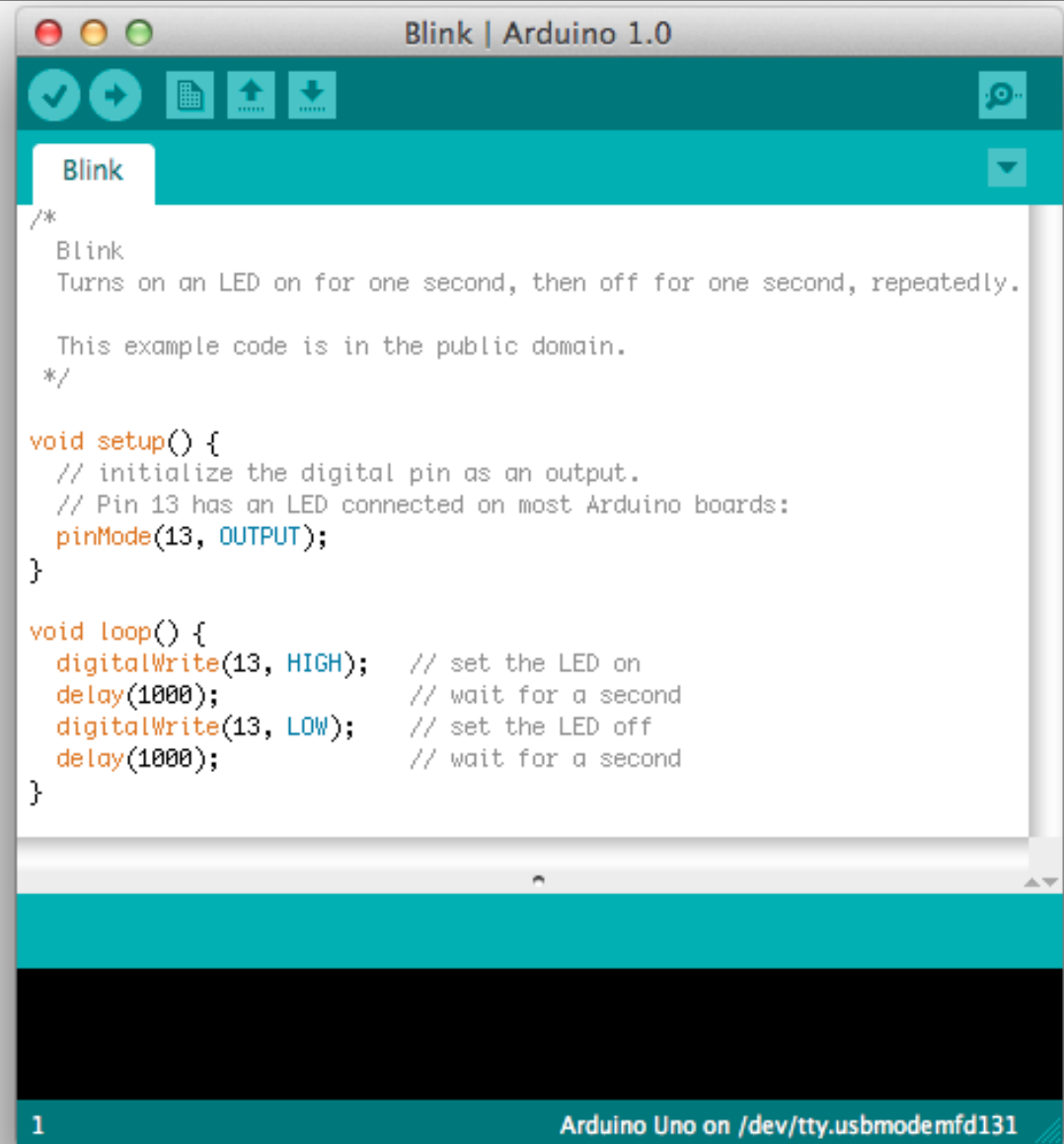
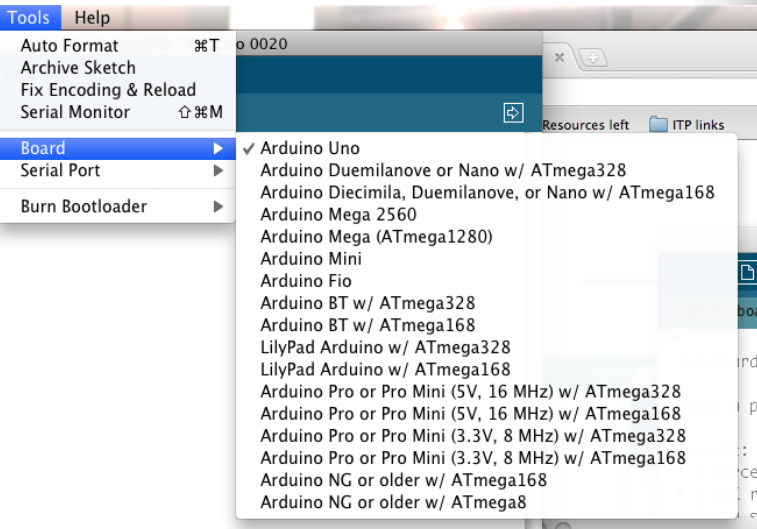
Arduino-Boards

μ C: Arduino - Plattform



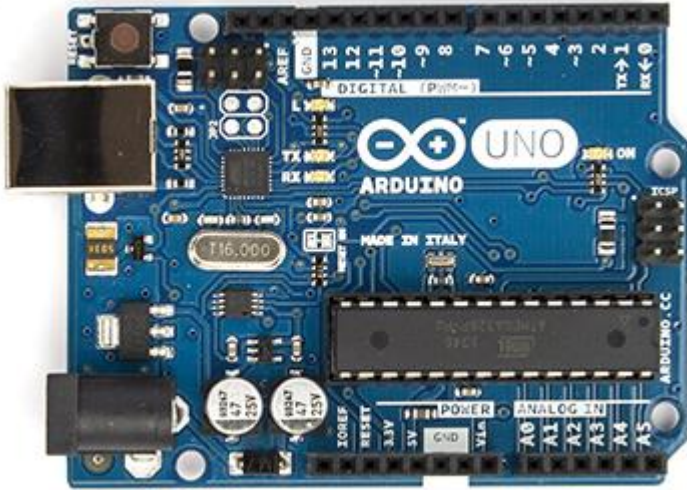
Quelle: arduino.cc

μC: Arduino - IDE



<http://arduino.cc/>

μ C: Arduino – UNO ... MEGA ...

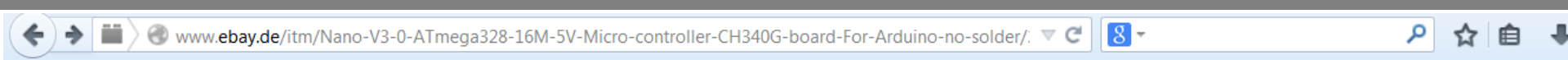


- AVR 8-Bit RISC @ 16 MHz*
- 2 ... 32 kB RAM,
32 ... 256 kB Flash, EEPROM
- GPIO, UART, I2C, SPI, (USB),
ADC, PWM
- Preis: 15 ... 25 €

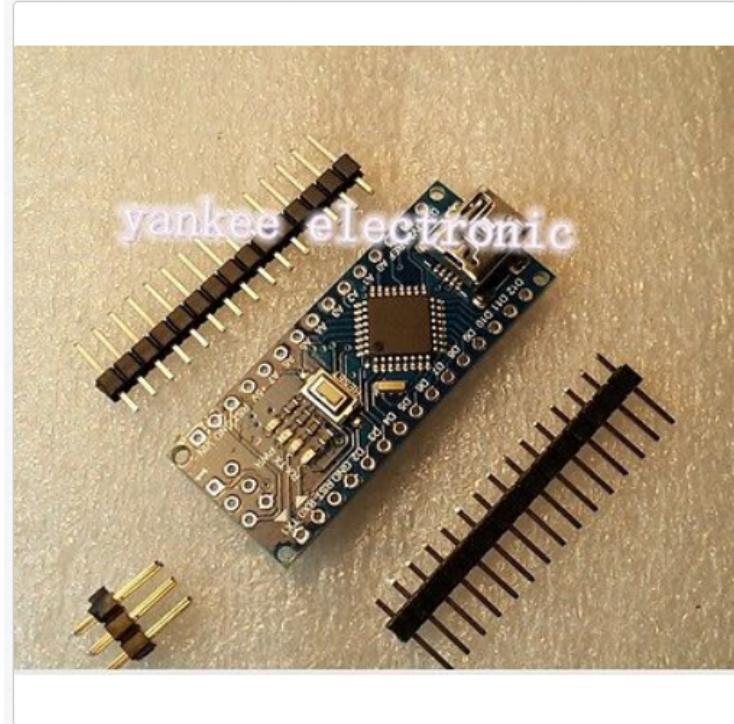
* ohne Arduino DUE

<http://arduino.cc/>

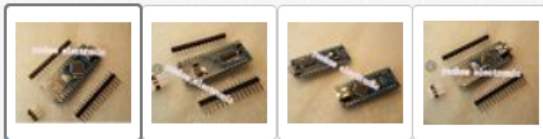
μC: Arduino – aus China...



Entwicklungswerkzeuge & -Kits



Klicken, um das Bild zu vergrößern und weitere Ansichten zu sehen



Nano V3.0 ATmega328 16M 5V Micro-controller CH340G board For Arduino no solder

🔥 4 verkauft in den letzten 24 Stunden

Artikelzustand: **Neu**

Stückzahl:

Mehr als 10 verfügbar
316 verkauft

EUR 3,15

Sofort-Kaufen

In den Warenkorb

Auf die Beobachtungsliste

★ **Zur Kollektion hinzufügen**

52 Beobachter

**Kostenloser
Inlandsversand**

316 verkauft

**Erfahrener
Verkäufer**

Versand: **Kostenlos** Versandoptionen aus dem Ausland |
[Weitere Details](#)

Artikelstandort: Shenzhen, China

Versand nach: Weltweit [Ausschlussliste anzeigen](#)

Lieferung: Zwischen **Fr, 5. Dez. und Di, 23. Dez.** bei heutigem Zahlungseingang

📧 📘 🐦 📌 | Auf die Beobachtungsliste

Angaben zum Verkäufer

yankee_electronic (26037) ⭐

99,4% Positive Bewertungen

+ Diesem Verkäufer folgen

[Andere Artikel aufrufen](#)

Shop besuchen: [yankee_electronic](#)

Angemeldet als gewerblicher Verkäufer

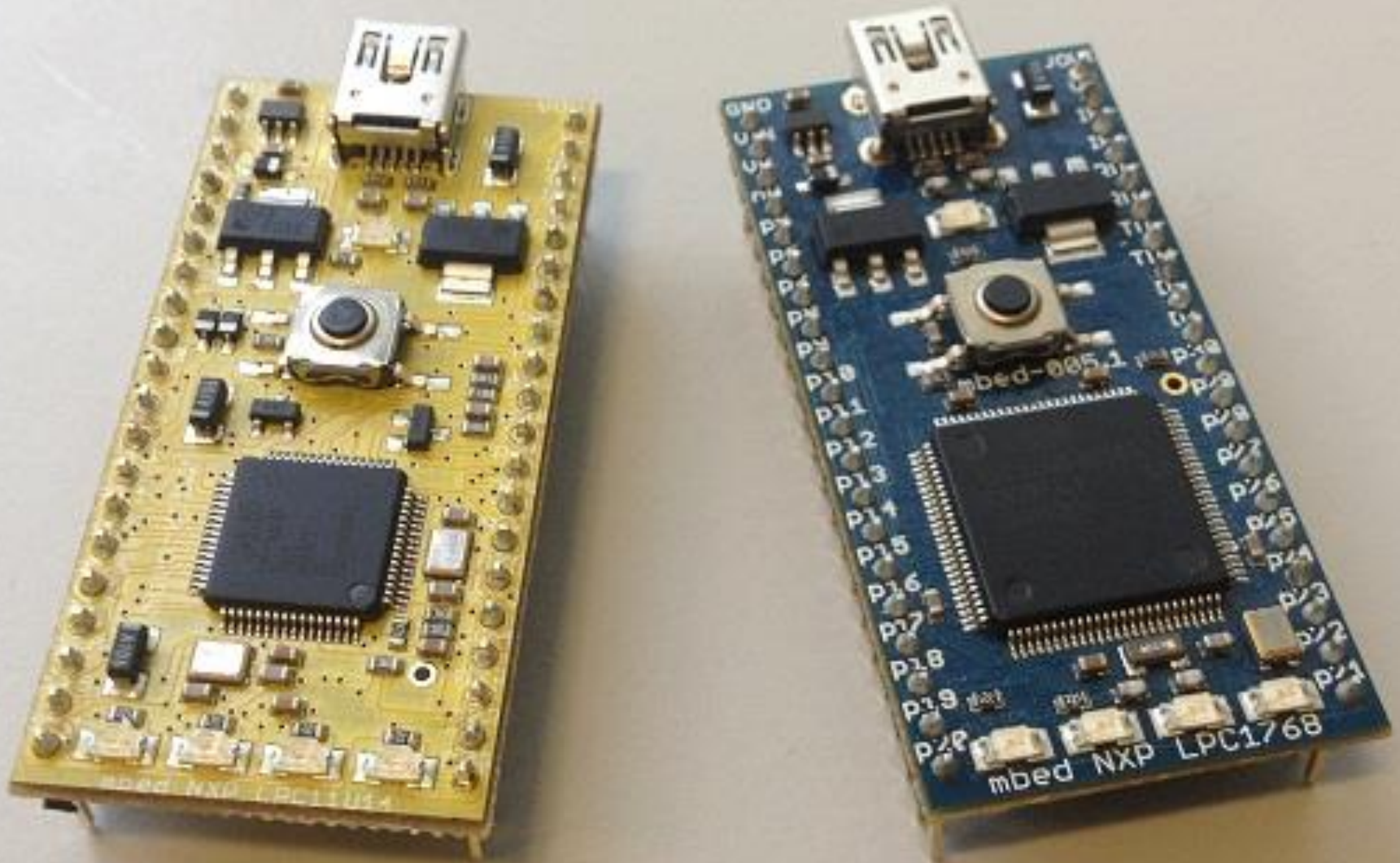
🏠 Ähnlichen Artikel verkaufen? [Selbst verkaufen](#)

Quelle: ebay.de



ARM-Boards
(Cortex M0, M3, M4)

μ C: mbed - Plattform



Quelle: mbed.org

μC: mbed - Plattformen

 <p>mbed LPC1768</p> <ul style="list-style-type: none"> Cortex-M3, 96MHz 512KB Flash, 32KB RAM 	 <p>mbed LPC11U24</p> <ul style="list-style-type: none"> Cortex-M0, 48MHz 32KB Flash, 8KB RAM 	 <p>FRDM-KL25Z</p> <ul style="list-style-type: none"> Cortex-M0+ 128KB Flash, 16KB RAM USB OTG 	 <p>NXP LPC800-MAX</p> <ul style="list-style-type: none"> Cortex-M0+ 16KB Flash, 4KB RAM 	 <p>ST Nucleo F302R8</p> <ul style="list-style-type: none"> Cortex-M4, 72MHz 64-KB Flash, 16-KB SRAM 	 <p>ST Nucleo L152RE</p> <ul style="list-style-type: none"> Cortex-M3, 32MHz 512-KB Flash, 80-KB SRAM 	 <p>ST Nucleo L053R8</p> <ul style="list-style-type: none"> Cortex-M0+, 48MHz 64-KB Flash, 8-KB SRAM 	 <p>ST Nucleo F401RE</p> <ul style="list-style-type: none"> Cortex-M4, 84MHz 512-KB Flash, 96-KB SRAM
 <p>EA LPC4088 QuickStart Board</p> <ul style="list-style-type: none"> Cortex-M4, 120MHz 512KB Flash, 96KB SRAM 	 <p>DipCortex M0</p> <ul style="list-style-type: none"> Cortex-M0, 50MHz 32KB Flash, 8KB RAM 	 <p>DipCortex M3</p> <ul style="list-style-type: none"> Cortex-M3, 72MHz 64KB Flash, 12KB RAM 	 <p>BlueBoard-LPC11U24</p> <ul style="list-style-type: none"> Cortex-M0, 48MHz 32KB Flash, 8KB RAM 	 <p>ST Nucleo F030R8</p> <ul style="list-style-type: none"> Cortex-M0, 48MHz 64-KB Flash, 8-KB SRAM 	 <p>ST Nucleo F072RB</p> <ul style="list-style-type: none"> Cortex-M0, 48MHz 128-KB Flash, 16-KB SRAM 	 <p>ST Nucleo F334R8</p> <ul style="list-style-type: none"> Cortex-M4, 72MHz 64-KB Flash, 16-KB SRAM 	 <p>FRDM-K64F</p> <ul style="list-style-type: none"> Cortex-M4, 120MHz 1MB Flash, 256KB RAM Ethernet, SD Filesystem
 <p>WiFi DipCortex</p> <ul style="list-style-type: none"> Cortex-M3, 72MHz 64KB Flash, 12KB RAM 	 <p>Seeeduino-Arch</p> <ul style="list-style-type: none"> Cortex-M0, 48MHz 32KB Flash, 8KB RAM 	 <p>mbed LPC1114FN28</p> <ul style="list-style-type: none"> Cortex-M3, 48MHz 32KB Flash, 4KB RAM 	 <p>u-blox C027</p> <ul style="list-style-type: none"> Cortex-M3, 96MHz 512 KB Flash, 32KB RAM Onboard cellular module 	 <p>Nordic nRF51822</p> <ul style="list-style-type: none"> Bluetooth v4.1 Cortex-M0, 16MHz 128KB Flash, 16KB RAM 	 <p>FRDM-KL05Z</p> <ul style="list-style-type: none"> Cortex-M0+, 48MHz 32KB Flash, 4KB RAM 	 <p>LPCXpresso1549</p> <ul style="list-style-type: none"> Cortex-M3, 72MHz 256KB Flash, 36KB RAM Arduino Formfactor headers 	 <p>TG-LPC11U35-501</p> <ul style="list-style-type: none"> Cortex-M0, 48MHz 64KB Flash, 10KB RAM
 <p>EA LPC11U35 QuickStart Board</p> <ul style="list-style-type: none"> Cortex M0, 48MHz 64KB Flash, 10KB RAM 	 <p>ST Nucleo F103RB</p> <ul style="list-style-type: none"> Cortex-M3, 72MHz 128-KB Flash, 20-KB SRAM 	 <p>FRDM-KL46Z</p> <ul style="list-style-type: none"> Cortex-M0+, 48MHz 256KB Flash, 32KB RAM USB OTG 	 <p>Seeeduino-Arch-Pro</p> <ul style="list-style-type: none"> Cortex-M3, 96MHz 512KB Flash, 32KB RAM 	 <p>LPCXpresso11U68</p> <ul style="list-style-type: none"> Cortex-M0+, 50MHz 256KB Flash, 36KB RAM Arduino Formfactor headers 	 <p>Seed Xadow M0</p> <ul style="list-style-type: none"> Cortex-M0, 48MHz 64KB Flash, 10KB RAM 	 <p>Seed Arch BLE</p> <ul style="list-style-type: none"> Bluetooth Low Energy Cortex M0, 16MHz 128KB Flash, 16KB RAM 	 <p>ST Nucleo F411RE</p> <ul style="list-style-type: none"> Cortex-M4, 100MHz 512-KB Flash, 128-KB SRAM

Quelle: mbed.org



μ C: mbed – IDE (Browser)

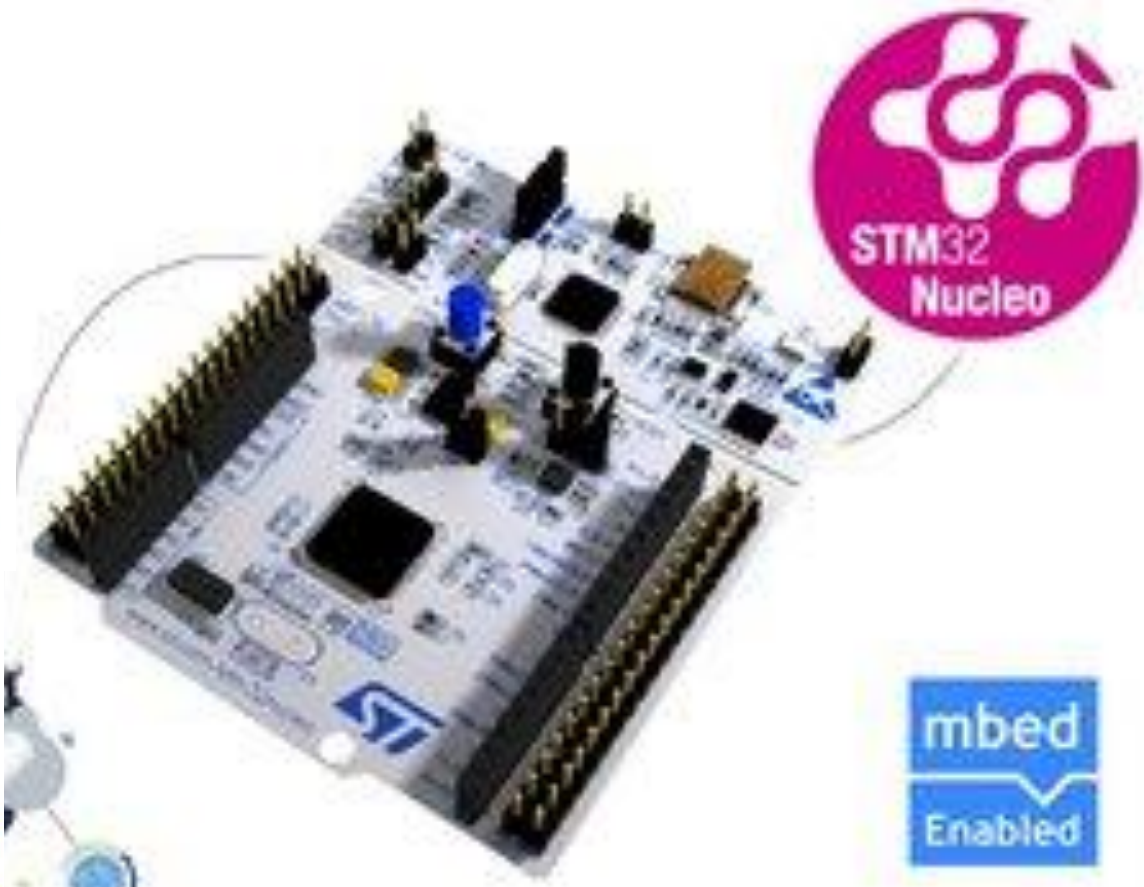
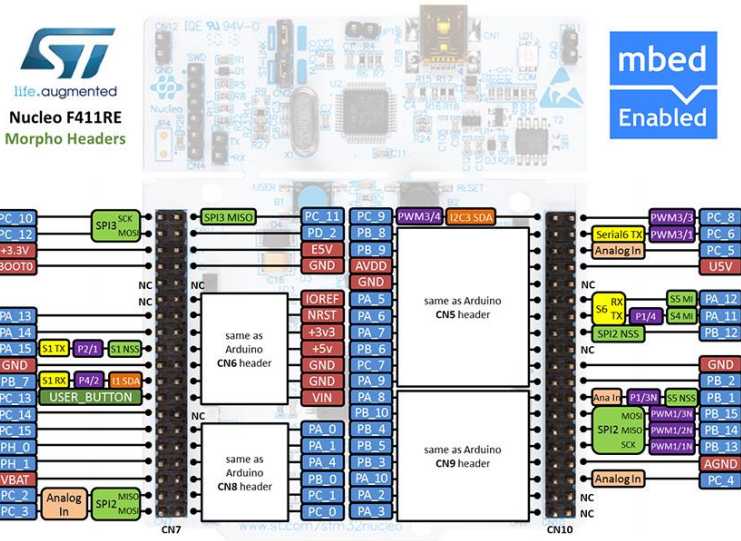
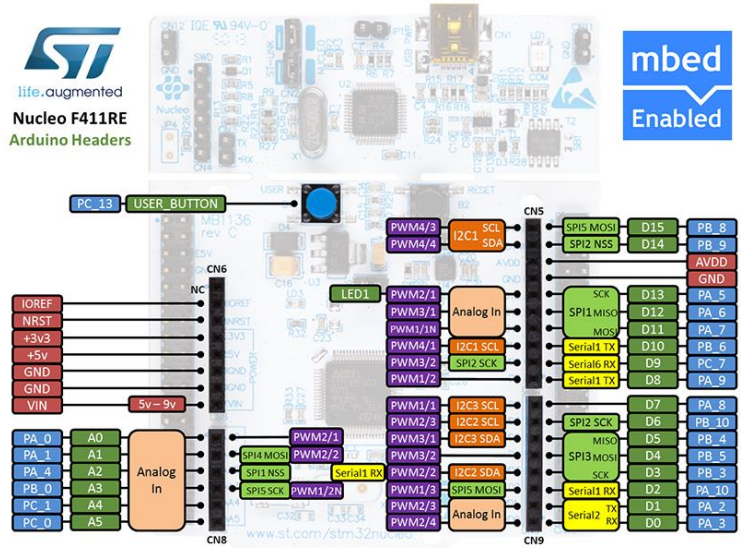
The screenshot displays the mbed IDE browser interface. The top bar shows the file path `/Float_SQRT_test/main.cpp` and the target device `ST Nucleo F401RE`. The menu bar includes options like `New`, `Import`, `Save`, `Save All`, `Compile`, `Commit`, and `Revisions`.

The **Program Workspace** on the left lists various projects, with `Float_SQRT_test` and its `main.cpp` file selected. The main editor window shows the following C++ code:

```
31 int main()
32 {
33     for (int i=0; i<15; i++) {
34         pwm[i].period_us(200);
35         pwm[i].pulsewidth_us(10*i+10);
36     }
37
38 //     double v = 0; // 20µs at Cortex M4 (nucleo 401)
39 float v = 0; // 0.3µs at Cortex M4 (nucleo 401)
40 int i;
41
42 while(1) {
43
44     v += 1234567.89f;
45
46     for ( i=0; i<500; i++ ) {
47         v = sqrt ( v );
48     }
49 }
```

A tooltip for the `sqrt` function is visible, showing its signature: `double sqrt (double);`. Below the code editor, the **Compile output for program: Float_SQRT_test** panel shows 0 errors, 0 warnings, and 0 infos. A table with columns `Description`, `Error Number`, `Resource`, and `In Folde` is present but empty. At the bottom, a status bar indicates `Ready.` and the current cursor position: `ln 47 col 19 61 INS`.

μC: mbed – ST Nucleo F411RE



Quelle: mbed.org

μ C: mbed – ST Nucleo F411RE



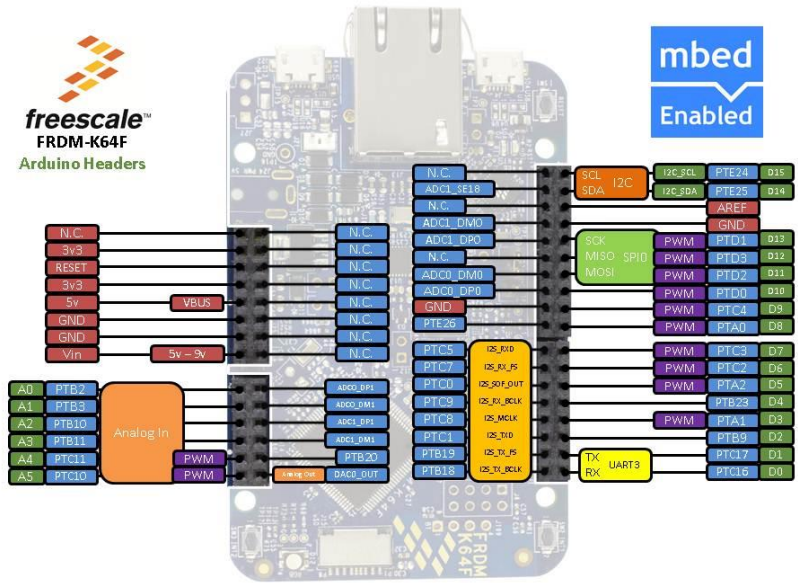
- ARM Cortex **M4** @ 100 MHz, **FPU**
- 128 kB RAM, 512 kB Flash
- 3x I2C, 3x USARTs, 5x SPIs, SDIO, 5x I2S, USB, 12bit ADC
- Preis: **13 €**

<http://developer.mbed.org/platforms/ST-Nucleo-F411RE/>

μC: mbed – FRDM-K64F



freescale™
FRDM-K64F
Arduino Headers



Quelle: mbed.org

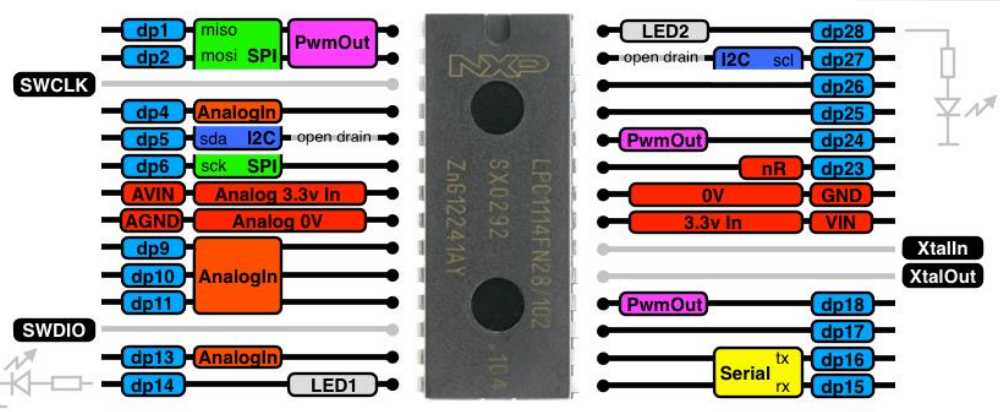
μ C: mbed – FRDM-K64F



- ARM Cortex **M4** @ 120 MHz, **FPU**, DSP
- 256 kB RAM, 1 MB Flash
- 3x I2C, 5x USARTs, 3x SPIs, I2S, USB OTG, 16bit ADC, 12bit DAC, CAN, **LAN**
- Preis: ~50 €

<http://developer.mbed.org/platforms/FRDM-K64F/>

μC: mbed – LPC1114FN28



Quelle: mbed.org

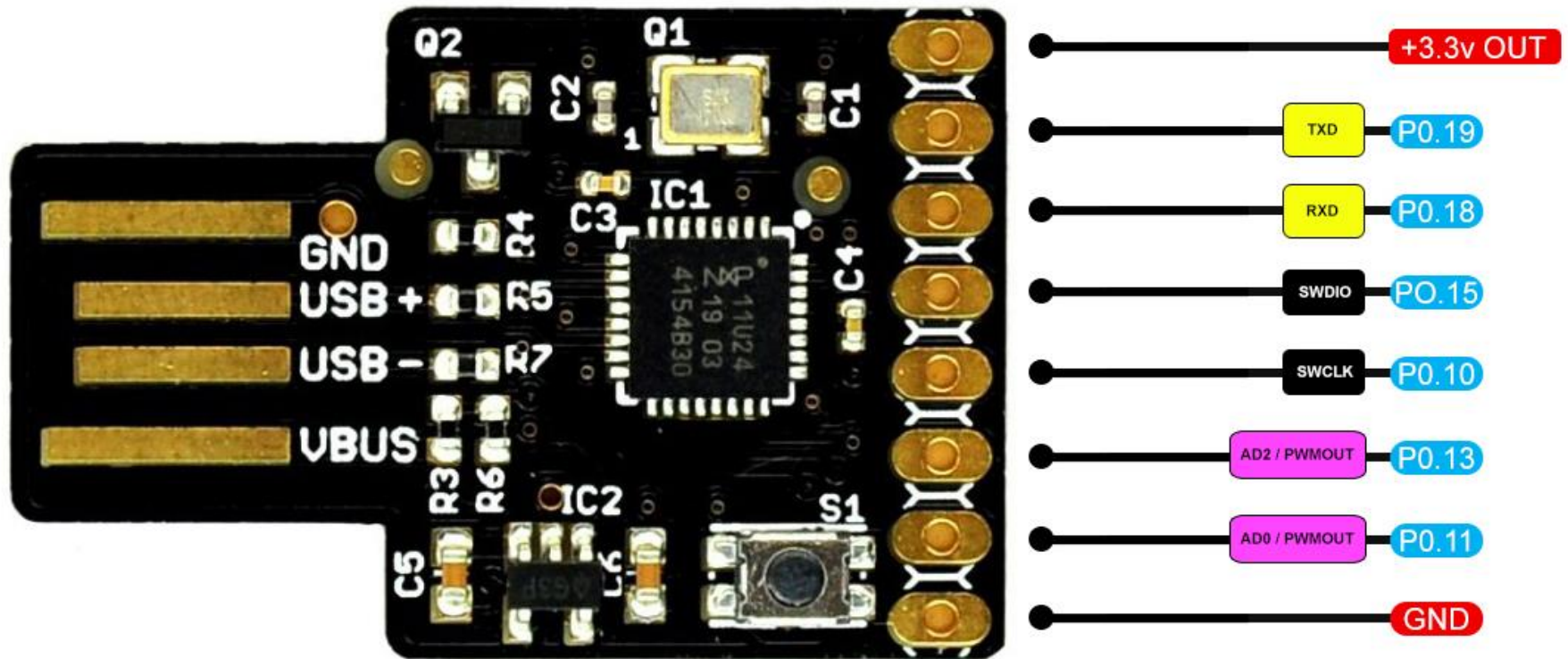
μ C: mbed – LPC1114FN28



- **ARM Cortex M0 @ 48 MHz, ohne Quarz**
- **4 kB RAM, 32 kB Flash**
- **I2C, USARTs, SPIs, I2S, 12bit ADC**
- **Preis: ~5 € (nur Chip)**

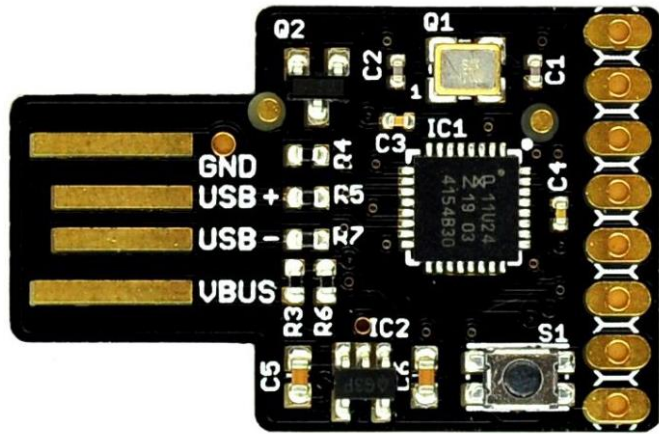
<http://developer.mbed.org/platforms/LPC1114FN28/>

μ C: mbed – USB Prank



Quelle: mbed.org

μ C: mbed – USB Prank



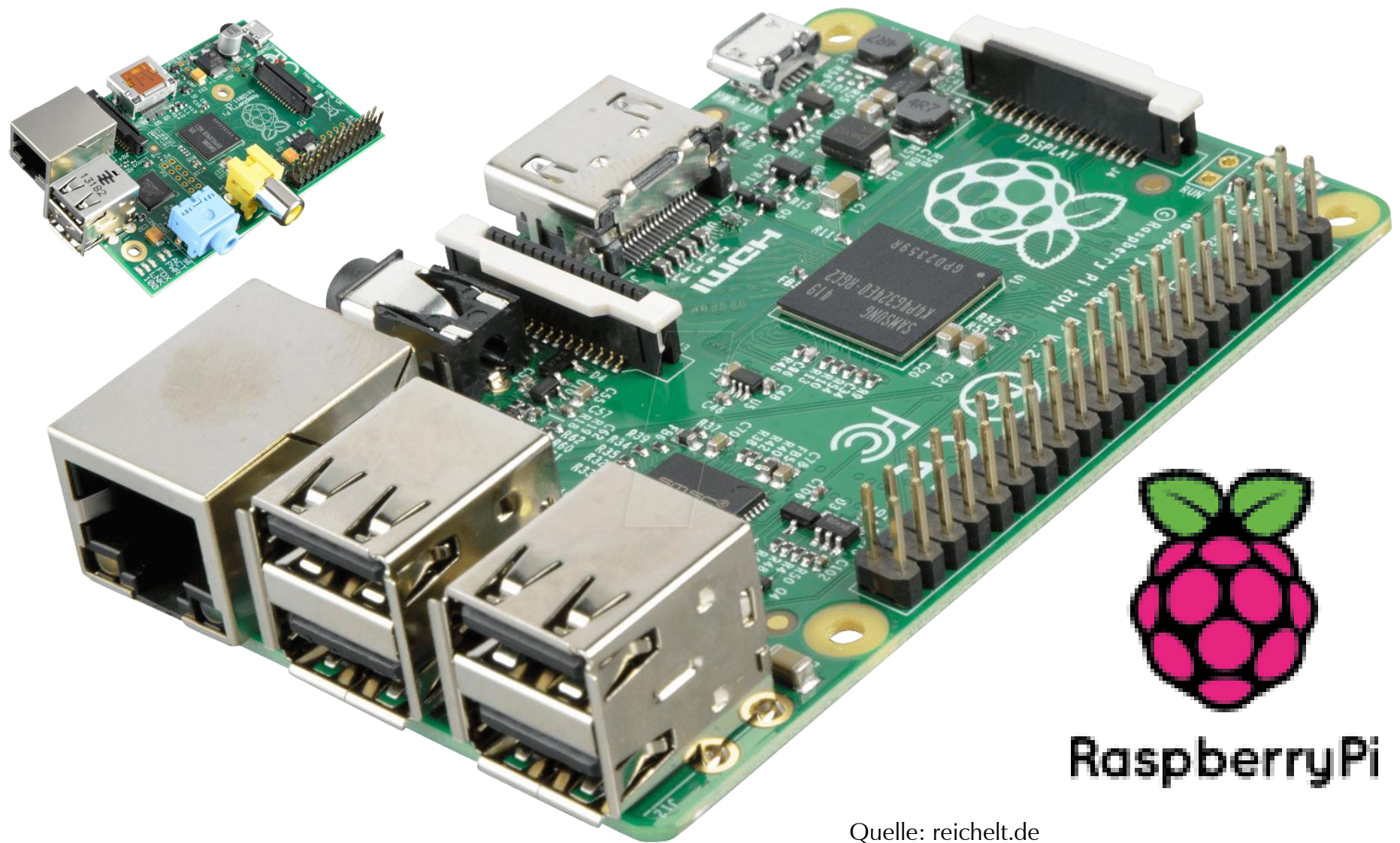
- ARM Cortex M0 @ 50 MHz
- 8 kB RAM, 32 kB Flash
- USART, 12bit ADC, PWM, **USB(Device)**
- Preis: **\$10**
(+ \$10 Porto aus USA)

<http://www.outrageouscircuits.com/shop/product/7>



**Linux-Boards
für Multimedia
(Cortex A7, A9)**

μ Linux: Raspberry Pi (Rev B+)

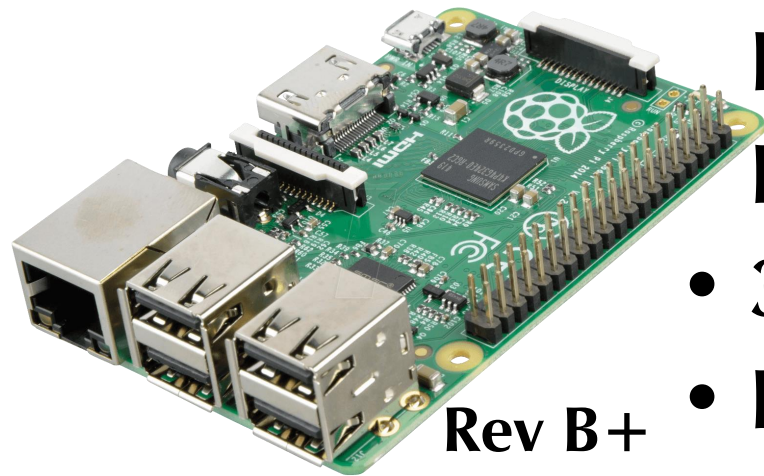


Quelle: reichelt.de

μ Linux: Raspberry Pi (Rev B+)



Rev B

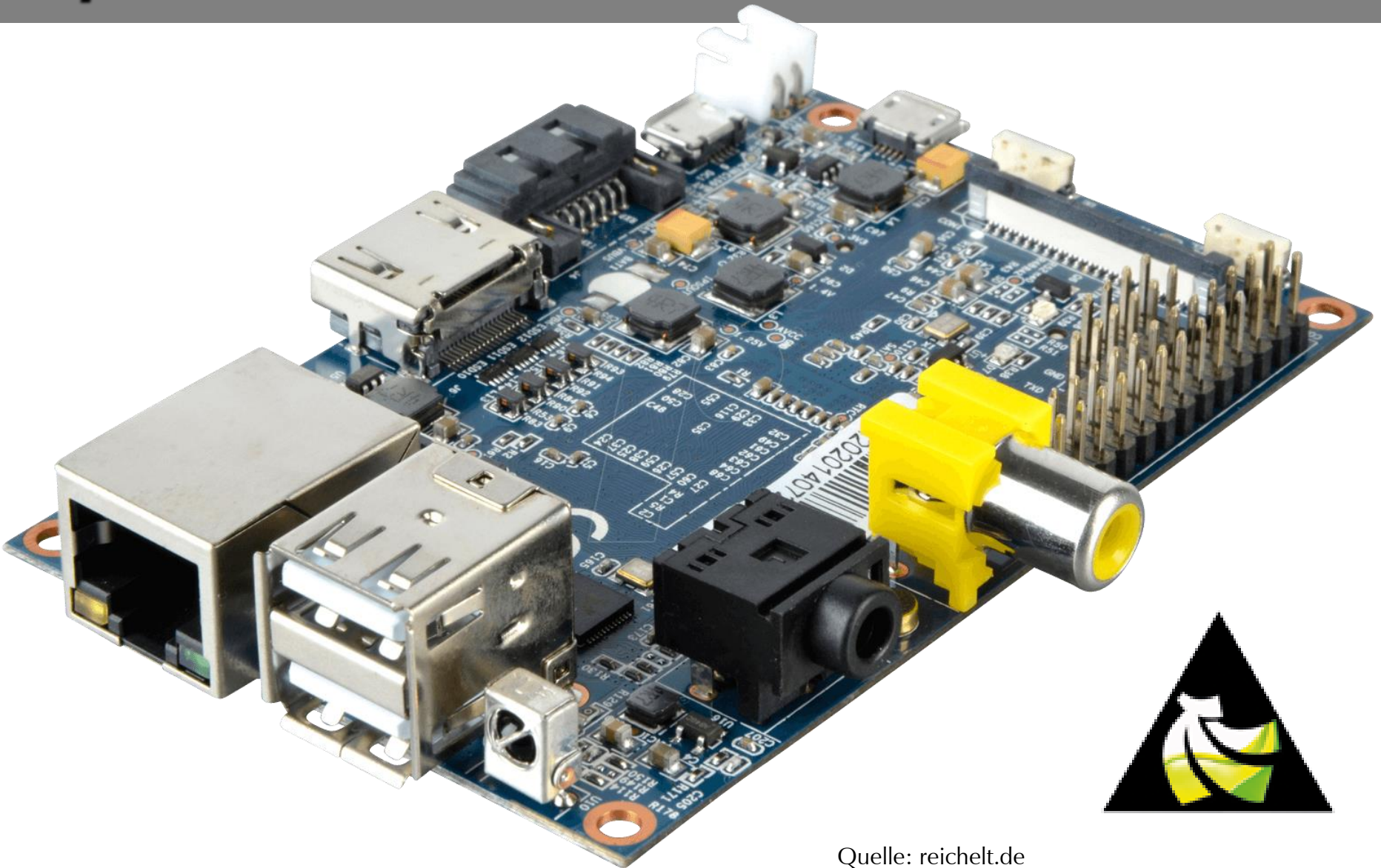


Rev B+

- ARM 1176JZFS @ 700 MHz
- Linux: Noops, Raspbian, Raspbmc, Pidora, Openelec...
- 512 MB RAM, microSD
- GPIO, UART, SPI, I2C, 4x USB, Ethernet (100), LVDS, Cam, HDMI (**H264 Decoder**) Audio
- 3.5 ... 5 Watt @ 5V
- Preis: ~30 €

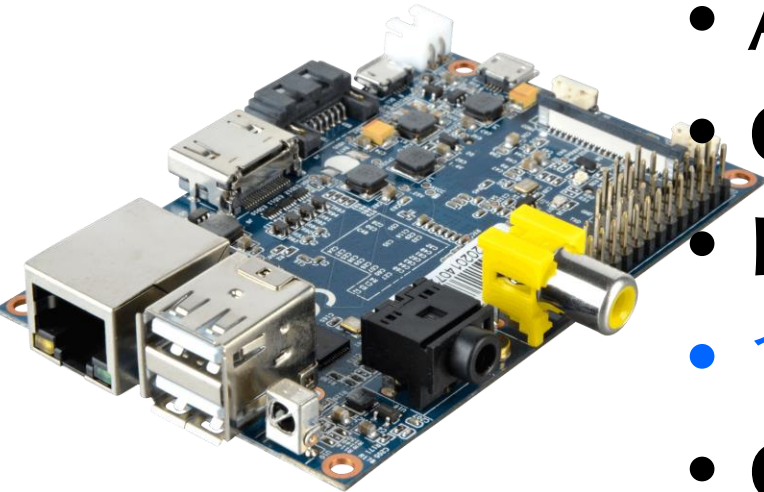
<http://www.raspberrypi.org/>

μ Linux: Banana Pi

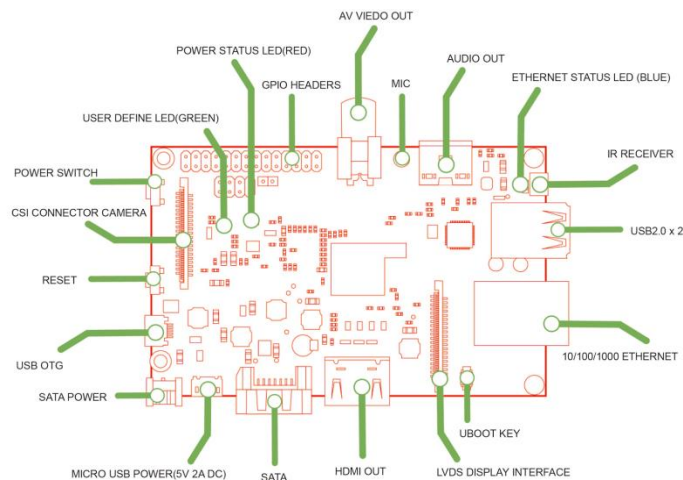


Quelle: reichelt.de

μ Linux: Banana Pi

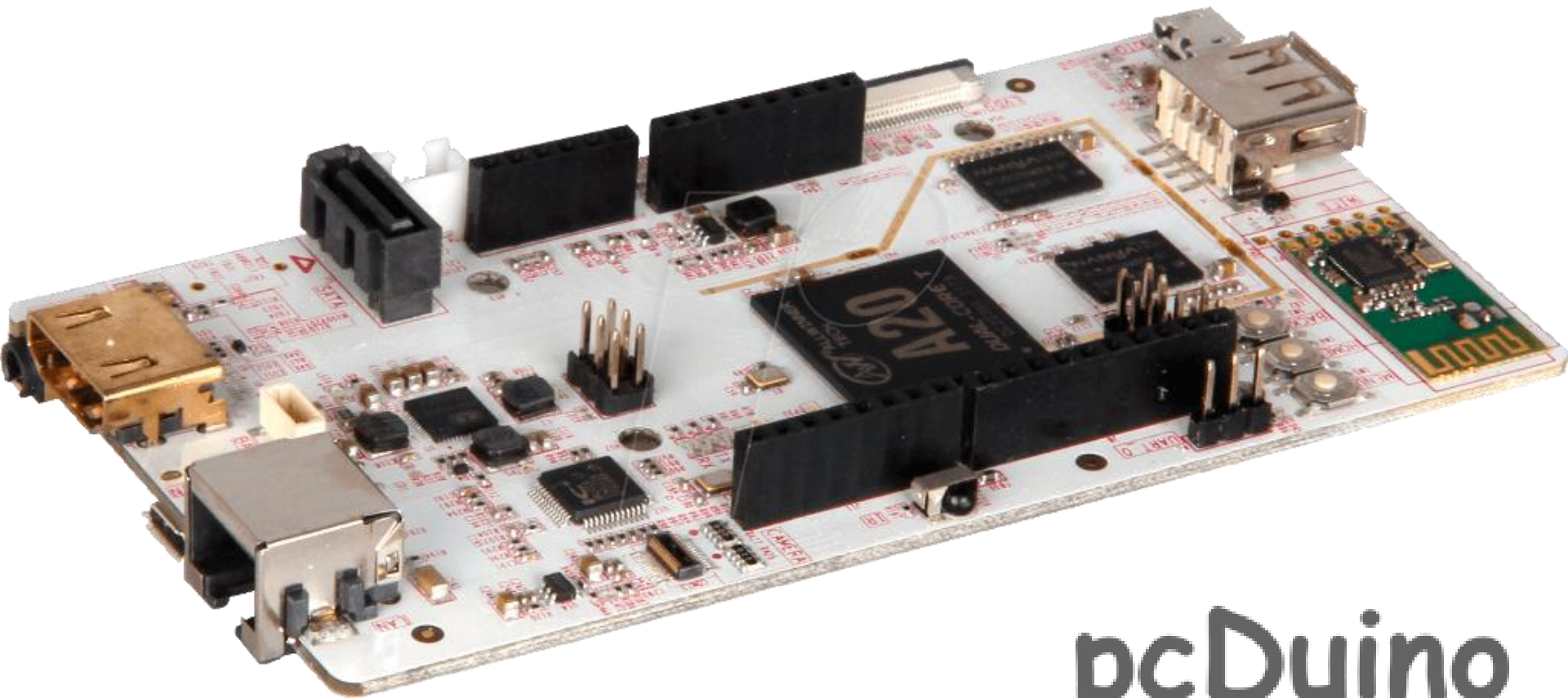


- ARM Cortex A7 Dual @ **1 GHz**
- GPU: OpenGL ES 2.0/1.1
- Linux: Debian, Raspbien, Arch
- **1 GB RAM, SD/MMC, SATA**
- GPIO, UART, SPI, I2C,
2x USB, Ethernet (**1000**), LVDS,
Cam, HDMI, Audio
- ~5 Watt @ 5V
- Preis: ~45 €



<http://www.bananapi.org/>

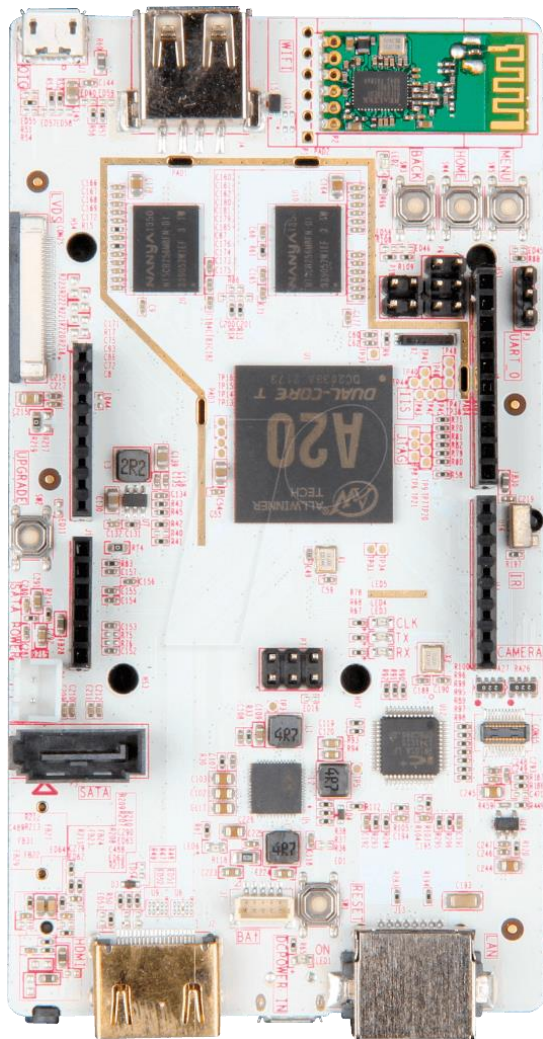
μ Linux: pcDuino



pcDuino

Quelle: reichelt.de

μ Linux: pcDuino

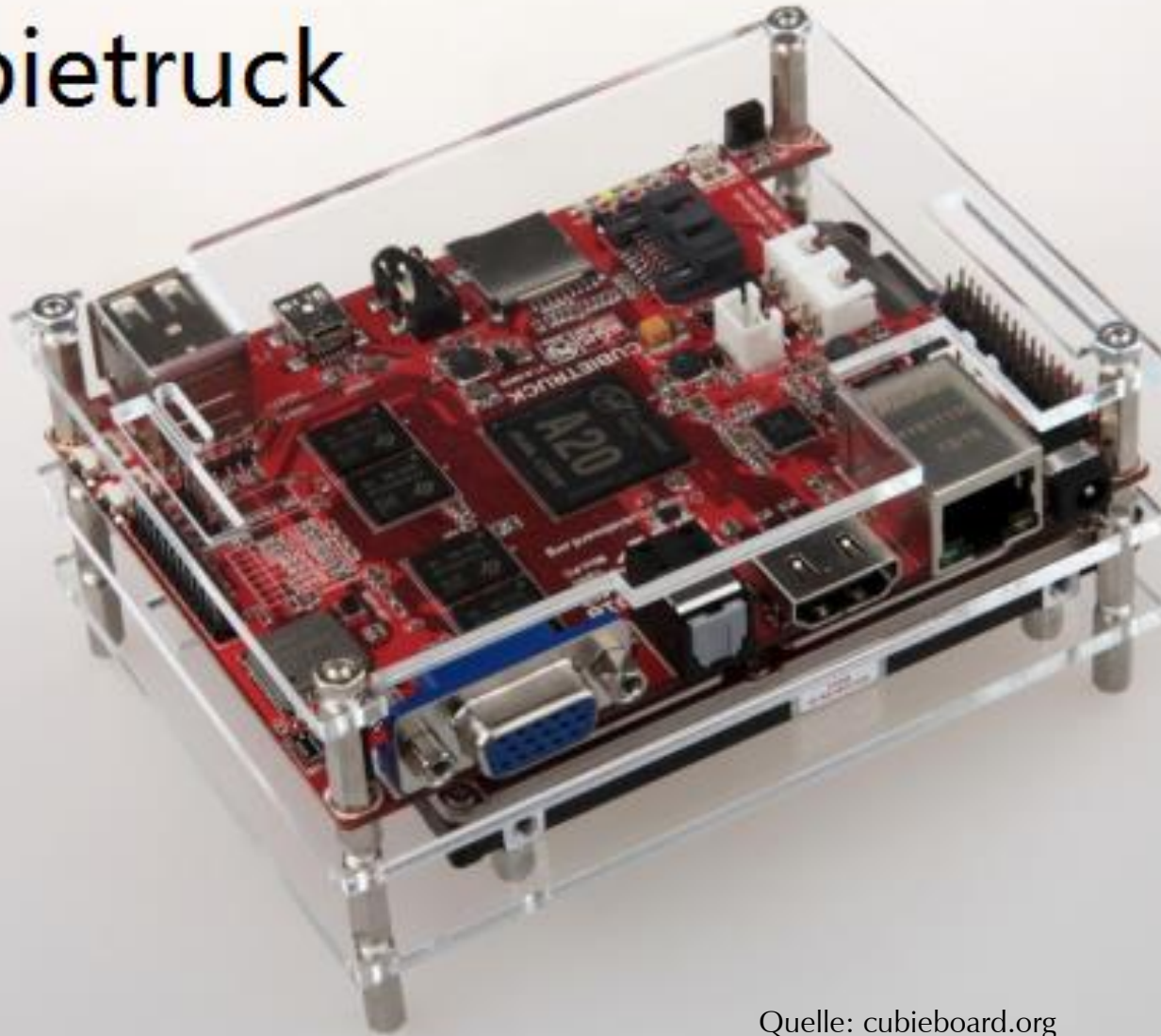


- ARM Cortex A7 Dual @ **1 GHz**
- GPU: OpenGL ES2.0, OpenVG
- Linux: Ubuntu 12.04
- **1 GB** RAM, 4 GB Flash, **SATA**
- GPIO, UART, SPI, I2C,
2x USB, Ethernet (100), **WiFi**,
LVDS, HDMI, Audio, **Arduino**
- ~5 Watt @ 5V
- Preis: ~50 €

<http://???>

μ Linux: CubieTruck

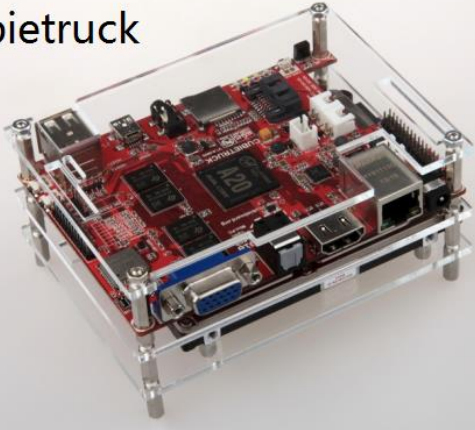
cubietruck



Quelle: cubieboard.org

μ Linux: CubieTruck

cubietruck



- **ARM Cortex A7 Dual @ 1 GHz**
- **GPU: OpenGL ES2.0, OpenVG**
- **Linux: Ubuntu**
- **2 GB RAM, 8 GB Flash,**
- **SATA, Ethernet (1000), WiFi,**
VGA, HDMI, Audio, SPDIF
- **~7.5 Watt @ 5V**
- **Preis: ~90 €**

<http://cubieboard.org/model/>

μLinux: Weitere...

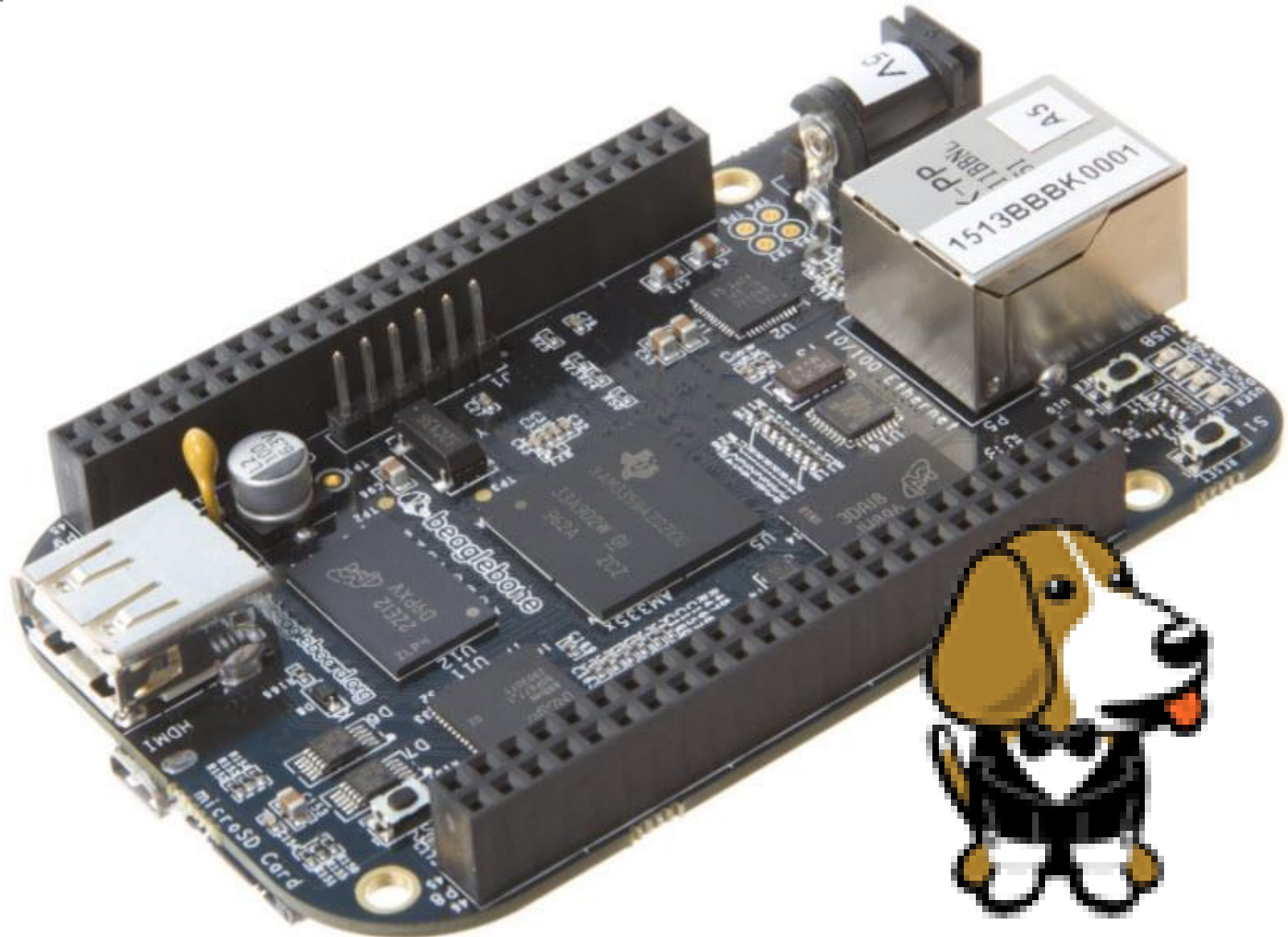


Quelle: mbed.org

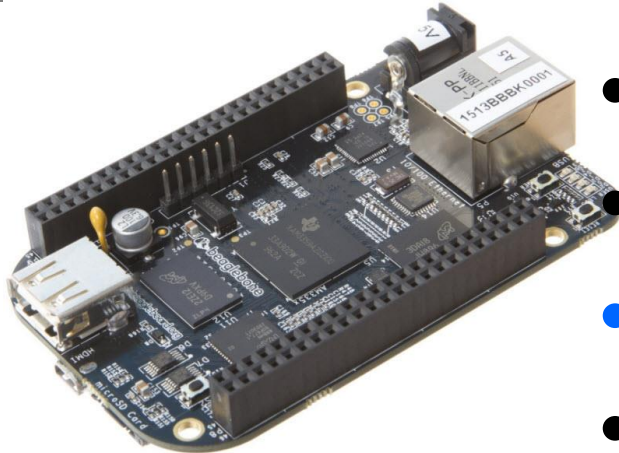


Linux-Boards für Steuerungen

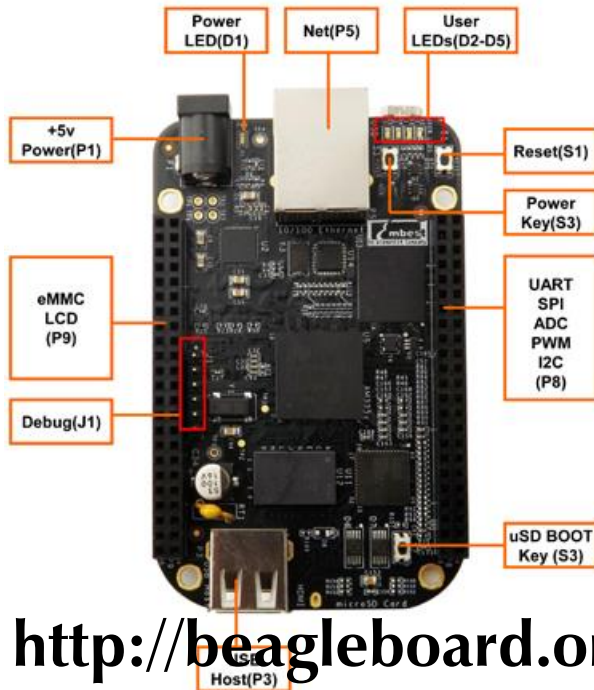
μ Linux: BeagleBone Black (Rev C)



μLinux: BeagleBone Black (Rev C)

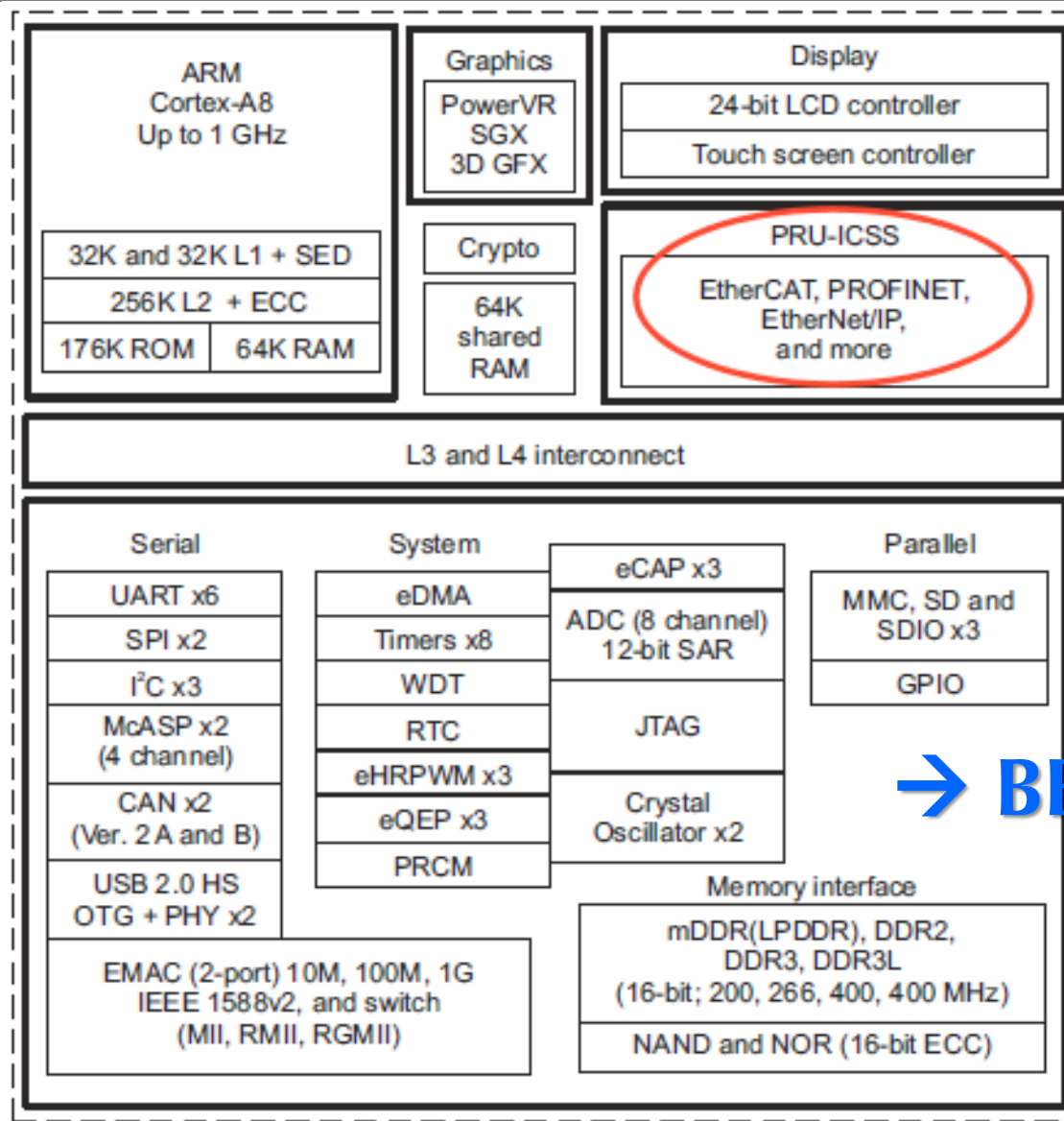


- ARM Cortex A8 @ 1 GHz
- Linux: ~~Angstrom~~, Debian
- 1 GB RAM, 2 4 GB Flash
- GPIO, UART, SPI, I2C, ADC
USB, Ethernet (100),
HDMI (nur Framer)
- PRU (Prog. Realtime Unit) @ 200 MHz
- 2.5 Watt @ 5V
- Preis: ~60 €



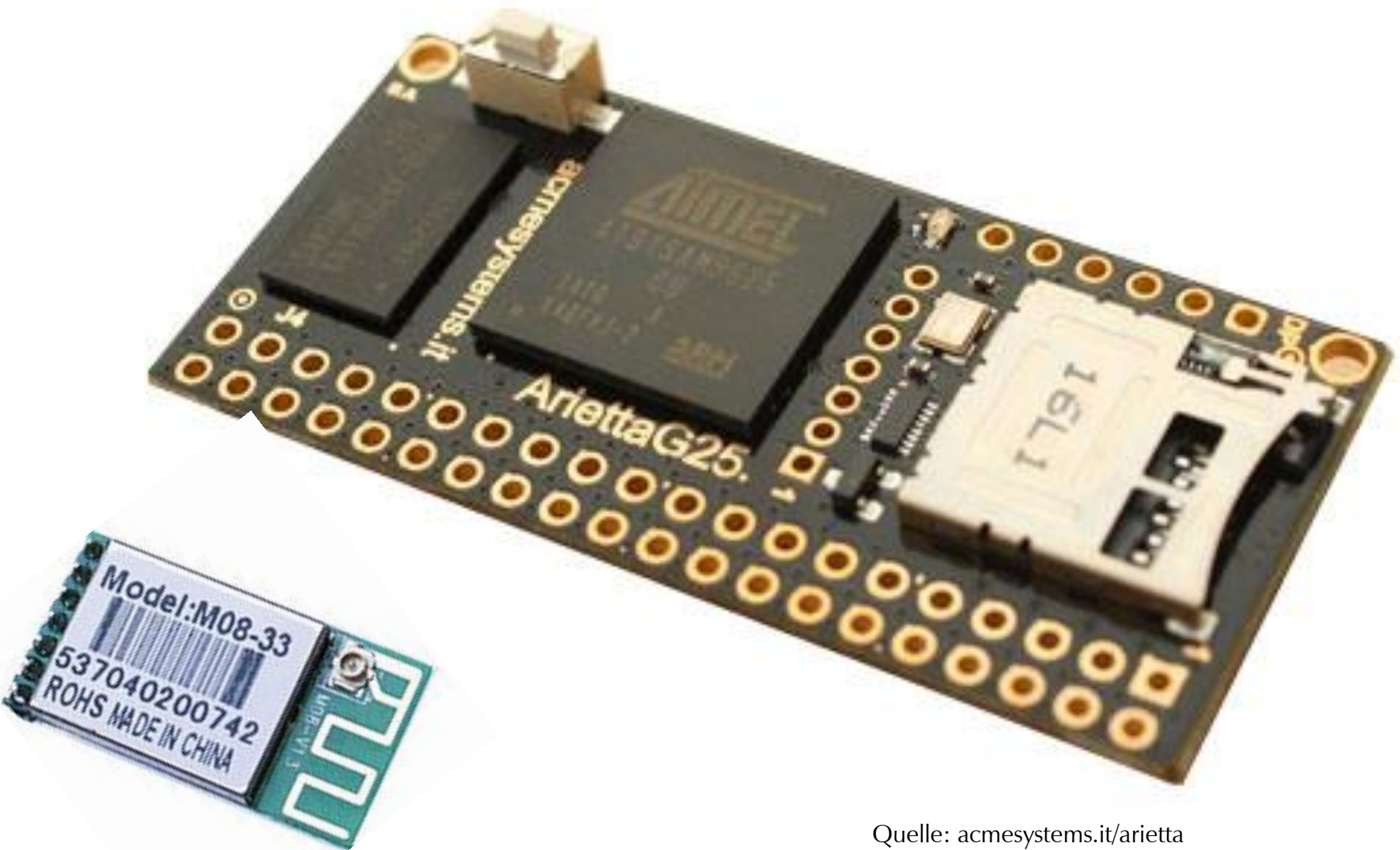
<http://beagleboard.org/black>

μLinux: Cortex A8 PRU



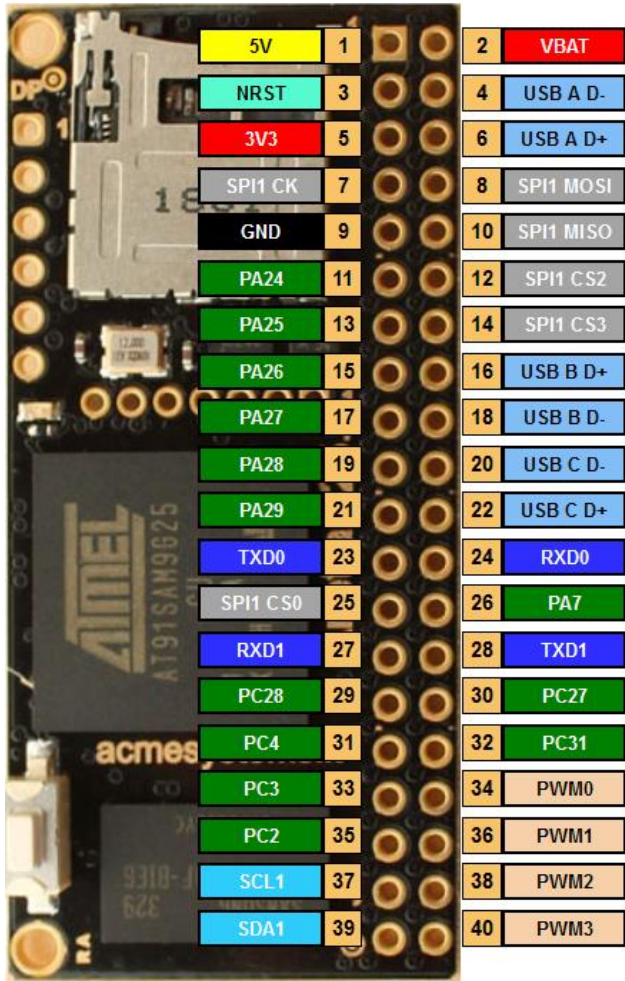
→ **BBB LinuxCNC**

μLinux: Arietta G25



Quelle: acmesystems.it/arietta

μLinux: Arietta G25



- ARM 9 @ 400 MHz
- Linux 3.16.1
- EmDebian Wheezy 7.6
- 128 / **256** MB RAM, microSD
- UART, SPI, I2C, I2S, PWM, 3x USB, ADC, (opt: WiFi)
- 0.3 ... 0.8 Watt @ 5V
- Preis: 22 / **35** €
+ 11 € WiFi-Modul



<http://www.acmesystems.it/arietta>

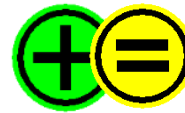


wann was wo?

wann was wo?

→ Lernen (Anfänger, englisch)

Programmieren, Elektronik, Schule,
Tutorials, Community, Bücher



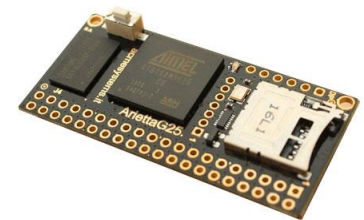
Arduino-Boards



ARM-Boards



Linux
für Multimedia

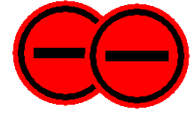


Linux
für Steuerungen

wann was wo?

→ Lernen (Anfänger, **deutsch**)

Programmieren, Elektronik, Schule,
Tutorials, Community, Bücher



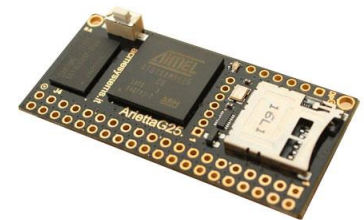
Arduino-Boards



ARM-Boards



Linux
für Multimedia



Linux
für Steuerungen

wann was wo?

→ Steuern/Regeln (langsam)

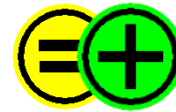
Lichtsteuerung, Hausautomatisierung,
Zugangskontrolle



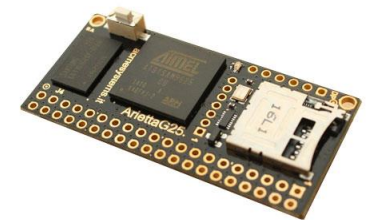
Arduino-Boards



ARM-Boards



Linux
für Multimedia

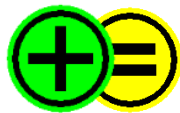


Linux
für Steuerungen

wann was wo?

→ Steuern/Regeln (**zeitkritisch**)

Motoren (DC, BLDC, Stepper), Roboter,
Servo, Fließkomma-Berechnungen



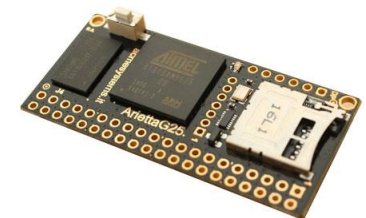
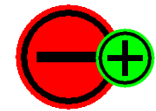
Arduino-Boards



ARM-Boards



Linux
für Multimedia



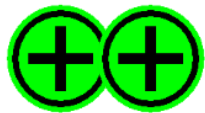
Linux
für Steuerungen

BBB LinuxCNC

wann was wo?

→ Sensoren

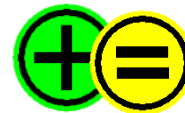
Temperatur, Energie, Beschleunigung, Schalter
über 1W, I2C, SPI, PWM



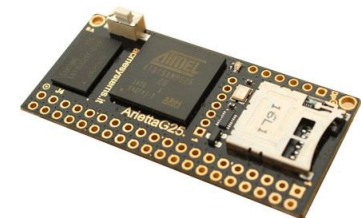
Arduino-Boards



ARM-Boards



Linux
für Multimedia



Linux
für Steuerungen

wann was wo?

→ 3D-Drucker

RAMPS, RADDs, Sanguinololu, Megatronics, ...



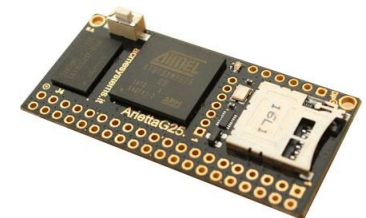
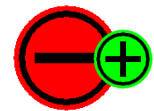
Arduino-Boards



ARM-Boards



Linux
für Multimedia



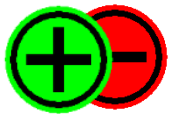
Linux
für Steuerungen

BBB LinuxCNC

wann was wo?

→ Internet of Things (IOT)

Sensoren, LAN/WLAN, DB



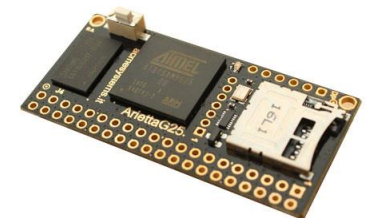
Arduino-Boards



ARM-Boards



Linux
für Multimedia



Linux
für Steuerungen

wann was wo?

→ Data Logger

SD-Card, SQL-DB



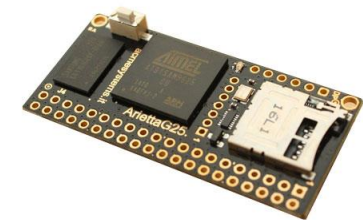
Arduino-Boards



ARM-Boards



Linux
für Multimedia



Linux
für Steuerungen

wann was wo?

→ Bedienoberfläche

GUI, Web-Server, HW-Display, SSH



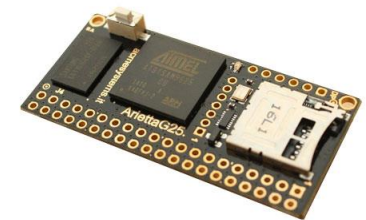
Arduino-Boards



ARM-Boards



Linux
für Multimedia



Linux
für Steuerungen

wann was wo?

→ Media-Center / PC-Ersatz

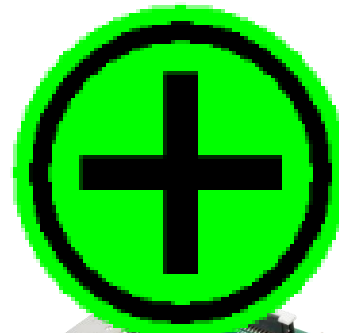
Wohnzimmer-“PC“



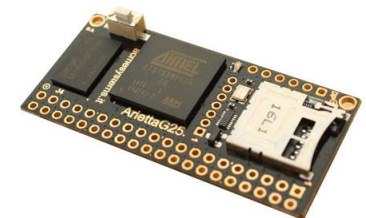
Arduino-Boards



ARM-Boards



Linux
für Multimedia



Linux
für Steuerungen



**Programme für
eigene Hardware
an Linux-Boards?!?**

LxHW → Problem: Device Tree

```
/dts-v1/;
/plugin/;
/ {
    compatible = "ti,beaglebone", "ti,beaglebone-black";

    part-number = "BB-W1";
    version = "00A0";

    /* state the resources this
    exclusive-use =
        /* the pin head
        "P9.22",
        /* the hardware
        "gpio0_2";

    fragment@0 {
        target = &am33xx_pinmux;
        __overlay__ {
            dallas_w1_pins: pinmux_dallas_w1_pins {
                pinctrl-single,pins = <0x150 0x37>;
            };
        };

    fragment@1 {
        target = &ocp;
        __overlay__ {
            onewire@0 {
                compatible = "w1-gpio";
                pinctrl-names = "default";
                pinctrl-0 = <&dallas_w1_pins>;
                status = "okay";
                gpios = <&gpio1 2 0>;
            };
        };
    };
};
```

```
dtc -O dtb -o BB-W1-00A0.dtbo -b 0 -@ BB-W1-00A0.dts
cp BB-W1-00A0.dtbo /lib/firmware/
echo BB-W1:00A0 > /sys/devices/bone_capemgr.9/slots
echo /sys/bus/w1/devices/10-000802ad4fae/w1_slave ...
```



LxHW → Lösung: Python

Programmiersprache

Python

LxHW → Python ...

... auf Raspberry Pi

```
import RPi.GPIO as GPIO
import time

GPIO.setup(11, GPIO.OUT)

while True:
    GPIO.output(11, GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output(11, GPIO.LOW)
    time.sleep(0.5)
```

<http://www.raspberrypi.org/documentation/usage/gpio/>



LxHW → Python ...

... auf BeagleBone

```
import Adafruit_BBIO.GPIO as GPIO
import time
```

```
GPIO.setup("P8_10", GPIO.OUT)
```

```
while True:
```

```
    GPIO.output("P8_10", GPIO.HIGH)
```

```
    time.sleep(0.5)
```

```
    GPIO.output("P8_10", GPIO.LOW)
```

```
    time.sleep(0.5)
```

<https://learn.adafruit.com/downloads/pdf/bleeping-an-led-with-beaglebone-black.pdf>



LxHW → Python ...

... auf Arietta G25

```
from ablib import Pin
from time import sleep

led = Pin('W9', 'OUTPUT')

while True:
    led.on()
    sleep(0.5)
    led.off()
    sleep(0.5)
```

<http://www.acmesystems.it/gpio>



Alle Logos, Marken und Namen sind Eigentum der jeweiligen Firmen