Fruit Pi Raspberry Pi versus Banana Pi versus Orange Pi

Autor:

Martin Strohmayer freeebook@strohmayers.com http://strohmayers.com

Fotografen:

Andreas Eberhard, Manfred Wallner

Version 3.0 5. Oktober 2017 Free PDF Edition

Inhaltsverzeichnis

L	Eck	daten und Kaufempfehlungen	11
	1.1	Raspberry Pi Modell B	11
	1.2	Raspberry Pi Modell B $+$	11
	1.3	Raspberry Pi 2 Modell B	13
	1.4	Raspberry Pi 3 Modell B	14
	1.5	Raspberry Pi Zero (W)	15
	1.6	Allgemeines Raspberry Pi	17
	1.7	Banana Pi (BPi-M1)	17
	1.8	Banana Pro	18
	1.9	Allgemeines Banana Pi/Pro	19
	1.10	Orange Pi PC	19
		Orange Pi Lite	20
		Orange Pi Zero 256/512	21
		Allgemeines Orange Pi	22
		Kaufempfehlungen	22
2	Betı	${f riebssystem}$	25
	2.1	Raspberry Pi	25
		2.1.1 Linux	25
		2.1.2 HTPC	25
		2.1.3 Spielkonsole	25
	2.2	Banana Pi (BPi-M1)	25
		2.2.1 Linux	25
	2.3	Orange Pi	26
		2.3.1 Linux	26
		2.3.2 HTPC	26
		2.3.3 Spielkonsole	26
3		callation Linux	27
	3.1	Raspberry Pi	27
		3.1.1 Installation auf MicroSD-Karte	27
		3.1.2 Einstellungen	28
		3.1.3 Aktivierung I2C- und SPI-Bus	30
		3.1.4 WiringPi C-Bibliothek	31
	3.2	Banana Pi	33
		3.2.1 Installation auf SD-Karte	33
		3.2.2 Einstellungen	34

		3.2.3	Grüne Status LED	3
		3.2.4	Aktivierung I2C- und SPI-Bus (Kernel 3.4.x)	3
		3.2.5	Kernel 4.4.x (Mainline)	3
		3.2.6	Aktivierung I2C-Bus	7
		3.2.7	Zurückschalten auf Kernel 3.4	7
		3.2.8	WiringPi C-Bibliothek	3
	3.3	Orange	Pi PC/Lite/Zero)
		3.3.1	Installation MicroSD-Karte)
		3.3.2	Einstellungen)
		3.3.3	WiringPi C-Bibliothek	2
		3.3.4	Einrichtung WLAN (Orange Pi Lite/Zero)	2
		3.3.5	Ressource Monitor	3
		3.3.6	Kühlung	3
		3.3.7	$\ddot{\mathrm{U}}\mathrm{bertakten}$	5
4			et / OTG Modus	
	4.1		erbrauch/Optimierungen	
		4.1.1	Raspberry Pi Zero	
		4.1.2	Orange Pi Lite	
		4.1.3	Orange Pi Zero	
		4.1.4	Vergleich	
	4.2		s Gerät (Serial mode)	
		4.2.1	Raspberry Pi Zero - Client	
		4.2.2	Orange Pi Lite/Zero - Client	
		4.2.3	Windows - Host	
	4.3		speichergerät (Mass storage mode)	
		4.3.1	Partition / RAM-Disk	
		4.3.2	Datei	
	4.4		etgerät (Ethernet mode)	
		4.4.1	Raspberry Pi Zero - Client	
		4.4.2	Windows 10 - Host	
		4.4.3	Windows 7 - Host	
		4.4.4	Linux (Ubuntu 16.04) - Host	2
5	GPI	ω	65	<
J	5.1		rry Pi	
	5.2	-	$_{0}$ Pi/Pro	
	$5.2 \\ 5.3$	Orange	Pi PC	
	0.0	Orange	1110	,
6	Peri	forman	ce 73	3
	6.1	CPU .		3
		6.1.1	Installation	3
		6.1.2	Ausführung	3
		6.1.3	Berechnung	1
		6.1.4	Ergebnisse	1
	6.2		rk/USB-HDD	5
		6.2.1	Installation	5
		6.2.2	Ausführung	
		6.2.3	Ergebnisse Netzwerk	

In halts verzeichn is

	6.2.4	Ergebnisse USB-HDD	77
	6.2.5	Ergebnisse SATA- und USB-SSD mit Banana Pi	79
6.3	I/O (V	WiringPi C-Bibliothek)	80
	6.3.1	GPIO	80
	6.3.2	SPI	81
	6.3.3	I2C	83

Vorwort CC



Dieses Werk steht unter der Lizenz Creative Commons BY-NC-ND 3.0

(https://creativecommons.org/licenses/by-nc-nd/3.0/at/). Sie erlaubt ausdrücklich, das Werk zu vervielfältigen, zu verbreiten und öffentlich zugänglich machen. Es muss der Urheber genannt werden, die Weitergabe darf nicht kommerziell erfolgen.

Es ist nicht erlaubt diese Werk zu verändern und darauf aufbauen zu erweitern.

Dieses Werk ist eine reduzierte kostenfreie Version des kommerziell erhältlichen E-Book's. Es kann über

Amazon https://www.amazon.de/dp/B01LY28XVD/ und

 $Google\ \ https://play.google.com/store/books/details?id=kVEyDQAAQBAJ\ bezogen\ werden.$

Wenn Sie die Arbeit des Autors unterstützen wollen oder die orignale Version (qualitative bessere Bilder und vollständiger Umfang) besitzen wollen so erwerben Sie das E-Book, Danke!

Vorwort

Einplatinencomputer sind seit der Einführung des Raspberry Pi's 2012 beliebte Bastelobjekte und Basis für viele Projekte. Die Einsatzmöglichkeiten gehen von günstige HTPC Anwendungen bis zur Steuer- und Messtechnik. Auch der Einsatz als Home-Server für vielerlei Dienste ist ein oft genutztes Anwendungsgebiet.

Mittlerweile erlebt der Markt einen regelrechten Boom. Es gibt unzählige Konkurrenzprodukte zum etablierten Raspberry Pi der Raspberry Pi Foundation. Diese versuchen den Marktführer in Sachen Performance, Ausstattung und Preis herauszufordern. Dabei wird aber auch immer auf GPIO-Kompatibilität verwiesen, was nicht immer vollständig gegeben ist!

Der Banana Pi der chinesischen Firma Shenzhen LeMaker Technology Co., Limited ist einer der Konkurrenten, der bereits früh versucht hat mit eigenen Produkten zu punkten. Der Community Support kann zwar bei Weitem nicht an den des Raspberry Pi's heranreichen, ist aber für eine chinesische Firma recht gut.

Neu am Markt ist der Orange Pi vom chinesischen Anbieter Shenzhen Xunlong Software CO., Limited. Es gibt ihn in den unterschiedlichsten Ausstattungen und Preisen. Günstig beziehen kann man ihn allerdings nur über Aliexpress direkt aus China. Dafür ist der Preis Vergleichsweise niedrig (besonders bei den Einstiegsmodellen Zero, One, Lite und PC). Leider ist der Support vom Hersteller sehr bescheiden und so muss die Community einspringen (z. B. Personen rund um die Distribution Armbian).

Die Produkte Raspberry Pi, Banana Pi/Pro und Orange Pi werde in weiterer Folge intensiv beleuchtet und verglichen. Die Eckdaten und Bezugsquellen, die Installation, GPIO's und die Performance stehen dabei im Mittelpunkt. Bei der Performance wird nicht nur Wert auf die CPU- bzw. Core-Leistung gelegt, sondern auch auf die Transfergeschwindigkeiten der Schnittstellen und die GPIO-Geschwindigkeiten mit der WiringPi C-Bibliothek.

Raspberry PiTM ist ein eingetragenes Warenzeichen der Raspberry Pi Foundation. Banana PiTM und Banana ProTM sind eingetragene Warenzeichen von Shenzhen LeMeiKe Science & Technology CO., Limited.

Orange PiTM ist ein eingetragenes Warenzeichen von Shenzhen Xunlong Software CO., Limited.

Kapitel 1

Eckdaten und Kaufempfehlungen

1.1 Raspberry Pi Modell B

CPU Broadcom BCM2835 SoC mit 1 × 700 MHz ARM1176JZF-S CPU (ARMv6 Architektur)

Video Broadcom VideoCore IV (Dual Core, 128 KB L2-Cache, 250 MHz) OpenGL ES 2.0, OpenVG 1.1

Speicher 256 MB oder 512MB (seit Oktober 2012 bei Model B)

USB $2 \times \text{USB-Hosts } 2.0 \text{ (SMSC LAN9512)}$

Netzwerk 10/100-MBit-Ethernet (SMSC LAN9512)

GPIO 26 Pin (UART, I2C, SPI, PWM, I2S ab Rev. 2)

Anschlüsse CSI-Kamera, DSI-Display

Video/Audio Anschlüsse HDMI, 3,5 mm Klinkenbuchse für Audio, Cinchbuchse für Composite-Video

Steckplatz SD (Push-Pull)

 ${\bf Schaltplan\ https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/README.md}$

Datenblatt CPU https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835

Der Raspberry Pi B ist inzwischen veraltet und wird nicht mehr zum Kauf angeboten.

1.2 Raspberry Pi Modell B+

CPU Broadcom BCM2835 SoC mit 1 × 700 MHz ARM1176JZF-S CPU (ARMv6 Architektur)

Video Broadcom VideoCore IV (Dual Core, 128 KB L2-Cache, 250 MHz) OpenGL ES 2.0, OpenVG 1.1

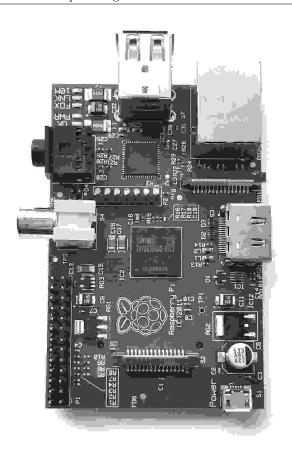


Abbildung 1.1: Raspberry Pi Modell B (Rev. 1)

Speicher 512 MB

 $\mathbf{USB}\ 4 \times \mathbf{USB\text{-}Hosts}\ 2.0\ (\mathrm{SMSC\ LAN9514})$

Netzwerk 10/100-MBit-Ethernet (SMSC LAN9514)

 $\mathbf{GPIO}\ 40\ \mathrm{Pin}\ (\mathrm{UART},\ \mathrm{I2C},\ \mathrm{SPI},\ \mathrm{PWM},\ \mathrm{I2S})$

Anschlüsse CSI-Kamera, DSI-Display

Steckplatz MicroSD (Push-Push)

Leerlauf Verbrauch 0,2 A

Preis 26-30 €

 ${\bf Shop\ http://geizhals.at/raspberry-pi-modell-b-a1240945.html}$

 ${\bf Schaltplan\ https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/README.md}$

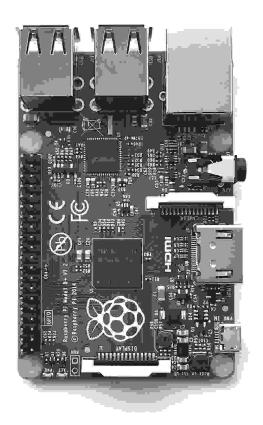


Abbildung 1.2: Raspberry Pi Modell B+

Datenblatt CPU https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835

1.3 Raspberry Pi 2 Modell B

CPU Broadcom BCM2836 SoC mit 4 × 900 MHz ARM Cortex-A7 CPU (ARMv7 Architektur)

Video Broadcom VideoCore IV (Dual Core, 128 KB L2-Cache, 250 MHz) OpenGL ES 2.0, OpenVG 1.1

Speicher 1024 MB

USB $4 \times \text{USB-Hosts } 2.0 \text{ (SMSC LAN9514)}$

Netzwerk 10/100-MBit-Ethernet (SMSC LAN9512)

GPIO 40 Pin (UART, I2C, SPI, PWM, I2S)

Anschlüsse CSI-Kamera, DSI-Display

Video/Audio Anschlüsse HDMI, Kombinierte 3,5 mm Klinkenbuchse für Audio und Composite-Video

Steckplatz MicroSD (Push-Push)

Leerlauf Verbrauch 0,22 A



Abbildung 1.3: Raspberry Pi 2 Modell B

Preis 33-35 €

Shop http://geizhals.at/raspberry-pi-2-modell-b-a1225369.html

Datenblatt CPU https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2836

1.4 Raspberry Pi 3 Modell B

 \mathbf{CPU} Broadcom BCM2837 SoC mit 4 × 1200 MHz ARM Cortex-A53 CPU (ARMv8 Architektur)

 $\bf Video$ Broadcom Video Core IV (Dual Core, 128 KB L2-Cache, 300/400 MHz) Open GL ES 2.0, Open VG 1.1

Speicher 1024 MB

USB $4 \times \text{USB-Hosts } 2.0 \text{ (SMSC LAN9514)}$

Netzwerk 10/100-MBit-Ethernet (SMSC LAN9512)

GPIO 40 Pin (UART, I2C, SPI, PWM, I2S)

Anschlüsse CSI-Kamera, DSI-Display

Steckplatz MicroSD (Push-Pull)

Zusätze WLAN (2,4 GHz), Bluetooth

Leerlauf Verbrauch 0,3 A

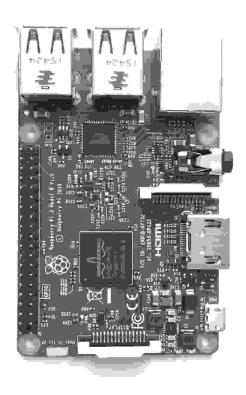


Abbildung 1.4: Raspberry Pi 3 Modell B

Preis 39-45 €

Shop http://geizhals.at/raspberry-pi-3-modell-b-a1400349.html

 ${\bf Schaltplan\ https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/README.md}$

1.5 Raspberry Pi Zero (W)

 ${f CPU}$ Broadcom BCM2835 SoC mit 1 × 1000 MHz ARM1176JZF-S CPU (ARMv6 Architektur)

Video Broadcom VideoCore IV (Dual Core, 128 KB L2-Cache, 250 MHz) OpenGL ES 2.0, OpenVG 1.1

Speicher 512 MB

USB $1 \times \text{Micro-USB-Hosts } 2.0$

GPIO 40 Pin (UART, I2C, SPI, PWM, I2S) ohne Pfostenstecker

Anschlüsse CSI-Kamera (ab Version 1.3)

Video/Audio Anschlüsse Mini-HDMI

Steckplatz MicroSD (Push-Pull)

Zusätze bei Zero W WLAN (2,4 GHz), Bluetooth

Leerlauf Verbrauch 0,1 A

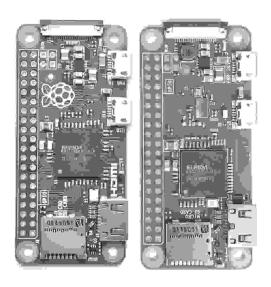


Abbildung 1.5: Raspberry Pi Zero und Zero W

Preis Zero 4,8 £ (ca. 5,5 €, excl. Versand)

Preis Zero W 9,6 £ (ca. 11 €, excl. Versand)

Shop UK http://thepihut.com/products/raspberry-pi-zero, https://shop.pimoroni.com/products/raspberry-pi-zero

 ${\bf Shop~DE~https://www.sertronics-shop.de/index.php?ID=213,~https://buyzero.de/index.php.ad/index.ph$

Shop NL https://www.kiwi-electronics.nl/raspberry-pi/raspberry-pi-zero

Datenblatt CPU https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835

Der Raspbery Pi Zero konnte Anfangs nur bei wenigen Internet Shops aus England oder USA bestellt werden. Seit Einführung der Raspberry Pi Zero W gibt es nun weit mehr Möglichkeiten den kleinen Einplatinencomputer zu bestellen (https://www.raspberrypi.org/products/ pi-zero). Es ist allerdings immer noch so, dass nur ein Stück pro Bestellung ausgeliefert werden darf. Für Österreich empfiehlt sich z. B. die Bestellung über Kiwi electronics in Holland.

1.6 Allgemeines Raspberry Pi

Alle Raspberry Pi Modelle müssen über den Micro-USB-Anschluss mit Energie versorgt werden. Je nach Modell werden unterschiedliche Angaben für den maximalen Strom gemacht. Für die älteren Modell B Modelle (Rev. 1 und Rev. 2) reichen 1,2 A aus. Für den Raspberry Pi B+ und 2 werden 1,8 A empfohlen. Für den Raspberry Pi 3 sollen es dann schon 2,5 A und 5,1 V sein. Schwächere Netzteile funktionieren zumeist auch, allerdings darf man dann keine leistungshungrigen USB-Geräte anschließen. Immer wieder kommt es aber zu Problemen, die auf ein schlechtes Netzteil zurückzuführen sind. Optimal sind deshalb die offiziellen Netzteile der Raspberry Pi Foundation. Leider werden sie allerdings zu recht unterschiedlichen Preisen angeboten, ein Preisvergleichsportal kann behilflich sein den besten Anbieter zu finden, http://geizhals.at/?cat=hwepcpsu&asuch=Raspberry. Mögliche Bezugsquellen sind auch RS Components und die englischen Anbieter der Raspberry Pi Zero PiHut und PiMoroni. Die älteren Versionen, Raspberry Pi Rev. 1 und Rev. 2 verwenden noch eine normale SD-Karte.

Die neueren Versionen verwenden eine MicroSD-Karte zur Speicherung des Betriebssystems.

Banana Pi (BPi-M1) 1.7

CPU Allwinner A20 SoC mit 2 × 1000 MHz ARM Cortex-A7 (ARMv7 Architektur)

Video Mali400MP2 (Dual Core, 256 KB L2-Cache, 500 MHz) OpenGL ES 2.0, OpenVG 1.1

Speicher 1 GB DDR3 SDRAM

USB $2 \times \text{USB-Hosts } 2.0, 1 \times \text{Micro-USB-OTG}$

Netzwerk 10/100/1000-MBit/s-Ethernet

GPIO 26 Pin (UART, I2C, SPI, CAN, PWM)

Anschlüsse CSI-Kamera, DSI-Display

Video/Audio Anschlüsse HDMI, 3,5 mm Klinkenbuchse für Audio, Cinchbuchse für Composite-Video

Steckplatz SD (Push-Pull)

Zusätze SATA-Port, Infrarot-Empfänger, Mikrofon

Leerlauf Verbrauch 0,23 A

Preis 34-36 €

Shop http://geizhals.at/banana-pi-m1-bpi-m1-a1108641.html

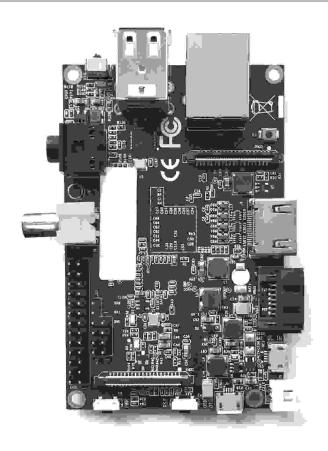


Abbildung 1.6: Banana Pi M1

Schaltplan http://linux-sunxi.org/images/9/9d/A20_Bananapi_Schematic.pdf

 ${\bf Handbuch\ http://www.ondaradio.es/portals/0/doc/Productos/44588/A_44588_141125225000483.}$ pdf

1.8 Banana Pro

 ${f CPU}$ Allwinner A20 SoC mit 2 × 1000 MHz ARM Cortex-A7 (ARMv7 Architektur)

Video Mali400MP2 (Dual Core, 256 KB L2-Cache, 500 MHz) OpenGL ES 2.0, OpenVG 1.1

Speicher 1 GB DDR3 SDRAM

USB $2 \times \text{USB-Hosts} \ 2.0, \ 1 \times \text{Micro-USB-OTG}$

 $\bf Netzwerk~10/100/1000\text{-}MBit/s\text{-}Ethernet$

GPIO 40 Pin (UART, I2C, SPI, CAN, PWM, SPDIF, I2S)

Anschlüsse CSI-Kamera, DSI-Display

Video/Audio Anschlüsse HDMI, Kombinierte 3,5 mm Klinkenbuchse für Audio und Composite-Video

Steckplatz MicroSD

Zusätze WLAN (2,4 GHz), SATA, Infrarot-Empfänger, Mikrofon

Preis 45-48 €

Shop http://geizhals.at/banana-pro-a1204547.html

Schaltplan http://linux-sunxi.org/images/3/3c/Banana_Pro_Schematic.pdf

Datenblatt CPU http://dl.linux-sunxi.org/A20/A20%20User%20Manual%20V1.20.pdf

1.9 Allgemeines Banana Pi/Pro

Banana Pi/Pro Modelle müssen über den Micro-USB-Anschluss mit Energie versorgt werden. Dazu sollte der Micro-USB-Stecker verwendet werden, der mit der Bezeichnung "DC-IN" versehen ist. Optimal ist ein Micro-USB Steckernetzteil mit 5 V und min. 2 A mit fixem Anschluss (z. B. https://www.reichelt.de/?ARTICLE=150733 oder http://at.rs-online.com/web/p/steckernetzteile/9098135/). Sollte eine 2,5" Festplatte angeschlossen sein, sollte das Steckernetzteil min. 2,5 A liefern können. 2,5 Zoll Festplatten können direkt vom System versorgt werden. Dazu wird allerdings ein spezielles Banana Pi SATA Kabel benötigt (https://geizhals.at/banana-pi-sata-kabel-a1201167.html).

Der Banana Pi/Pro ist der einzige Einplatinencomputer der SATA nativ über den SoC (Allwinner A20) unterstützt. Neuere Systeme (Orange Pi Plus, Banana Pi M3) verwenden einen "USB to SATA"-Bridge Chip (z. B. GL830). Das führt zu langsameren Übertragungsraten, da die USB-Bandbreite mit anderen Geräten (USB-HDD) geteilt werden muss.

Die Banana Pi verwendet eine SD-Karte, die Banana Pro eine MicroSD-Karte zur Speicherung des Betriebssystems.

1.10 Orange Pi PC

CPU Allwinner H3 SoC mit 4×1300 MHz ARM Cortex-A7 (ARMv7 Architektur)

Video Mali400MP2 (Dual Core, 256 KB L2-Cache, 600 MHz) OpenGL ES 2.0, OpenVG 1.1

Speicher 1 GB DDR3 SDRAM

USB $3 \times \text{USB-Hosts } 2.0, 1 \times \text{Micro-USB-OTG}$

Netzwerk 10/100-MBit/s-Ethernet

GPIO 40 Pin (UART, I2C, SPI, I2S)

Anschlüsse CSI-Kamera

Video/Audio Anschlüsse HDMI, Kombinierte 3,5 mm Klinkenbuchse für Audio und Composite-Video Steckplatz MicroSD (Push-Push)

Zusätze Infrarot-Empfänger, Mikrofon

Leerlauf Verbrauch 0,22 A

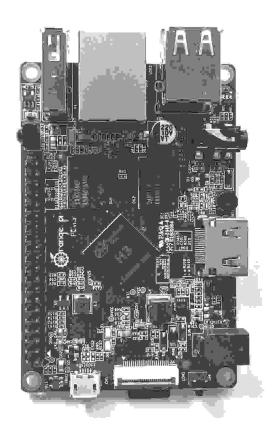


Abbildung 1.7: Orange Pi PC

Preis 14-15 €, 18-19 € (Set 3 inkl. Gehäuse und USB-Adapter), excl. Versand

Shop http://de.aliexpress.com/wholesale?SearchText=Orange+Pi+PC bzw.http://de.aliexpress.com/wholesale?SearchText=Orange+Pi+PC+set

Handbuch http://www.tato.ind.br/files/OrangePi_PC_user_manual_v0.9.1.pdf

Schaltplan https://drive.google.com/open?id=OBwQTOFP71kTbRFJQamNwWTAzWms

Datenblatt CPU http://linux-sunxi.org/File:Allwinner_H3_Datasheet_V1.2.pdf

1.11 Orange Pi Lite

CPU Allwinner H3 SoC mit 4 × 1200 MHz ARM Cortex-A7 (ARMv7 Architektur)

Video Mali400MP2 (Dual Core, 256 KB L2-Cache, 600 MHz) OpenGL ES 2.0, OpenVG 1.1

Speicher 512 MB DDR3 SDRAM

USB $2 \times \text{USB-Hosts } 2.0, 1 \times \text{Micro-USB-OTG}$

WLAN 2,4 GHz inkl. Antenne

GPIO 40 Pin (UART, I2C, SPI, I2S)

Anschlüsse CSI-Kamera

Video/Audio Anschlüsse HDMI, Kombinierte 3,5 mm Klinkenbuchse für Audio und Composite-Video

Steckplatz MicroSD (Push-Push)

Zusätze Infrarot-Empfänger, Mikrofon

Leerlauf Verbrauch 0,23 A

Preis 11-12 €, 20-21 € (Set 3 inkl. Gehäuse und USB-Adapter), excl. Versand

Shop http://de.aliexpress.com/wholesale?SearchText=Orange+Pi+Litebzw.http://de.aliexpress.com/wholesale?SearchText=Orange+Pi+Lite+set

Schaltplan https://linux-sunxi.org/images/8/88/Orange_pi-lite-v1_1.pdf

Datenblatt CPU http://linux-sunxi.org/File:Allwinner_H3_Datasheet_V1.2.pdf

1.12 Orange Pi Zero 256/512

CPU Allwinner H2+ SoC mit 4 × 1200 MHz ARM Cortex-A7 (ARMv7 Architektur)

Video Mali400MP2 (Dual Core, 256 KB L2-Cache, 600 MHz) OpenGL ES 2.0, OpenVG 1.1

Speicher 256 MB oder 512 MB DDR3 SDRAM

USB $1 \times \text{USB-Hosts} 2.0, 1 \times \text{Micro-USB-OTG}$

Netzwerk 10/100-MBit/s-Ethernet, inkl. POE

WLAN 2,4 GHz inkl. Antenne

 \mathbf{GPIO} 26 Pin (UART, I2C, SPI)

Anschlüsse 13 Pin Funktionserweiterung

Steckplatz MicroSD (Push-Push)

Leerlauf Verbrauch 0,17 A

Über Erweiterungkarte:

Video/Audio Anschlüsse Kombinierte 3,5 mm Klinkenbuchse für Audio und Composite-Video

USB $2 \times \text{USB-Hosts } 2.0$

Zusätze Infrarot-Empfänger, Mikrofon (über Erweiterungkarte)

Preis 256 MB 6-7 €, 10-11 € (Set 6 inkl. Gehäuse und Erweiterungkarte), excl. Versand

Preis 512 MB 8-9 €, 12-13 € (Set 6 inkl. Gehäuse und Erweiterungkarte), excl. Versand

Shop http://de.aliexpress.com/wholesale?SearchText=Orange+Pi+Zero bzw.http://de.aliexpress.com/wholesale?SearchText=Orange+Pi+Zero+set

1.13 Allgemeines Orange Pi

Nur die Orange Pi Zero Modelle können über den Micro-USB-Anschluss mit Energie versorgt werden. Alle anderen Orange Pi Modelle werden über einen Hohlstecker mit einem Steckernetzteil verbunden. Das Steckernetzteil muss 5 V und min. 2 A liefern können und kann auch bei AliExpress bezogen werden (http://de.aliexpress.com/wholesale?SearchText=charger+orange+pi). Das gelieferte Netzteil hat ein CE-Zeichen.

Optional kann aber auch ein Hohlstecker auf USB-Adapter gekauft werden (z. B. http://de.aliexpress.com/wholesale?SearchText=orange+pi+power+charger+cable). Dieser muss dann mit einem leistungsstarken Netzteil kombiniert werden (z. B. https://www.reichelt.de/?ARTICLE=176975). Es empfiehlt sich deshalb gleich bei AliExpress ein Set zu bestellen. Das Set kann Netzteil bzw. Adapter, Gehäuse und Kamera umfassen. Zu beachten ist, dass dieser Lieferant in China liegt und möglicherweise Einfuhrumsatzsteuer und Bearbeitungsgebühren der Post anfallen können. In Österreich sind alle Bestellungen unter 22 € gebührenbefreit (https://www.bmf.gv.at/zoll/post-internet/internet-shopping.html), diese Grenze sollte also beachtet werden.

Alle Orange Pi Versionen verwenden eine MicroSD-Karte zur Speicherung des Betriebssystems. Die Orange Pi Modelle mit Flash-Speicher (Modelle mit Plus im Namen) haben einen EMMC-Chip (Embedded Multimedia Card) auf der Platine, der für das Betriebssystem benutzt werden kann. Die folgende Tabelle enthält alle bei AliExpress verfügbaren Orange Pi Modelle. Der angegebene Preis ist der Verkaufspreis inkl. Versand im April/Mai 2017.

1.14 Kaufempfehlungen

Für Einsteiger empfiehlt es sich den Raspberry Pi 3 zu kaufen. Er hat ausreichend CPU-Leistung und Speicher um direkt darauf arbeiten zu können. Der Community Support, Bücher, Dokumentationen und Beispiele im Internet sind ausreichend vorhanden. Damit fällt der Einstieg am leichtesten.

Für dezidierte Projekte oder Einsatzgebiete kann ein Raspberry Pi B, B+ oder Zero (W) auch ausreichen sein. Als erfahrener Benutzer kann man sich auch an den limitierten Raspberry Pi Zero heranwagen. Der günstige Preis tauscht allerdings etwas, weil zum Betrieb möglicherweise Adapter benötigt werden und er nur in bestimmten Online-Anbietern bezogen werden kann.

Banana Pi und Banana Pro sind vor allem für den Einsatz als NAS bzw. wenn gute Netzwerkperformance benötigt wird, die erste Wahl. Der native SATA-Anschluss und Gigabit-Ethernet

Tabelle 1.1: Orange Pi Modelle

Name	Preis	CPU	RAM	Flash	USB	LAN	HDMI	WLAN	SATA
	€		GB	GB		$\mathrm{MBit/s}$			
Zero 256	10,50	H2+	0,25	-	1	100	-		-
Zero 512	11,73	H2+	0,5	-	1	100	-		-
One	12,66	Н3	0,5	-	1	100		-	-
Lite	14,52	Н3	0,5	-	2	-			-
PC	17,36	Н3	1	-	3	100		-	-
PC Plus	23,84	Н3	1	8	3	100			-
Zero Plus 2	20,98	Н3	0,5	8	-	-			-
Zero Plus 2	21,90	H5	0,5	8	-	-			-
PC 2	21,98	H5	1	-	3	1000		-	-
Win	26,97	A64	1	-	4	1000			-
Win Plus	32,15	A64	2	opt.	4	1000			-
Prime	32,15	H5	2	-	3	1000			-
Plus 2E	36,07	Н3	2	16	3	1000			-
Plus 2	48,74	Н3	2	16	4	1000			

sorgen für gute HDD- und Netzwerk-Performance. Mittlerweile wird das System bzw. der Allwinner A20 SoC auch vom Mainline Kernel unterstützt.

GPIO Geschwindigkeit ist keine Stärke der Plattform. Sie unterstützt allerdings als einziges den CAN-Bus. Sie bieten aber 2 CPU-Kerne weniger im Vergleich zu Raspberry Pi 2 und 3 und Orange Pi.

Der Orange Pi besticht durch den niedrigen Preis und der sehr guten Prozessorleistung (auch bei nur 1,2 bzw. 1,3 GHz). Er kann allerdings kostengünstig nur direkt aus China bezogen werden. Inzwischen ist er auch bei Amazon verfügbar allerdings zu einem wesentlich höheren Preis.

Aufgrund des fehlenden Supports vom Hersteller und der geringen Community empfiehlt er sich eher für erfahrene Benutzer. Bei der GPIO-Leistung sowie Ausstattung muss man gegenüber dem Raspberry Pi einige Abstriche machen. Es fehlt z. B. an PWM- und GCLK-Funktionen und der SPI-Bus hat nur einen CE-Ausgang. Die Performance der GPIOs liegt auch weit hinter dem Raspberry Pi. Als Ausgleicht hat er dafür einige UART-Schnittstellen an den GPIOs. Zurzeit gibt es noch keinen stabilen Mainline Linux Kernel 4.x für die verwendetet Prozessoren. Für die Orange Pi Modelle mit H5 und A64 Prozessoren gibt es zur Zeit noch kein stabiles Armbian Betriebssystem (nur experimentelle Nightly Builds).

der Orange Pi Zero 256/512 ist etwas spezieller, da kein HDMI Anschluss vorhanden ist. Bei allen Orange Pi Zero Modellen ist nur ein 26-Pin GPIO vorhanden. Die Belegung der GPIOs weichen von der anderen Orange Pi Modellen ab. Wie bei der Raspberry Pi Zero sind die GPIO-Kontakte nicht bestückt. Verschiedene Funktionen wie Audio, IR-Empfänger, Mikrofon und USB können über zusätzliche Kontakte erweitert werden. Sie sind bereits mit einem Pfostenstecker bestückt. Es gibt auch kostengünstige Aufsteckboards um diese Funktionen nutzen zu können.

Kapitel 2

Betriebssystem

2.1 Raspberry Pi

2.1.1 Linux

Betriebssystem Raspbian

Systembasis Debian 8 (Debian Jessie) ARM-hard-float-Architektur, Kernel 4.4.x

Download https://www.raspberrypi.org/downloads/raspbian/

2.1.2 HTPC

Betriebssystem LibreELEC, OSMC

Systembasis Linux

Download https://libreelec.tv/downloads/, https://osmc.tv/download/

2.1.3 Spielkonsole

 ${\bf Betriebs system} \ \ {\bf Retro Pie}, \ {\bf Recalbox}$

Systembasis Raspbian

 $\label{lock_com_retropie} \textbf{Download} \ \ \text{http://blog.petrockblock.com/retropie/retropie-downloads/, https://github.com/recalbox/recalbox-os/releases$

Eine Liste aller weiteren Betriebssysteme kann der Homepage des Herstellers entnommen werden, https://www.raspberrypi.org/downloads/.

2.2 Banana Pi (BPi-M1)

2.2.1 Linux

Betriebssystem Armbian

Systembasis Debian 8 (Debian Jessie) ARM-hard-float-Architektur, Kernel 3.4.112 (LTS) oder Kernel 4.6.3

Download http://www.armbian.com/download/

Betriebssystem Bananian

Systembasis Debian 8 (Debian Jessie) ARM-hard-float-Architektur, Kernel 3.4.111 (LTS) und Kernel 4.4.10 (LTS)

Download https://www.bananian.org/download

Eine Liste aller weiteren Betriebssysteme kann der Homepage des Herstellers entnommen werden, http://www.lemaker.org/product-bananapi-resource.html.

2.3 Orange Pi

2.3.1 Linux

Betriebssystem Armbian

Systembasis Debian 8 (Debian Jessie) ARM-hard-float-Architektur, Kernel 3.4.112 (LTS)

Download http://www.armbian.com/download/

2.3.2 HTPC

Betriebssystem OpenELEC

Systembasis Linux

Download http://down.nu/images/

2.3.3 Spielkonsole

Betriebssystem Lakka

Systembasis Linux

Download http://sources.lakka.tv/nightly/

Eine Liste aller weiteren Betriebssysteme kann der Homepage des Herstellers entnommen werden, http://www.orangepi.org/downloadresources/.

Kapitel 3

Installation Linux

3.1 Raspberry Pi

Raspbian ist das offizielle Betriebssystem für den Raspberry Pi. Mittlerweile kann man es in zwei unterschiedlichen Varianten herunterladen. Man kann zwischen einer minimalen Version (Lite) oder eine Version mit grafischer Oberfläche und vielen vorinstallierten Programmen wählen. Man benötigt in allen Fällen mindestens eine 4 GB große MicroSD-Karte. Man sollte aber, um Reserven zu haben, mindesten 8 GB verwenden.

Zur Installation benötigt man ein beliebiges Linux oder Windows System mit einem MicroSD-Kartenlesegerät. Die aktuelle Raspbian Version kann als Image-Datei von der Seite https://www.raspberrypi.org/downloads/ heruntergeladen werden. Aktuelle Versionen unterstützen alle Raspberry Pi Versionen.

In dieser Anleitung wird das minimale Raspbian Jessie Lite verwendet.

3.1.1 Installation auf MicroSD-Karte

Linux

```
apt-get install unzip
cd /tmp

wget --trust-server-names http://downloads.raspberrypi.org/raspbian_lite_latest
unzip 2016-11-25-raspbian-jessie-lite.zip
rm 2016-11-25-raspbian-jessie-lite.zip
```

Nun kann die enthaltene Image Datei mit dem Programm dd auf eine MicroSD-Karte übertragen werden.

Es ist unbedingt vor dem Ausführen des Befehls zu prüfen, ob das angegebene Laufwerk bzw. Device auch der vorgesehenen MicroSD-Karte entspricht!

dmesg | tail -n 10

```
sd 3:0:0:3: [sdf] Attached SCSI removable disk
sd 3:0:0:4: [sdg] Attached SCSI removable disk
sd 3:0:0:0: [sdc] 15448064 512-byte logical blocks: (7.90 GB/7.36 GiB)
sd 3:0:0:1: [sdd] Attached SCSI removable disk
sd 3:0:0:0: [sdc] Write Protect is off
sd 3:0:0:0: [sdc] Mode Sense: 03 00 00 00
sd 3:0:0:0: [sdc] No Caching mode page found
sd 3:0:0:0: [sdc] Assuming drive cache: write through
sdc: sdc1 sdc2
sd 3:0:0:0: [sdc] Attached SCSI removable disk
```

dd if=2016-11-25-raspbian-jessie-lite.img of=/dev/sdc bs=4M

```
1323+0 Datensätze ein
1323+0 Datensätze aus
1387266048 Bytes (1,4 GB) kopiert, 127,6147 s, 10,4 MB/s
```

Windows

Zuerst muss man die erhaltene Datei "2016-11-25-raspbian-jessie-lite.zip" entpacken. Die Image-Datei "2016-11-25-raspbian-jessie-lite.img" kann mit dem Programm "Win32 Disk Imager" (Download auf http://sourceforge.net/projects/win32diskimager) auf die MicroSD-Karte übertragen werden. Dazu wählt man die Datei aus und gibt das Laufwerk an, in dem sich die zu schreibende MicroSD-Karte befindet (Achtung: Daten von dem Laufwerk werden überschrieben!). Nach dem Drücken von "Write" und dem Bestätigen der Warnmeldung, dass alle Daten gelöscht werden, beginnt der Schreibvorgang.



3.1.2 Einstellungen

Nun muss noch eine leere Datei mit dem Namen "ssh" auf der MicroSD-Karte angelegt werden. Dies wird benötigt da sonst der SSH-Dienst nicht automatisch gestartet wird. Danach kann die MicroSD-Karte in die Raspberry Pi gesteckt werden und an die Versorgung angeschlossen werden.

Achtung: Beim Anmelden wird der Benutzer "pi" mit dem Passwort "raspberry" verwendet. Bei einer lokalen Anmeldung und der Verwendung einer deutschen Tastatur ist die Z und Y Taste vertauscht, weil der englische Tastaturtreiber aktiv ist. In dem Fall muss man als Passwort "raspberrz" eingeben!

Nach der Anmeldung kann das Raspberry Pi Software Configuration Tool (raspi-config) gestartet werden. Dazu verwendet man den Aufruf "sudo raspi-config".

Der Menüpunkt "Change User Password" ändert das Passwort für den Benutzer "pi".

<OK> drücken

Enter new UNIX password:

Retype new UNIX password:

<OK> drücken

"Boot Options" konfiguriert, was am Ende des Bootvorganges gestartet werden soll.

Nur Konsole: Console - Text console, requiring login (default)

Nur Konsole mit automatischer Anmeldung: Console Autologin - Text console, automatically logged in as 'pi' user

Grafische Oberfläche: Desktop - Desktop GUI, requiring user to login

Grafische Oberfläche mit automatischer Anmeldung: Desktop Autologin - Desktop GUI, automatically logged in as 'pi' user

Für den Serverbetrieb sollte "Console - Text console, requiring login (default)" ausgewählt werden.

Danach sollte man die Regionseinstellungen mit dem Menüpunkt "Internationalisation Options" einstellen.

Im folgenden Untermenü kann man mit "Change Locale" die Sprache und Zeichensatz des Systems setzen. Hier wählt man z. B. "de_AT.UTF-8 UTF8" für Österreich oder "de_DE.UTF-8 UTF8" für Deutschland. "en_GB.UTF-8 UTF8" und "C.UTF-8" sind bereits ausgewählt und können zusätzlich aktiv sein. Im nächsten Fenster kann man dann die Sprache "de_AT.UTF-8 UTF8" bzw. "de_DE.UTF-8 UTF8" als Standardeinstellung übernehmen.

Nun kann man mit "Internationalisation Options" und "Change Timezone" die aktuelle Zeitzone auswählen. Für das geografische Gebiet kann man "Europe" auswählen. Danach kann man als Zeitzone die Stadt "Vienna" oder "Berlin" auswählen.

Bei der dritten Einstellung "Internationalisation Options" und "Change Keyboard Layout" kann man das Tastatur-Layout auswählen:

Keyboard model: Generic 105-key (Intl) PC (Tastatur mit Windows Taste)

Keyboard layout: Other

Country of origin for the keyboard: German

Keyboard layout: German - German (eliminate dead keys) Key to function as AltGr: The default for the keyboard layout

Compose key: No compose key

Mit "Overclock" aktiviert man die automatische Übertaktung bei hoher CPU Last (nähere Informationen http://elinux.org/RPi_Overclocking).

Der Raspberry Pi B/B+ bietet Übertaktungseinstellungen von 800 bis 1000 MHz. Bei der höchsten Übertaktungsstufe "Turbo" wird der RAM sowie die GPU (Grafikkern) zusätzlich stark übertaktet. Nicht jeder Raspberry Pi B läuft mit dieser Stufe stabil (besonders ältere Rev. 1 Fabrikate). Beim Starten kann man die Shift-Taste gedrückt halten, dann wird die Übertaktung ausgeschaltet.

Bei dem Raspberry Pi B/B+ ist die Übertaktung von 800 MHz problemlos möglich und kann als moderaten Standard gewählt werden. Wer mehr Leistung benötigt, sollte den Modus "Turbo" versuchen.

<OK> drücken.

Modest 800MHz ARM, 250MHz core, 400MHz SDRAM, 0 overvolt

<OK> drücken.

Der aktivierte Modus "Modest" wird noch einmal ausgegeben.

<OK> drücken.

Der Raspberry Pi 2 läuft standardmäßig bereits mit 900 MHz, für ihn gibt es einen eigenen Eintrag mit dem Namen "High". Er aktiviert einen maximalen CPU-Takt von 1000 MHz, einen Core-Takt von 500 MHz und einen RAM-Takt von 500 MHz.

<OK> drücken.

High 1000MHz ARM, 500MHz core, 500MHz SDRAM, 2 overvolt

<OK> drücken.

Der aktivierte Modus "High" wird noch einmal ausgegeben.

<OK> drücken.

Der Raspberry Pi 3 und Zero können im Programm nicht übertaktet werden. Die Standardeinstellungen des Raspberry Pi Zero ist ähnlich der "Turbo" Einstellungen der Raspberry Pi B+. Der CPU-Takt ist bei beiden auf 1000 MHz eingestellt. Der Core-Takt liegt beim Raspberry Pi

Zero auf 300 MHz und der RAM-Takt auf 450 MHz.

Nun kann man "Advanced Options" ausführen, um weitere Einstellungen zu machen. Im folgenden Untermenü kann man mit "Memory Split" die Speicherzuordnung für den Grafikkern anpassen.

How much memory should the GPU have? e.g. 16/32/64/128/256

64

Für die Anwendung als Server ohne Bedarf an Spielen und Video kann 16 MB gewählt werden. Ansonsten ist die Standardeinstellung von 64 MB zu bevorzugen.

Mit dem Menüpunkt "Advanced Options" und "SSH" ändert man die Einstellung bzw. Funktion des SSH-Dienstes.

Would you like the SSH server enabled or disabled? <Enable>

<Enable> aktiviert den Dienst und <Disable> deaktiviert den Dienst.

"Advanced Options" und "Update" aktualisiert die apt-Paketliste und das Configuration Tool (raspi-config) selbst.

<Finished>

Danach befindet man sich in der Konsole.

Zum Schluss kann man dann das Betriebssystem noch aktualisieren und unnötig belegten Speicher auf der $\mathrm{SD/MicroSD\text{-}Karte}$ freigeben.

```
sudo sudo apt-update && sudo apt-get -y upgrade
sudo apt-get clean
sudo reboot
```

3.1.3 Aktivierung I2C- und SPI-Bus

Die Installation der Module für den I2C- und SPI-Bus kann entweder manuell erfolgen oder automatisch. Am einfachsten geht es, wenn man die Konfiguration über das Raspberry Pi Software Configuration Tool durchführt.

sudo raspi-config

```
8 Advanced Options A7 I2C Enable/Disable automatic loading Would you like the ARM I2C interface to be enabled? < Ja> The ARM I2C interface is enabled < Ok> Would you like the I2C kernel module to be loaded by default? < Ja> I2C kernel module will now be loaded by default < Ok>
```

```
8 Advanced Options A6 SPI Enable/Disable automatic loading Would you like the SPI interface to be enabled? < Ja> The SPI interface is enabled < Ok> Would you like the SPI kernel module to be loaded by default? < Ja> SPI kernel module will now be loaded by default < Ok> Finish> Would you like to reboot now? < Ja>
```

Durch die permanente Installation der I2C und SPI-Schnittstelle ("loaded by default") wird in die Datei "config.txt" der Eintrag "dtparam=i2c_arm=on" und "dtparam=spi=on" hinzugefügt. Dies ist Teil des neuen Systems "Device Tree", das nun Ressourcenzuordnung und das Laden von Kernel-Modulen übernimmt. Es soll in Zukunft die automatische Konfiguration von zusätzlicher Aufsteckhardware erleichtern. Mehr zu dem Thema erfährt man auf der Seite https://www.raspberrypi.org/documentation/configuration/device-tree.md. Der Eintrag führt dazu, dass das Kernel-Modul "i2c_bcm2708" und "spi_bcm2708" automatisch geladen wird. Optional kann noch die Zeile "dtparam=i2c1_baudrate=400000" hinzugefügt werden, um die Geschwindigkeit des I2C-Busses von 100 kHz auf 400 kHz zu erhöhen.

3.1.4 WiringPi C-Bibliothek

WiringPi ist eine C-Bibliothek für den Zugriff auf Funktionen der GPIO's des Prozessors. Sie wurde von Gordon Henderson für den Raspberry Pi programmiert. Sie wird unter der GNU LGPLv3 Lizenz angeboten.

```
sudo apt-get install wiringpi

Update:

sudo apt-get install git
cd /usr/src
sudo git clone git://git.drogon.net/wiringPi
cd wiringPi
sudo git pull origin
sudo ./build
```

Raspberry Pi B (Rev. 2)

gpio -v

gpio version: 2.32 Copyright (c) 2012-2015 Gordon Henderson This is free software with ABSOLUTELY NO WARRANTY. For details type: gpio $\mbox{-warranty}$

Raspberry Pi Details:

Type: Model B, Revision: 03, Memory: 512MB, Maker: Egoman * Device tree is enabled.

- * This Raspberry Pi supports user-level GPIO access.
 - -> See the man-page for more details -> ie. export WIRINGPI_GPIOMEM=1

gpio readall

+++Model B2+-++									
BCM wPi Name	Mode V Physi	.cal V Mode	Name	wPi	BCM				
++	+-+	+-+	+	+	++				
3.3v	1	2	5v						
2 8 SDA.1	IN 1 3	4	5V						
3 9 SCL.1	IN 1 5	6	0 v						
4 7 GPIO. 7	IN 1 7	8 1 ALT0	TxD	15	14				
0v	9	10 1 ALT0	RxD	16	15				
17 0 GPIO. 0	IN 0 11	12 0 IN	GPIO. 1	1	18				
27 2 GPIO. 2	IN 0 13	14	0v		1 1				
22 3 GPIO. 3	IN 0 15	16 0 IN	GPIO. 4	4	23				
3.3v	17	18 0 IN	GPIO. 5	5	24				
10 12 MOSI	IN 0 19	20	0v						
9 13 MISO	IN 0 21	22 0 IN	GPIO. 6	6	25				
11 14 SCLK	IN 0 23	24 1 IN	CEO	10	8				
0v	25	26 1 IN	CE1	11	7				
++	+	+-+	+	+	++				
28 17 GPIO.17	IN 0 51	52 0 IN	GPIO.18	18	29				
30 19 GPIO.19	IN 0 53	54 0 IN	GPI0.20	20	31				
++	+	+-+	+	+	++				
BCM wPi Name	Mode V Physi	.cal V Mode	Name	wPi	BCM				
++	+-+Model	B2+-+	+	+	++				

Raspberry Pi 2

gpio -v

gpio version: 2.32 Copyright (c) 2012-2015 Gordon Henderson This is free software with ABSOLUTELY NO WARRANTY. For details type: gpio -warranty

Raspberry Pi Details:

- Type: Pi 2, Revision: 01, Memory: 1024MB, Maker: Sony
- \star Device tree is enabled.
- * This Raspberry Pi supports user-level GPIO access.
 - -> See the man-page for more details -> ie. export WIRINGPI_GPIOMEM=1

++++									
BCM wPi Name			Name M	ode V Phy	sic	cal V Mod	de Name	wPi	L BCM
+	+-	+	+-	+-+	++-	+	+	+	-++
			3.3v	1	\Box	2	5v		1 1
	2	8	SDA.1	IN 1 3	\Box	4	5V		1 1
	3	91	SCL.1	IN 1 5	\Box	6	10v		1 1
- 1	4	7 0	GPIO. 7	IN 1 7	\Box	8 1 AL7	TXD OT	15	14
			0v	9	\Box	10 1 AL7	0 RxD	16	15
	17	0 0	GPIO. 0	IN 0 11	\Box	12 0 IN	GPIO.	1 1	18
	27	2 0	GPIO. 2	IN 0 13	\Box	14	10v		1 1
- 1	22	3 0	GPIO. 3	IN 0 15	\Box	16 0 IN	GPIO.	4 4	23
- 1			3.3v	17	\Box	18 0 IN	GPIO.	5 5	24
- 1	10	12	MOSI	IN 0 19	\Box	20	10v		1 1
- 1	91	13	MISO	IN 0 21	\Box	22 0 IN	GPIO.	6 6	25
	11	14	SCLK	IN 0 23	$ \cdot $	24 1 IN	CEO	10	8

```
| |25 || 26|1|IN
                                        ICE1
                                                |11 |7
              0v1
  0| 30| SDA.0|
                   IN|1|27 || 28|1|IN
                                        |SCL.0
                                                |31 |1
  5| 21|GPIO.21|
                   IN|1|29
                           11 301 1
                                        10v
                           || 32|0|IN
                                        |GPIO.26|26 |12
  61 22|GPI0.22|
                   IN|1|31
                   IN|0|33
 13| 23|GPI0.23|
                              341 1
                                        10v
 19| 24|GPIO.24|
                   IN|0|35
                           || 36|0|IN
                                        |GPI0.27|27 |16
 26| 25|GPIO.25|
                   IN|0|37
                              38|0|IN
                                        |GPI0.28|28
              0 v I
                     | |39 || 40|0|IN
                                        |GPI0.29|29 |21
|BCM|wPi| Name |Mode|V|Physical|V|Mode| Name |wPi|BCM|
                     -+-+--Pi 2--+
```

Beispiel Kompilierung mit WiringPi und Mathematik Bibliothek:

```
gcc -o spi_test spi_test.c -Wall -lwiringPi -lm
```

3.2 Banana Pi

3.2.1 Installation auf SD-Karte

Der Banana Pi unterstützt eine Vielzahl an verschiedenen Betriebssystemen. Für das Betriebssystem Bananian (es basiert auf Debian Jessie mit armhf Architektur) benötigt man eine mindestens 4 GB große SD-Karte. Zur Installation benötigt man ein beliebiges Linux oder Windows System mit einem SD-Kartenlesegerät.

Die aktuelle Image-Datei für die Installation kann man von der Seite https://www.bananian.org/download herunterladen.

Allerdings wird dieses Betriebssystem in Zukunft nicht weitergeführt (kein Support für Debian Stretch). Alternativ kann auch das Betriebssystem Armbian (http://www.armbian.com/download/) eingesetzt werden. Die Installation verläuft dann entsprechend der Anleitung für den Orange Pi.

Linux

```
cd /tmp
wget http://dl.bananian.org/releases/bananian-latest.zip
unzip bananian-latest.zip
rm bananian-latest.zip
```

Nun kann die enthaltene Image Datei mit dem Programm dd auf eine SD-Karte übertragen werden

Es ist unbedingt vor dem Ausführen des Befehls zu prüfen, ob das angegebene Laufwerk bzw. Device auch der vorgesehenen SD-Karte entspricht!

dmesg | tail -n 10

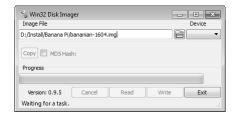
```
sd 4:0:0:4: [sdg] Attached SCSI removable disk sd 4:0:0:0: [sdc] 30392320 512-byte logical blocks: (15.5 GB/14.4 GiB) sd 4:0:0:1: [sdd] Attached SCSI removable disk sd 4:0:0:3: [sdf] Attached SCSI removable disk sd 4:0:0:0: [sdc] Write Protect is off sd 4:0:0:0: [sdc] Mode Sense: 03 00 00 00 sd 4:0:0:0: [sdc] No Caching mode page found sd 4:0:0:0: [sdc] Assuming drive cache: write through sdc: sdc1 sdc2 sd 4:0:0:0: [sdc] Attached SCSI removable disk
```

dd if=bananian-1604.img of=/dev/sdc bs=1M

```
1811+1 Datensätze ein
1811+1 Datensätze aus
1900000000 Bytes (1,9 GB) kopiert, 175,313 s, 10,3 MB/s
```

Windows

Zur Installation des Betriebssystem muss man zuerst die erhaltene Datei "bananian-latest.zip" entpacken. Die Image-Datei "bananian-1604.img" kann mit dem Programm "Win32 Disk Imager" (Download auf http://sourceforge.net/projects/win32diskimager) auf die SD-Karte übertragen werden. Dazu wählt man die Datei aus und gibt das Laufwerk an, in dem sich die zu schreibende SD-Karte befindet (Achtung: Daten von dem Laufwerk werden überschrieben!). Nach dem Drücken von "Write" und dem Bestätigen der Warnmeldung, dass alle Daten gelöscht werden, beginnt der Schreibvorgang.



Nun kann die SD-Karte eingesteckt und an die Versorgung angeschlossen werden. Nach dem Bootvorgang fängt die grüne LED zu blinken an. Dann kann über eine SSH-Verbindung die weitere Einrichtung erfolgen. Achtung: Für eine SSH-Verbindung unter Windows wird Putty ab Version 0.63 benötigt!

3.2.2 Einstellungen

Achtung: Beim Anmelden wird der Benutzer "root" mit dem Passwort "pi" verwendet. Nach der Anmeldung kann das Konfigurationsprogramm "bananian-config" gestartet werden.

bananian-config

```
Welcome to bananian-config!

This script assists you to set up some basic parameters...

For news and updates check: http://www.bananian.org

Your current keyboard layout is 'us'. Do you want to change it? (y/N)
```

Y

Keyboard layout: Other

Country of origin for the keyboard: German

Keyboard layout: German - German (eliminate dead keys)

Your current root password is still 'pi'. You have to change it now!

Configuring root password...

Enter new UNIX password: Retype new UNIX password: passwd: password updated successfully

Your current timezone is 'Etc/UTC'. Do you want to change it? (y/N)

```
Kapitel 3. Installation Linux
Y
Geographic area: Europe
Time zone: Berlin
Your current locale is 'en_US.UTF-8'. Do you want to change it? (y/N)
Y
\overline{\text{Loc}} ales to be generated:
[*] de DE.UTF-8 UTF-8
[*] en_US.UTF-8 UTF-8
Default locale for the system environment: de_DE.UTF-8 \,
Your current hostname is 'bananapi'. Do you want to change it? (y/N) \,
Y
Enter new hostname: BananaPi
Your current hardware configuration is: BananaPi
Do you want to change it? (y/N)
Ν
Do you want to expand the root file system (recommended)? (y/N) \,
Y
```

```
Expanding root file system... (ignore the warnings and reboot immediately)
done! please reboot your system now! (shutdown -r now)
bananian-config 25.69s user 2.99s system 5% cpu 8:37.42 total
```

reboot

bananian-update

```
You are already using Bananian 16.04 r01
No upgrades available...
```

Zum Schluss kann man dann das Betriebssystem noch aktualisieren und unnötig belegten Speicher auf der SD-Karte freigeben.

```
apt-get update && apt-get -y upgrade
apt-get dist-upgrade
apt-get clean
reboot
```

uname -a

Linux BananaPi 3.4.113-bananian #8 SMP PREEMPT Sat Nov 26 00:48:28 UTC 2016 armv71 GNU/Linux

3.2.3 Grüne Status LED

Die Funktion der grünen LED auf dem Platine kann konfiguriert werden. Standard ist "heartbeat" also eine Blinkfunktion. Man kann aber auch z. B. mit mmc0 den SD-Kartenzugriff darstellen.

cat /sys/class/leds/green:ph24:led1/trigger

```
none battery-charging-or-full battery-charging battery-full battery-charging-blink-full-solid ac-online usb-online mmc0 timer [heartbeat] backlight gpio cpu0 cpu1 default-on
```

```
echo mmc0 > /sys/class/leds/green:ph24:led1/trigger
```

Um die sehr helle grün blinkende LED permanent zu deaktivieren, muss man die Funktion auf "none" schalten, indem man folgenden Befehl ausführt.

```
echo none > /sys/class/leds/green:ph24:led1/trigger
```

Dieser Befehl kann auch automatisch nach dem Bootvorgang über die Datei rc.local gestartet werden.

nano /etc/rc.local

/etc/init.d/rc.local

```
# Green LED off

#echo none > /sys/class/leds/green:ph24:led1/trigger

# Green LED SD-Card access
echo mmc0 > /sys/class/leds/green:ph24:led1/trigger

exit 0
```

3.2.4 Aktivierung I2C- und SPI-Bus (Kernel 3.4.x)

Der I2C-Bus Treiber ist bereits im Kernel enthalten und muss nicht aktiviert werden. Für den SPI-Bus muss das Kernelmodul "spi-sun7i" geladen werden.

```
modprobe spi-sun7i
echo "#Enable SPI-Bus\nspi-sun7i" >> /etc/modules
ls /dev/spi*
/dev/spidev0.0 /dev/spidev0.1
apt-get install i2c-tools
i2cdetect -1
        i2c
                  sunxi-i2c.0
                                        I2C adapter
i2c-1
        i2c
                  sunxi-i2c.1
                                       I2C adapter
I2C adapter
i2c-2
                  sunxi-i2c.2
       i2c
i2c-3
        i2c
                  sunxi-i2c.3
                                        I2C adapter
i2c-4
                  sunxi-hdmi-i2c
                                        I2C adapter
```

3.2.5 Kernel 4.4.x (Mainline)

Optional kann bei Bananian auch der aktuelle 4.4 Kernel verwendet werden.

```
apt-get install linux-image-4.4-bananian
reboot
uname -a
Linux bananapi 4.4.14-bananian #2 SMP Sat Jul 9 17:37:43 UTC 2016 armv7l GNU/Linux
```

3.2.6 Aktivierung I2C-Bus

Auch beim Kernel 4.4 ist der I2C-Bus Treiber bereits im Kernel enthalten und muss auch nicht aktiviert werden.

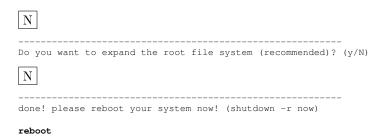
apt-get install i2c-tools i2cdetect -1

Der I2C-Bus mit Nummer 1 ist am 26-Pin Port verfügbar.

3.2.7 Zurückschalten auf Kernel 3.4

Wenn man wieder den alten 3.4 Kernel aktivieren möchte, kann dies über das Konfigurationsprogramm "bananian-config" erfolgen.

bananian-config Welcome to bananian-config! This script assists you to set up some basic parameters... For news and updates check: http://www.bananian.orgYour current keyboard layout is 'us'. Do you want to change it? (y/N) N Do you want to change your root password? (y/N) $\,$ Ν Your current timezone is 'Europe/Vienna'. Do you want to change it? (y/N) N Your current locale is 'de_DE.UTF-8'. Do you want to change it? (y/N) N Your current hostname is 'BananaPi'. Do you want to change it? (y/N)Ν Mainline Kernel (4.x) is currently enabled. Do you want to disable it? (y/N)Y Mainline Kernel disabled... Do you also want to remove all mainline Kernel packages? (Y/n)Y Your current hardware configuration is: BananaPi Do you want to change it? (y/N)



3.2.8 WiringPi C-Bibliothek

WiringPi ist eine C-Bibliothek für den Zugriff auf Funktionen der GPIO's des Prozessors. Sie wurde von Gordon Henderson für den Raspberry Pi programmiert. Sie wird unter der GNU LGPLv3 Lizenz angeboten.

Diese Bibliothek wurde für verschiedene Einplatinencomputer wie den Banana Pi angepasst und veröffentlicht. Die portierte C-Bibliothek trägt den Projektnamen WiringBP. Leider wird sie aber nicht ständig aktualisiert (ältere Version gegenüber Raspberry Pi).

```
apt-get install git build-essential sudo
cd /usr/src
git clone https://github.com/LeMaker/WiringBP.git -b bananapi
cd WiringBP
chmod +x ./build
sudo ./build

gpio -v

gpio version: 2.14
Copyright (c) 2012-2014 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty
This Raspberry Pi is a revision 3 board.
```

gpio readall

+-		+-	-Rev3	+-		+-		+-		+-	
	wiringPi		GPIO		Phys	1	Name		Mode		Value
Ī	0	l	17		11	Ī	GPIO 0	Ī	ALT4		Low
1	1	ı	18		12	1	GPIO 1		IN		Low
1	2	ı	27		13	1	GPIO 2		ALT4		Low
	3	I	22		15	1	GPIO 3		ALT4		Low
	4	l	23		16		GPIO 4		IN		Low
	5	ı	24		18	I	GPIO 5		IN		Low
	6	ı	25		22	1	GPIO 6		ALT4		Low
1	7	ı	4		7	1	GPIO 7		IN		Low
	8	I	2		3	1	SDA		ALT5		Low
1	9	ı	3		5	1	SCL		ALT5		Low
1	10	ı	8		24	1	CE0		IN		Low
	11	I	7		26	1	CE1		IN		Low
	12	l	10		19		MOSI		IN		Low
1	13	ı	9		21	1	MISO		IN		Low
1	14	ı	11		23	1	SCLK		IN		Low
1	15	ı	14		8	1	TxD		ALT0		Low
	16	I	15		10	1	RxD		ALT0		Low
1	17	ı	28		3	I	GPIO 8		IN		Low
1	18	ı	29		4	I	GPIO 9		ALT4		Low
1	19	ı	30		5	I	GPIO10		OUT		High
1	20	I	31		6		GPIO11		ALT4		Low
+-		+.		+-		+-		+-		+-	

Beispiel Kompilierung mit WiringPi und Mathematik Bibliothek:

```
gcc -o spi_test spi_test.c -Wall -lwiringPi -lm
```

3.3 Orange Pi PC/Lite/Zero

3.3.1 Installation MicroSD-Karte

Der Orange Pi unterstützt verschiedene Betriebssysteme. Zur Installation benötigt man ein beliebiges Linux oder Windows System mit einem MicroSD-Kartenlesegerät.

Die aktuelle Image-Datei für die Installation von Armbian, kann man von der Seite http://www.armbian.com/download/ heruntergeladen werden. Hier werden für alle Orange Pi Varianten eigene Images angeboten.

In dieser Anleitung wird das minimale Debian "Jessie server" Legacy Image (Kernel 3.4.112) verwendet. Das experiementelle Image mit Mainline Kernel wird aufgrund fehlenden Treiber nicht empfohlen.

Bei der Orange Pi Zero wurde das neuere 'Jessie server" Legacy Image (Kernel 3.4.113) verwendet.

Alternativ gibt es noch Betriebssystem-Images von der Seite http://www.orangepi.org/orangepibbsen/forum.php?mod=viewthread&tid=342. Diese sind aber veraltet und wurden länger nicht gewartet. Sie verwenden noch die originalen CPU Einstellungen des Herstellers sodass die CPU auf 1,53 GHz taktet. Diese Taktrate führt aber ohne Verbesserungsmaßnahmen der Kühlung zu einer Überhitzung. Dadurch kommt es zu einer automatischen Reduzierung der CPU-Leistung. Beim Armbian Image werden sinnvollere Einstellungen verwendet sodass die CPU auf maximal 1,3 GHz beim Orange Pi PC und 1,2 GHz beim Orange Pi Lite und One taktet. Dadurch ist ein stabiler und konstanter Betrieb möglich, auch ohne Kühlkörper auf der CPU.

Linux

```
apt-get install p7zip-full
cd /tmp
wget http://mirror.igorpecovnik.com/Armbian_5.20_Orangepipc_Debian_jessie_3.4.112.7z
7z e Armbian_5.20_Orangepipc_Debian_jessie_3.4.112.7z
rm Armbian_5.20_Orangepipc_Debian_jessie_3.4.112.7z
```

Nun kann die enthaltene Image Datei mit dem Programm dd auf eine MicroSD-Karte übertragen werden.

Es ist unbedingt vor dem Ausführen des Befehls zu prüfen, ob das angegebene Laufwerk bzw. Device auch der vorgesehenen MicroSD-Karte entspricht!

dmesg | tail -n 10

```
sd 4:0:0:4: [sdg] Attached SCSI removable disk sd 4:0:0:0: [sdc] 30392320 512-byte logical blocks: (15.5 GB/14.4 GiB) sd 4:0:0:1: [sdd] Attached SCSI removable disk sd 4:0:0:3: [sdf] Attached SCSI removable disk sd 4:0:0:0: [sdc] Write Protect is off sd 4:0:0:0: [sdc] Wode Sense: 03 00 00 00 sd 4:0:0:0: [sdc] No Caching mode page found sd 4:0:0:0: [sdc] Assuming drive cache: write through sdc: sdc1 sdc2 sd 4:0:0:0: [sdc] Attached SCSI removable disk
```

dd if=Armbian_5.20_Orangepipc_Debian_jessie_3.4.112.img of=/dev/sdc bs=1M

```
1190+0 Datensätze ein
1190+0 Datensätze aus
1247805440 Bytes (1,2 GB) kopiert, 118,934 s, 10,5 MB/s
```

Windows

Zuerst muss man die erhaltene Datei "Armbian_5.20_Orangepipc_Debian_jessie_3.4.112.7z" entpacken. Dazu kann das Programm 7-Zip verwendet werden (Download auf http://www.

7-zip.de/download.html). Die Image-Datei "Armbian_5.20_Orangepipc_Debian_jessie_3.4.112.img" kann mit dem Programm "Rufus" (Download auf https://rufus.akeo.ie/) auf eine MicroSD-Karte übertragen werden. Dazu klickt man das Festplatten/CD Symbol an (Wählen Sie eine Abbild-Datei...) und wählt dann die Datei aus. Unter Laufwerk wählt man das Laufwerk aus, in dem sich die zu schreibende MicroSD-Karte befindet. Achtung: Daten von dem Laufwerk werden überschrieben! Nach dem Drücken von "Start" beginnt der Schreibvorgang.



Nun kann die MicroSD-Karte in die Orange Pi gesteckt und an die Versorgung angeschlossen werden. Danach beginnt der Einrichtungsprozess von selbst, dies kann einige Minuten dauern.

3.3.2 Einstellungen

Beim ersten Anmeldungsvorgang wird der Benutzer "root" mit dem Passwort "1234" verwendet. Danach muss das Passwort geändert werden. Dazu gibt man einmal das alte Passwort und zweimal das neue Passwort ein. Danach wird das System automatisch neu gestartet.

```
Welcome to ARMBIAN Debian GNU/Linux 8 (jessie) 3.4.112-sun8i
System load:
               0.18
                                Up time:
                                               2 min
Memory usage:
               5 % of 1000Mb
                                IP:
                                               192.168.0.102
               22°C
CPU temp:
Usage of /:
               8% of 15G
Last login: Mon May 9 21:53:53 2016
Thank you for choosing Armbian! Support: www.armbian.com
Creating new account. Please provide a username (eg. your forename): orangepi
```

Nach der Anmeldung kann man einen neuen normalen Benutzer anlegen. Dazu muss man einen Namen und ein Passwort vergeben.

```
Adding user 'orangepi' ..
Adding new group 'orangepi' (1000) ...
Adding new user 'orangepi' (1000) with group 'orangepi' ...
Creating home directory 'home/orangepi' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for orangepi
Enter the new value, or press ENTER for the default
         Full Name []:
         Room Number []:
         Work Phone []:
         Home Phone []:
         Other []:
Is the information correct? [Y/n]
Y
Dear , your account orangepi has been created and is sudo enabled.
```

Please use this account for your daily work from now on.

Dann könnte man noch die Auflösung des Displays ändern. Dies kann aber auch jederzeit nachträglich mit dem Programm h3disp durchgeführt werden.

```
Your display settings are currently 720p (1280x720). To change this use the h3disp utility. Do you want to change display settings now? [nY]
```



Die weiteren Einstellungen müssen manuell durchgeführt werden.

Tastatur-Layout ändern:

dpkg-reconfigure console-data

Policy for handling keymaps: Select keymap from full list Keymap: pc / qwertz / German / Standard / latin1 - no dead keys

Sprache und Zeichensatz des Systems setzen:

dpkg-reconfigure locales

Hier wählt man z. B. "de_AT.UTF-8 UTF8" für Österreich oder "de_DE.UTF-8 UTF8" für Deutschland. "en_US.UTF-8 UTF8" und "C.UTF-8" ist bereits ausgewählt und kann zusätzlich aktiv sein. Im nächsten Fenster kann man dann die Sprache "de_AT.UTF-8 UTF8" bzw. "de_DE.UTF-8 UTF8" als Standardeinstellung übernehmen.

Zeitzone des Systems setzen:

dpkg-reconfigure tzdata

Für das geografische Gebiet kann man "Europe" auswählen. Danach kann man als Zeitzone die Stadt "Vienna" oder "Berlin" auswählen.

Hostname ändern:

echo OrangePiPC > /etc/hostname

Zum Schluss kann man dann das Betriebssystem noch aktualisieren und unnötig belegten Speicher auf der MicroSD-Karte freigeben.

```
apt-get update && apt-get upgrade
sudo apt-get clean
reboot
```

3.3.3 WiringPi C-Bibliothek

WiringPi ist eine C-Bibliothek für den Zugriff auf Funktionen der GPIO's des Prozessors. Sie wurde von Gordon Henderson für den Raspberry Pi programmiert. Sie wird unter der GNU LGPLv3 Lizenz angeboten.

Diese Bibliothek wurde für verschiedene Einplatinencomputer wie den Banana Pi und Orange Pi angepasst und veröffentlicht. Die portierte C-Bibliothek für den Orange Pi trägt den Projektnamen WiringOP. Leider wird sie aber nicht ständig aktualisiert (ältere Version gegenüber Raspberry Pi) und für den Orange Pi wurden nur teilweise Anpassung durchgeführt. Der Aufruf "gpio -v" meldet z. B. "Banana Pro, Revision: 1.2" statt Orange Pi.

```
cd /usr/src
git clone https://github.com/zhaolei/WiringOP.git -b h3
cd WiringOP
chmod +x ./build
./build

gpio -v

gpio version: 2.20
Copyright (c) 2012-2014 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Banana Pro Details:
    Type: Banana Pro, Revision: 1.2, Memory: 1024MB, Maker: LeMaker
```

gpio readall

++		+	+	+-+	Ora	ange	ePiI	PC-	+	+	+	++
BCM	wPi	Name	Mode	VI	Phy	/sic	cal	V	Mode	Name	wPi	BCM
++		+	+	+-+		-++-		+-+	+	+	+	++
		3.3v			1	$ \cdot $	2			5v		
2	8	SDA.0	ALT5	0	3	\Box	4			5V		
3	9	SCL.0	ALT5	0	5	\Box	6			0 v		
4	7	GPIO.7	ALT3	0	7	\Box	8	0	ALT3	TxD3	15	14
		0v			9	\Box	10	0	ALT3	RxD3	16	15
17	0	RxD2	ALT3	0	11	\Box	12	0	ALT3	GPIO.1	1	18
27	2	TxD2	ALT3	11	13	\Box	14			0 v		1 1
22	3	CTS2	ALT3	0	15	\Box	16	0	ALT3	GPIO.4	4	23
		3.3v			17	\Box	18	0	ALT3	GPIO.5	5	24
10	12	MOSI	ALT4	0	19	\Box	20			0 v		1 1
9	13	MISO	ALT4	0	21	\Box	22	0	ALT3	RTS2	16	25
11	14	SCLK	ALT4	0	23	\Box	24	0	ALT4	CEO	10	8
		0v			25	\Box	26	0	ALT3	GPIO.11	11	7
0	30	SDA.1	ALT4	0	27	\Box	28	0	ALT4	SCL.1	31	1
5	21	GPIO.21	ALT3	0	29	\Box	30			0 v		1 1
6	22	GPIO.22	ALT3	0	31	\Box	32	0	ALT3	RTS1	126	12
13	23	GPIO.23	ALT3	0	33	\Box	34			0 v		1 1
19	24	GPIO.24	ALT3	0	35	\Box	36	0	ALT3	CTS1	27	16
26	25	GPIO.25	ALT3	0	37	\Box	38	0	ALT3	TxD1	28	20
1 1		0v	l	ĺ	39	$ \cdot $	40	0	ALT3	RxD1	129	21
++		+	+	+-+		-++-		+	+	+	+	++
BCM	wPi	Name	Mode	VI	Phy	sio	cal	١V	Mode	Name	wPi	BCM
+++OrangePIPC+++												

Beispiel Kompilierung mit WiringPi und Mathematik Bibliothek:

```
gcc -o spi_test spi_test.c -Wall -lwiringPi -lm
```

3.3.4 Einrichtung WLAN (Orange Pi Lite/Zero)

ifconfig wlan0 up iwlist wlan0 scan

```
wlan0
          Scan completed :
Cell 01 - Address: 00:01:B3:1C:73:7F
 ESSID: "ESSID_OF_WLAN"
 Protocol: IEEE 802.11bg
 Mode:Master
 Frequency: 2.447 GHz (Channel 8)
 Encryption key:on
 Bit Rates:54 Mb/s
 Extra:wpa_ie=dd180050f20201000050f20101000040f20201000040f2010000
 IE: WPA Version 1
   Group Cipher : TKIP
   Pairwise Ciphers (1) : TKIP
   Authentication Suites (1): PSK
 Quality=0/100 Signal level=68/100
 Extra:fm=0003
```

nmtui

Please select an option: Activate a connection

WiFi: ESSID_OF_WLAN

Passwords or encryption keys are required to access the wireless network 'ESSID' OF WLAN'.

Password:

3.3.5 Ressource Monitor

armbianmonitor -r

Browser: http://<IP-Address>:8888

3.3.6 Kühlung

Der H3 Prozessor der Orange Pi kann sich unter Last sehr schnell auf gefährliche Temperaturen erhitzen. Passiert dies so werden Schutzmechanismen aktiv. Sie reduzieren automatisch die maximale CPU-Frequenz und können im Extremfall sogar CPU-Kerne abschalten. Auch wenn es so nicht zur Beschädigung kommt, so sollte dieser Bereich doch gemieden werden, da die CPU-Performance dann einbricht.

Ein passiver Kühlkörper und ein offene Gehäuse sollte man dem Orange Pi gönnen. Im Extremfall kann aber auch diese Maßnahme nicht vor Überhitzung schützen. Wird das System auch für längeren Volllastbetrieb mit vier Kernen ausgelegt, so muss ein zumindest leicht drehender Lüfter verwendet werden.

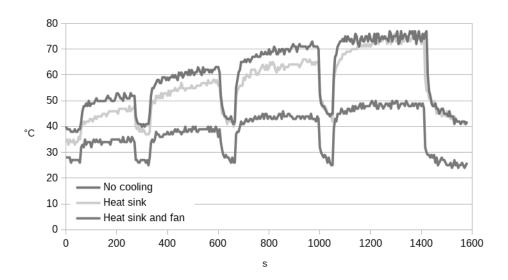
Im folgenden Diagramm wurde ein Orange Pi PC einmal ohne Kühlkörper, einmal mit Kühlkörper und einem mit zusätzlich noch einem 12 V 40x40 Lüfter, der mit nur 5 V betrieben wurde, gekühlt. Der Lüfter ist durch die niedrige Spannung sehr leise führt aber zu einer massiv kühleren Betrieb. Beim Kühlkörper können verschiede Modell mit Klebestreifen verwendet werden, z.B. RAM Kühler aus Aluminium oder spezielle Kühler aus Keramik die oft als Set mit anderen Komponen Angeboten werden.

Keramik Kühler:

https://de.aliexpress.com/wholesale?SearchText=heat+sink+ceramic+Pi

Aluminium RAM Kühler:

https://de.aliexpress.com/wholesale?SearchText=heat+sink+aluminum+RAM+DDR



3.3.7 Übertakten

Der H3 Prozessor des Orange Pi PC wird bei Armbian maximal mit 1296 MHz betrieben. Der Prozessor unterstützt aber eine maximale Taktrate von 1536 MHz. Das Problem bei dieser Taktrate ist, dass die CPU-Spannung dafür von 1,3 auf 1,5 V angehoben werden muss. Dadurch erhöht sich die Verlustleistung (Wärme die abgeführt werden muss) enorm. Eine passive Kühlung mit Kühlkörper reicht dann nicht mehr aus. Man benötigt eine aktive Kühlung mit Lüfter. Selbst aktiv gekühlt kann es sein, dass die CPU bei maximaler Belastung zu heiß wird und sich automatisch heruntertaktet. Empfohlen wird darum, dass bei Übertaktung auf 1,5 GHz zugleich 2 Kerne abgeschaltet werden. Dann erreicht man ungefähr das Niveau von 4 Kernen mit 1,3 GHz was Stromaufnahme und CPU-Temperatur betrifft. Sinn macht dies nur wenn wenn eine höhere Singlecore Geschwindigkeit benötigt, wie z. B. bei Spiel-Emulatoren oder anderen Programmen die nur einen Thread benutzen.

Damit die Übertaktung der CPU Funktioniert muss man mehrere Anpassungen vornehmen. Die neue maximale Taktrate muss in der FEX-Datei bzw. "script.bin" Datei definiert werden. Zusätzlich muss noch in der Konfigurationsdatei "cpufrequtils" die maximal zu verwendende Taktrate spezifiziert werden. Möchte man die Anzahl der aktiven Kerne reduzieren so kann dies in der Datei "rc.local" geschehen. Dazu müssen die angegeben Zeilen vor der Anweisung "exit 0" eingefügt werden. In der FEX-Datei muss "corekeeper_enabled = 0" gesetzt werden damit die Kerne nicht automatisch reaktiviert werden.

```
cd /boot/bin/
bin2fex orangepipc.bin orangepipc.fex
cp orangepipc.fex orangepipc_oc.fex
sed -i "s/1296000/1536000/g; s/1320/1500/" orangepipc_oc.fex
sed -i "s/corekeeper_enabled\ =\ 1/corekeeper_enabled = 0/" orangepipc_oc.fex
fex2bin orangepipc_oc.fex orangepipc_oc.bin
mv /boot/script.bin /boot/script.bin.orig
ln -s /boot/bin/orangepipc_oc.bin /boot/script.bin
cp /etc/default/cpufrequtils /etc/default/cpufrequtils.bkup
sed -i "s/1296000/1536000/" /etc/default/cpufrequtils
```

/etc/rc.local

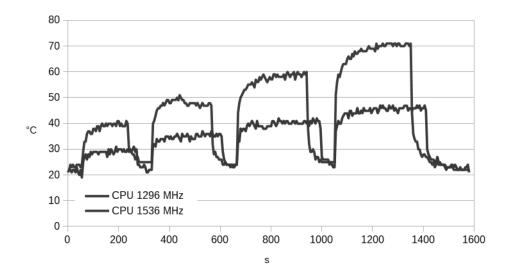
```
# disable core 1 & 3 - quad core to dual core CPU
echo 0 > /sys/devices/system/cpu/cpu1/online
echo 0 > /sys/devices/system/cpu/cpu3/online
exit 0
```

reboot

Im Diagramm sieht man den Temperaturverlauf bei 100 % CPU Last auf 1 bis 4 Kernen. Dieser Belastungstest wurde einmal mit dem Standardtakt von 1296 MHz und einmal mit 1536 MHz durchgeführt. Auf die CPU wurde eine Kühler aufgeklebt und mit einem 12 V 40x40 Lüfter, der mit nur 5 V betrieben wurde, gekühlt.

Wie zu sehen ist beträgt die maximale Temperatur selbst bei aktiver Kühlung um die 70 °C. Dies ist nur wenige Grad unterhalb der Maximaltemperatur für die CPU, bei der zum Schutz die Taktrate automatisch reduziert wird. Wenn man nur zwei Kerne voll auslastet, so ist die maximale Temperatur bei ca. 50 °C. Im Normalbetrieb liegt die maximale Temperatur bei allen 4 Kernen bei ca. 47 °C.

bench.sh



```
#!/bin/bash
FILENAME="Temp1to4CoreSysbench.log"
armbianmonitor -m | tee $FILENAME &
sleep 60
sysbench --num-threads=1 --cpu-max-prime=10000 --test=cpu run
sleep 60
sysbench --num-threads=2 --cpu-max-prime=20000 --test=cpu run
sleep 60
sysbench --num-threads=3 --cpu-max-prime=30000 --test=cpu run
sleep 60
sysbench --num-threads=4 --cpu-max-prime=40000 --test=cpu run
sleep 180
killall armbianmonitor
```

Kapitel 4

USB Gadget / OTG Modus

Beim USB-Gadget oder OTG-Betrieb kann ein Einplatinencomputer direkt über den Micro-USB-Anschluss mit einem PC oder Laptop verbunden werden. Der Einplatinencomputer verhält sich dann wie ein USB-Gerät und kann z. B. ein Massenspeicher-, Serielles- oder Netzwerkgerät simulieren.

Beim seriellen Gerät, wird nach dem Anschließen eine virtuelle serielle Schnittstelle am Host-PC installiert. Über den virtuellen COM-Port kann dann mithilfe eines Terminal Programms eine Verbindung zum Einplatinencomputer hergestellt werden.

Verhält er sich als Netzwerkgerät kann eine Netzwerkverbindung über ein virtuelles Netzwerk zum Gerät hergestellt werden. Hier ist allerdings eine erweitere Konfiguration und evtl. Zusatzprogramme nötig.

Weitere Informationen über den OTG-Betrieb und alle möglichen Simulationsvarianten kann der Git-Hub Seite https://gist.github.com/gbaman/50b6cca61dd1c3f88f41 entnommen werden. In weitere Folge wird die Einrichtung eines serielles Geräts (Serial mode), Massenspeichergeräts (Mass storage mode) und Netzwerkgeräts (Ethernet mode) beschrieben.

4.1 Stromverbrauch/Optimierungen

Beim USB-Gadget Betrieb wird der Einplatinencomputer üblicherweise vom USB-Bus versorgt. Unter Umständen kann der angeschlossene USB-Host Kontroller aber nur die spezifizierten 500 mA liefern. Die Stromaufnahme vom Gerät sollte also mit ausreichend Reserven unter dieser Grenze liegen.

Im folgenden wurde die Stromaufnahme von Raspberry Pi Zero, Orange Pi Lite und Orange Pi Zero 512 verglichen. Als Referenz wurden auch noch die Werte von dem Raspberry Pi 2 ermittelt. Er kann allerdings nicht im OTG-Modus betrieben werden.

4.1.1 Raspberry Pi Zero

Da der Raspberry Pi Zero mit der Normalkonfiguration nur eine geringe Stromaufnahme hat wurde auf eine Optimierung verzichtet. Bei der folgenden Vergleichsmessung wurde der Einplatinencomputer über USB OTG mit einem Laptop verbunden.

4.1.2 Orange Pi Lite

Da der Orange Pi Lite mit der Normalkonfiguration eine zu hohe Stromaufnahme besitzt wurden einige Optimierung aktiviert die den Stromverbrauch senken. Es wurde die CPU Frequenz von 1,2 GHz auf 912 MHz reduziert. Dadurch wird die CPU nur noch maximal mit 1,1 V versorgt und somit sinkt die Stromaufnahme im Volllastbetrieb. Die Stromaufnahme im Idle-Betrieb (ohne CPU Last) wurde hauptsächlich durch Deaktivierung der Grafikeinheit und Reduzierung des RAM-Taktes erreicht. Dieser wurde von 624 MHz auf 264 MHz reduziert. CPU-Kerne, USB und WLAN wurden nicht abgeschaltet. Dies könnte die Stromaufnahme für den Idle- und Vollastbetrieb noch weiter reduzieren, ist aber für den OTG-Betrieb nicht unbedingt nötig. Die Optimierung erfolgte mit dem Armbian Programm "h3consumption", das sich beim ersten Aufruf selbst installiert.

Bei der folgenden Vergleichsmessung wurde der Einplatinencomputer über WLAN mit einem Laptop verbunden. Die Stromaufnahme der WLAN-Schnittstelle betrug 100 mA.

h3consumption

```
Please be patient, external requirements are to be installed.
Usage: h3consumption [-h/-H] [-p] [-g on|off] [-m max_cpufreq] [-c 1|2|3|4]
      [-d dram_freq] [-D dram_freq] [-u on|off] [-e on|off|fast]
This tool allows to adjust a few consumption relevant settings of your
H3 device. Use the following switches
               displays help or verbose help text
               print currently active settings
 -p
 -g on|off
               disables GPU/HDMI for headless use
 -m \max_cpufreq adjusts \max_maximum allowed cpu clockspeed (mhz)
 -c 1|2|3|4
               activate only this count of CPU cores
 -d dram_freq
                adjusts dram clockspeed (408 - 624 mhz)
               like -d but as low as 132 mhz possible (experimental!)
 -D dram_freq
 -u on|off
                enables/disabled all USB ports
               enables/disables Ethernet, the fast switch
 -e onlofflfast
                forces 100 mbits/sec negotiation on gigabit devices
h3consumption -p
         1200 mhz allowed, 1200 mhz possible, 4 cores active
cpu
dram
         624 mhz
hdmi/qpu active
usb ports active
         IEEE 802.11bg ESSID:"ESSID_OF_WLAN" Nickname:"<WIFI@REALTEK>"
Mode:Managed Frequency:2.447 GHz Access Point: 00:01:A3:0D:73:8F
wlan0
         Bit Rate:54 Mb/s
                           Sensitivity:0/0
         Retry:off RTS thr:off Fragment thr:off
         Encryption key:****-****-****-**** Security mode:open
         Power Management:off
         Link Quality=92/100 Signal level=63/100 Noise level=0/100
         Rx invalid nwid: 0 Rx invalid crypt: 0 Rx invalid frag: 0
         Tx excessive retries:0 Invalid misc:0
h3consumption -d 408
h3consumption -D 264
h3consumption -g off
h3consumption -m 912
reboot
h3consumption -p
Active settings:
         1200 mhz allowed, 912 mhz possible, 4 cores active
cpu
```

```
264 mhz
dram
hdmi/gpu
         off
usb ports active
          Mode:Managed Frequency:2.447 GHz Access Point: 00:01:A3:0D:73:8F
wlan0
                            Sensitivity:0/0
          Bit Rate:54 Mb/s
          Retry:off RTS thr:off Fragment thr:off
          Encryption key: ***-***-***
                                                  - * * * * - * * * * - * * *
                                                                  Security mode:open
          Power Management:off
          Link Quality=92/100 Signal level=63/100 Noise level=0/100
          Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
          Tx excessive retries:0 Invalid misc:0
                                                   Missed beacon: 0
```

4.1.3 Orange Pi Zero

Der Orange Pi Zero hat mit der Normalkonfiguration bereits eine Stromaufnahme von unter 500 mA, dennoch kann die Stromaufnahme noch reduziert werden. Es wurde die CPU Frequenz von 1,2 GHz auf 912 MHz reduziert. Dadurch wird die CPU nur noch maximal mit 1,1 V versorgt und somit sinkt die Stromaufnahme im Volllastbetrieb. Die Stromaufnahme im Idle-Betrieb (ohne CPU Last) wurde hauptsächlich durch Reduzierung des RAM-Taktes erreicht. Dieser ist auf niedrige 408 MHz eingestellt kann aber noch auf 264 MHz weiter reduziert werden. CPU-Kerne, USB, LAN und WLAN wurden nicht abgeschaltet. Dies könnte die Stromaufnahme für den Idle- und Vollastbetrieb noch weiter reduzieren, ist aber für den OTG-Betrieb nicht unbedingt nötig. Die Optimierung erfolgte mit dem Armbian Programm "h3consumption", das sich beim ersten Aufruf selbst installiert.

Beim der folgenden Vergleichsmessung wurde der Einplatinencomputer über LAN und einem Switch mit einem Laptop verbunden. Die Stromaufnahme der LAN-Schnittstelle betrug dabei nur 10 mA.

h3consumption -p

```
1200 mhz allowed, 1200 mhz possible, 4 cores active
          408 mhz
hdmi/gpu off
usb ports active
          100Mb/s/Full, Link: ves
eth0
wlan0
          IEEE 802.11bgn ESSID:off/any
          Mode: Managed Access Point: Not-Associated
                                                       Tx-Power=20 dBm
          Retry long limit:7
                                RTS thr:off
          Encryption key:off
          Power Management:on
h3consumption -D 264
h3consumption -m 912
reboot
          912 mhz allowed, 1200 mhz possible, 4 cores active
cpu
dram
          264 mhz
          off
hdmi/gpu
usb ports active
          100Mb/s/Full, Link: ves
eth0
wlan0
          IEEE 802.11bgn ESSID:off/any
          Mode: Managed Access Point: Not-Associated
          Retry long limit:7
                                RTS thr:off
                                              Fragment thr:off
          Encryption key:off
          Power Management:on
```

4.1.4 Vergleich

Teilweise wird angegeben, dass beim Allwinner H3/H2+ Prozessor der Orange Pi ein Kühlkörper eingesetzt werden soll. Im folgenden Diagramm wurde der Temperaturverlauf bei einer maximalen CPU-Last (auf allen 4 Kernen) aufgezeichnet. Hier konnten keine kritischen Temperaturen erreicht werden, auch nicht beim Betrieb mit 1,3 GHz mit dem Orange Pi PC.

	Takt	Idle	1xCPU	4xCPU	WLAN	LAN
	MHz	mA	mA	mA	mA	mA
Orange Pi Lite	1200	230	340	590	690	-
Orange Pi Lite (Opt.)	912	150	180	270	370	-
Orange Pi Zero	1200	160	240	460	-	470
Orange Pi Zero (Opt.)	912	130	170	260	-	270
Raspberry Pi Zero	1000	110	190	-	-	-
Raspberry Pi 2	900	230	280	410	-	450



Abbildung 4.1: Stromaufnahme OTG-Betrieb

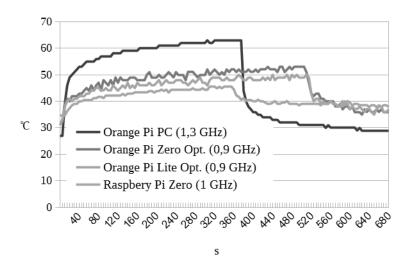


Abbildung 4.2: Temperaturverlauf OTG-Betrieb

4.2 Serielles Gerät (Serial mode)

4.2.1 Raspberry Pi Zero - Client

Nach der Einrichtung des Betriebssystems müssen noch ein paar Modifikationen an den Dateien der Boot-Partition durchgeführt werden.

Folgender Text muss nach der Anweisung "rootwait" in die Datei "cmdline txt" eingefügt werden:

modules-load=dwc2,g_serial

Folgende Zeile muss am Ende in die Datei "config.txt" hinzugefügt werden:

dtoverlay=dwc2

Um eine Verbindung über die virtuelle serielle Schnittstelle herstellen zu können muss am System folgender Befehl ausgeführt werden. Um dies zu ermöglichen, muss man sich über die serielle Schnittstelle (UART zu USB Adapter wird benötigt) verbinden.

sudo systemctl enable getty@ttyGS0.service

4.2.2 Orange Pi Lite/Zero - Client

Der serielle Gadget Mode ist bei der Orange Pi standardmäßig aktiviert. Dies wurde dadurch erreicht, dass in der Datei "/etc/modules" das Modul "g_serial" eingetragen ist. Zusätzlich ist ein Terminal auf dieser Verbindung aktiviert.

4.2.3 Windows - Host

Die Einplatinencomputer verwenden alle die selben Einstellungen für die Verbindung:

Baudrate: 115200 Datenbit: 8 Stopbit: 1 Parity: None Flowcontrol: None

Möglicherweise muss für die Installation des virtuellen COM-Ports ein Treiber installiert werden. Dieser kann als INF-Datei unter https://www.kernel.org/doc/Documentation/usb/linux-cdc-acm.inf heruntergeladen werden.

4.3 Massenspeichergerät (Mass storage mode)

Beim Massenspeichergerät oder "Mass storage mode" kann der Einplatinencomputer einen USB-Massenspeicher simulieren. Die Daten können dabei in einer Partition oder Datei gespeichert werden. Die Partition kann auch im Speicher mit einer RAM-Disk angelegt werden.

Nach der Einrichtung des Betriebssystems müssen noch ein paar Modifikationen an den Dateien der Boot-Partition durchgeführt werden. Folgender Text muss nach der Anweisung "rootwait" in die Datei "cmdline.txt" eingefügt werden:

ramdisk_size=49152 modules-load=dwc2

Folgende Zeile muss am Ende in die Datei "config.txt" hinzugefügt werden:

```
dtoverlay=dwc2
```

Der Eintrag "ramdisk_size" gibt die größer der RAM-Disk in Byte an. Es werden also im Beispiel 48 MB für eine RAM-Disk bzw. Massenspeichergerät vorgesehen.

4.3.1 Partition / RAM-Disk

/etc/modprobe.d/massstorage.conf [-rw-r--r-- root root]

```
options g_mass_storage file=/dev/ram0 stall=0 ro=1
```

Der Parameter "ro=1" gibt an, dass der Speicher nur lesend ist.

Der Parameter "file" gibt an die Partition an, in dem Fall die erste RAM-Disk.

Windows kann leider die Daten nur lesen wenn eine Partition erzeugt wurde. Dazu wird eine primäre Windows FAT32 Partition erzeugt. Damit Linux darauf zugreifen kann wird diese Partition noch eingehängt bzw. gemountet.

```
(echo o; echo n; echo p; echo; echo; echo; echo t; echo c; echo w) | sudo fdisk /dev/ram0
sudo mkdosfs /dev/ram0p1
sudo mkdir /ramdisk
sudo mount /dev/ram0p1 /ramdisk
```

Nun kann die vorbereitet Windows Partitionen per USB-OTG zur Verfügung gestellt werden. Achtung: Danach dürfen keine Daten vom Host-System auf dem Dateisystem geändert werden! Am Besten ist wenn man das Dateisystem aushängt bzw. unmount ausführt.

```
sudo umount /ramdisk
sudo modprobe g_mass_storage
```

Wenn man die Daten wieder freigeben will muss man folgende Befehle ausführen.

sudo rmmod g_mass_storage

4.3.2 Datei

/etc/modprobe.d/massstorage.conf [-rw-r--r- root root]

```
options g_mass_storage file=/USBMS.bin stall=0 ro=1 \,
```

Der Parameter "ro=1" gibt an, dass der Speicher nur lesend ist.

Der Parameter "file" gibt die Datei an in der eine virtuelle Festplatte zur Verfügung gestellt wird.

Windows kann leider die Daten nur lesen wenn eine Partition erzeugt wurde. Dazu wird eine primäre Windows FAT32 Partition in der Datei erzeugt. Damit Linux darauf zugreifen kann wird diese Partition noch über das Loop-System eingehängt bzw. gemountet. Da am Anfang die Partiontionstabelle liegt muss dabei ein Offset angegeben werden. Dieser errechnet sich aus Startwert multipliziert mit Sektor-Größe also im Beispiel 2048*512=1048576.

```
sudo dd if=/dev/zero of=/USBMS.bin bs=1M count=513
(echo o; echo n; echo p; echo; echo; echo t; echo c; echo w) | sudo fdisk /USBMS.bin

fdisk -lu /USBMS.bin

Disk /USBMS.bin: 513 MiB, 537919488 bytes, 1050624 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc50ebd9a

Device Boot Start End Sectors Size Id Type
/USBMS.bin1 2048 1050623 1048576 512M c W95 FAT32 (LBA)

sudo losetup -o 1048576 /dev/loop0 /USBMS.bin
sudo mkdosfs /dev/loop0
sudo mkdir /USBMS
sudo mount /dev/loop0 /USBMS
```

Die Partition kann auch automatisch eingehängt bzw. gemountet werden.

/etc/fstab [-rw-r--r-- root root]

```
/USBMS.bin /USBMS auto loop,offset=1048576 0 0
```

Nun kann die vorbereitet Windows Partitionen zur Verfügung gestellt werden. Achtung: Danach dürfen keine Daten vom Host-System auf dem Dateisystem geändert werden!

```
sudo modprobe q_mass_storage
```

Wenn man die Daten wieder freigeben will muss man folgende Befehl ausführen.

```
sudo rmmod g_mass_storage
sudo umount /USBMS
sudo losetup -d /dev/loop0
```

4.4 Ethernetgerät (Ethernet mode)

4.4.1 Raspberry Pi Zero - Client

Einstellungen Boot System

Nach der Einrichtung des Betriebssystems müssen noch ein paar Modifikationen an den Dateien der Boot-Partition durchgeführt werden. Unter Windows sollte man ein Editor, wie z. B. Notepad++ (https://notepad-plus-plus.org/download/) verwenden, der mit Linux Zeilenumbrüchen zurecht kommt.

Folgender Text muss nach der Anweisung "rootwait" in die Datei "cmdline.txt" eingefügt werden:

```
modules-load=dwc2,g_ether g_ether.host_addr=00:01:02:03:04:05 g_ether.dev_addr=00:01:02:03:04:06
```

Vor und nach dem Text muss ein Leerzeichen stehen.

Die Angabe der MAC-Adresse für Host und Gerät ist optional, es wird aber empfohlen da sonst diese Adressen zufällig vergeben werden. Die Werte können frei gewählt werden, sollten sich aber nicht mit den Adressen im Netz bzw. Host überschneiden.

Folgende Zeile muss am Ende in die Datei "config.txt" hinzugefügt werden:

dtoverlay=dwc2

Weiters muss eine leere Datei mit dem Namen "ssh" erzeugt werden damit der SSH-Dienst automatisch nach dem Start ausgeführt wird. Danach kann die MicroSD-Karte in den Raspberry Pigesteckt und an einen Computer angeschlossen werden. Es ist zu beachten, dass das USB-Kabel am mittleren MicroUSB-Anschluss angeschlossen wird!

Danach kann der Raspberry Pi Zero bereits mit dem Namen "raspberrypi.local" erreicht werden wenn ZeroConf am Host-System verfügbar ist.

Einstellungen Netzwerk (Statische IP-Adresse)

Folgende Einstellungen müssen nicht gemacht werden, erleichtern aber das Arbeiten. Der Raspberry Pi Zero hat dann eine statische IP-Adresse und kann unter Microsoft Windows leichter angesprochen und auch die Internetverbindung freigegeben werden.

Um die folgenden Einstellungen vornehmen zu können gibt es verschiedene Möglichkeiten:

- Man modifiziert die Daten direkt auf der MicroSD-Karte (Unter Windows nicht direkt möglich)
- Man verbindet sich über die serielle Schnittstelle (UART zu USB Adapter wird benötigt)
- Man verbindet sich über USB (OTG Ethernet) mit der virtuellen Netzwerkkarte mit dynamischer IP-Adresse (Unter Windows wird Programm Bonjour benötigt)

Am Client System, also der Raspberry Pi Zero, kann die Netzwerkadresse, der Gateway und ein DNS-Server eingestellt werden. Dieser Schritt ist unbedingt nötig wenn die Internetverbindung dem Gerät zur Verfügung gestellt werden soll. Die IP-Adresse die eingestellt wird muss für Windows aus dem Bereich 192.168.137.* sein (z. B. 192.168.137.10). Der Gateway (Routers) ist die IP-Adresse des Host-PC. Als DNS-Server kann z. B. der Server von Google mit der Adresse 8.8.8.8 verwendet werden. Die Einstellungen können in der Konfigurationsdatei für den DHCP-Client definiert werden.

Folgende Zeilen müssen am Ende der Datei eingefügt werden:

/etc/dhcpcd.conf [-rw-r--r-- root root]

```
# define static profile for Pi
profile static_usb0
static ip_address=192.168.137.10/24
static routers=192.168.137.1
static domain_name_servers=8.8.8.8
# static profile on usb0
interface usb0
fallback static_usb0
```

Optional kann die IP-Adresse des Host-Computers auch via eines DHCP-Server konfiguriert werden. Das hat den Vorteil, dass dort keine Konfiguration des Netzwerks (statische IP-Adresse) erfolgen muss. Allerdings wird dazu ein Terminal Zugang zum Einplatinencomputer benötigt um den DHCP-Server installieren zu können.

sudo apt-get install isc-dhcp-server

Falls die Internetverbindung am Raspberry Pi Zero nicht möglich ist, kann der Server auch manuell heruntergeladen, auf der Boot-Partition gespeichert und installiert werden.

Install-Datei: http://archive.raspbian.org/raspbian/pool/main/i/isc-dhcp/isc-dhcp-server_
4.3.1-6%2bdeb8u2_armhf.deb

```
sudo dpkg -i /boot/isc-dhcp-server_4.3.1-6+deb8u2_armhf.deb
```

Nach der Installation kann der DHCP-Server parametriert werden.

```
sudo nano /etc/dhcp/dhcpd.conf
```

Folgende Zeilen müssen am Ende der Datei eingefügt werden:

/etc/dhcp/dhcpd.conf [-rw-r--r-- root root]

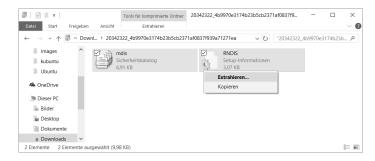
```
subnet 192.168.137.0 netmask 255.255.255.0{
  range 192.168.137.2 192.168.137.10;
  option broadcast-address 192.168.137.255;
}

#USB OTG Host-PC set IP-Address to 192.168.137.1
host USB_OTG_HOST {
  hardware ethernet 00:01:02:03:04:05;
  fixed-address 192.168.137.1;
}
```

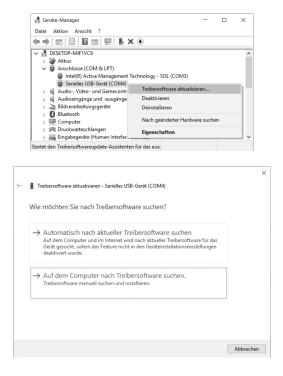
sudo reboot

4.4.2 Windows 10 - Host

Nach dem Boot wird der Raspberry Pi Zero als "Serielles USB-Gerät" erkannt und keine korrekten Treiber installiert. Dies muss manuell erfolgen. Dazu lädt man sich zuerst den zertifizierten Treiber "Acer Incorporated. - Other hardware - USB Ethernet-RNDIS Gadget" von der Microsoft Homepage herunter http://download.windowsupdate.com/msdownload/update/driver/drvs/2012/12/20342322_4b9970e3174b23b5cb2371af0837f939a71271ea.cab. Die benötigten Dateien sind in der komprimierten CAB-Datei enthalten. Durch einen Doppelklick auf der Datei kann der Inhalt dargestellt werden. Mit der rechten Maustaste und dem Kontexmenü können die Dateien z. B. nach c:\Drivers\RNDIS\ extrahiert werden.



Nun muss der Grätemanager geöffnet werden. Dann öffnet man das Kontextmenü in dem man die rechten Maustatste am "Serielles USB-Gerät" Eintrag drückt. Nun wählt man den Menüpunkt "Treiber Software aktualisieren..." aus. Im folgenden Dialog wählt man "Auf dem Computer nach Treibersoftware suchen" und dann gibt man das Verzeichnis an in dem die Treiberdaten extrahiert wurden, z. B. c:\Drivers\RNDIS\. Zum Schluss sollte der Treiber automatisch erfolgreich



installiert werden.

Wenn auf dem Raspberry Pi Zero keine statische IP-Adresse konfiguriert wurde, so muss Bonjour von Apple installiert werden. Falls der Dienst noch nicht von einem Apple Programm (iTunes oder Quicktime) installiert wurde, kann von der Seite https://support.apple.com/kb/DL999? locale=de_AT ein Setup heruntergeladen werden.

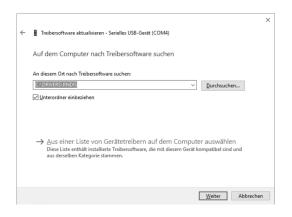
Danach kann die Verbindung mit dem Namen "raspberrypi.local" hergestellt werden. Wenn auf dem Raspberry Pi eine statische IP-Adresse vergeben wurde, so kann auch direkt eine Verbindung hergestellt werden ohne Bonjour installieren zu müssen.

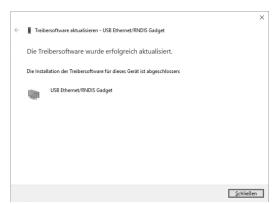
Damit am Gerät Internet funktionieren kann, muss das Internet für das neue Netzwerk freigegeben werden. Dazu öffnet man das Einstellungs-Fenster für die Netzwerkverbindungen.

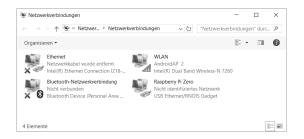
Zuerst kann man dem Netzwerkgerät "USB Ethernet/RNDIS Gadget" einen neuen Namen geben, z. B. Raspberry Pi Zero. Nun muss das Netzwerk gesucht werden, das mit dem Internet verbunden ist, z. B. WLAN. Bei den Eigenschaften zu dem Netzwerk kann der Reiter "Freigabe" ausgewählt werden. Danach kann man die Einstellung "Anderen Benutzern im Netzwerk gestatten, diese Verbindung des Computers als Internetverbindung zu verwenden" aktivieren und bei Heimnetzwerkverbindung kann das Netzwerk "Raspberry Pi Zero" ausgewählt werden.

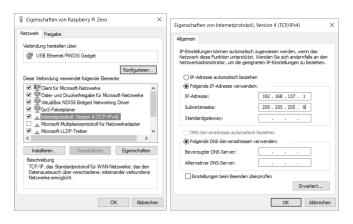
Nun kann die Verbindung zur Raspberry Pi mit dem Programm Putty und der Adresse "192.168.137.10", über das SSH-Protokoll hergestellt werden.

Mit dem Befehl "ping 8.8.8.8" kann die Internetverbindung getestet werden. Mit dem Befehl "ping google.com" kann dann der DNS-Server überprüft werden.







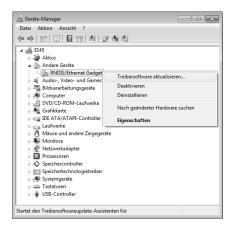




4.4.3 Windows 7 - Host

Nach dem Boot wird automatisch ein neues "RNDIS/Ethernet Gadget" erkannt und Treiber installiert. Sollte die Installation nicht erfolgreich abgeschlossen werden können, so muss man den Treiber manuell installieren.





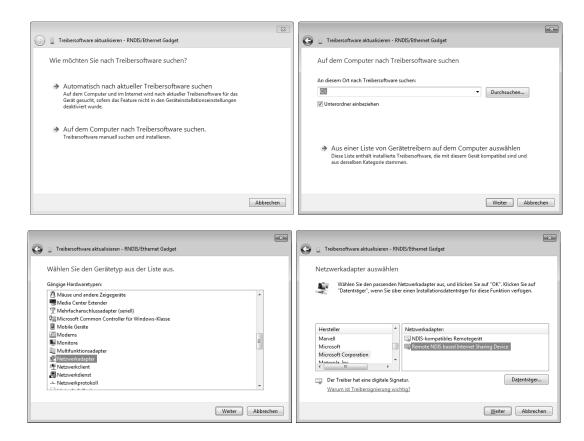
Dazu öffnet man zuerst den Grätemanager. Dann öffnet man das Kontextmenü in dem man die rechten Maustatste am fehlerhaften Gerät (Andere Geräte / RNDIS/Ethernet Gadget) drückt. Nun wählt man den Menüpunkt "Treiber Software aktualisieren..." aus. Im folgenden Dialog wählt man "Auf dem Computer nach Treibersoftware suchen" und dann "Aus einer Liste von Gerätetreibern auf dem Computer auswählen". Danach kann der Gerätetyp gewählt werden in dem man "Netzwerkadapter" auswählt. Dann wählt man den Hersteller "Microsoft Corporation" und den Netzwerkadapter "Remote NDIS based Internet Sharing Device". Sollte eine Kompatibilitätswarnung angezeigt werden, kann der Treiber trotzdem installiert werden. Zum Schluss sollte der Treiber automatisch erfolgreich installiert werden.

Wenn auf dem Raspberry Pi Zero keine statische IP-Adresse konfiguriert wurde, so muss Bonjour von Apple installiert werden. Falls der Dienst noch nicht von einem Apple Programm (iTunes oder Quicktime) installiert wurde, kann von der Seite https://support.apple.com/kb/DL999?locale=de_AT ein Setup heruntergeladen werden.

Danach kann die Verbindung mit dem Namen "raspberrypi.local" hergestellt werden. Wenn auf dem Raspberry Pi eine statische IP-Adresse vergeben wurde, so kann auch direkt eine Verbindung hergestellt werden ohne Bonjour installieren zu müssen.

Damit am Gerät Internet funktionieren kann, muss das Internet für das neue Netzwerk freigegeben werden. Dazu öffnet man das Einstellungs-Fenster für die Netzwerkverbindungen.

Zuerst kann man dem Netzwerkgerät "Remote NDIS based Internet Sharing Device" einen neuen Namen geben, z. B. Raspberry PI ZERO. Nun muss das Netzwerk gesucht werden, das mit dem

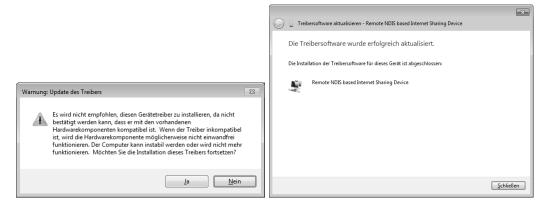


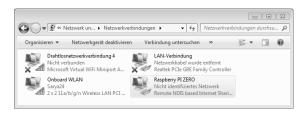
Internet verbunden ist, z. B. Onboard WLAN. Bei den Eigenschaften zu dem Netzwerk kann der Reiter "Freigabe" ausgewählt werden. Danach kann man die Einstellung "Anderen Benutzern im Netzwerk gestatten, diese Verbindung des Computers als Internetverbindung zu verwenden" aktivieren und bei Heimnetzwerkverbindung kann das Netzwerk "Raspberry PI ZERO" ausgewählt werden.

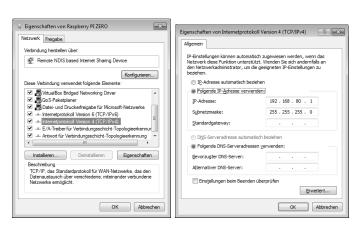
Windows 7 legt dann fest, dass die Netzwerkadresse 192.168.137.1 sein muss. Alternativ kann die Verbindung auch zuvor schon auf diese statische IP-Adresse gesetzt werden.

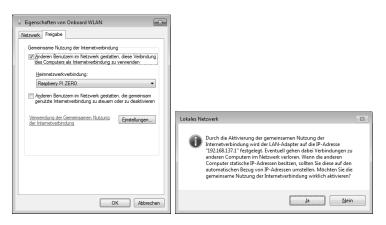
Nun kann die Verbindung zur Raspberry Pi mit dem Programm Putty und der Adresse "192.168.137.10", über das SSH-Protokoll hergestellt werden.

Mit dem Befehl "ping 8.8.8.8" kann die Internetverbindung getestet werden. Mit dem Befehl "ping google.com" kann dann der DNS-Server überprüft werden.









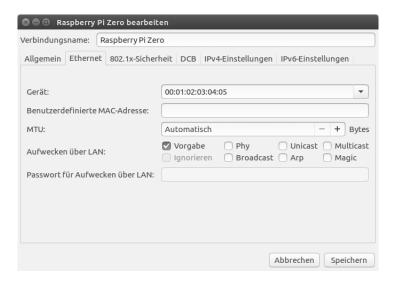
4.4.4 Linux (Ubuntu 16.04) - Host

Zur Konfiguration unter Linux (Ubuntu 16.04) muss zuerst "Netzwerkverbindungen" geöffnet werden. Dazu gibt man lediglich "netzwerkverbindungen" in der Suche ein und wählt das angezeigte Icons an.



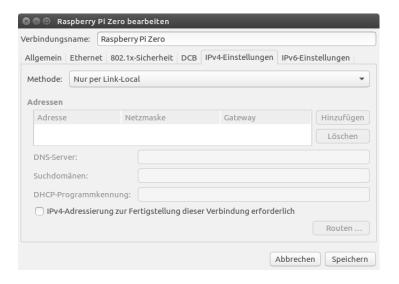
Dann kann die neue Kabelnetzwerkverbindung umbenannt werden, z. B. in Raspberry Pi Zero. Erkennen kann man das Netzwerk an der Mac-Adresse die man bei "g_ether.host_addr" angegeben hat (z. B. 00:01:02:03:04:05).

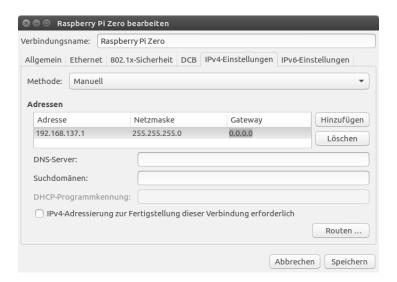




Wurde beim Raspberry Pi Zero keine statische IP-Adresse vergeben so muss das Netzwerk bei den IPv4-Einstellungen auf Methode "Nur per Link-Local" gestellt werden.

Wurde auf der Raspberry Pi Zero eine statische IP-Adresse vergeben (siehe 4.4.1), dann muss am Host-PC bei den IPv4-Einstellungen auf Methode "Manuell" gestellt werden. Dann kann eine statische IP-Adresse von 192.168.137.1 und eine Netmaske 255.255.255.0 vergeben werden.





Nun muss der Name des Netzwerkgeräts gesucht werden, das mit dem Internet verbunden ist. Dazu können alle Netzwerkgeräte und deren Status mit dem folgenden Kommando aufgelistet werden.

nmcli dev status

GERÄT	TYP	STATUS	VERBINDUNG
enx000102030405	ethernet	verbunden	Raspberry Pi Zero
wlp3s0	wifi	verbunden	Surya50
enp0s25	ethernet	nicht verfügbar	
10	loopback	nicht verwaltet	

Im Beispiel ist das WLAN-Gerät (Wifi) wlp3s0 mit dem Internet verbunden. Damit der Internetzugang freigegeben wird, müssen am Host-PC folgende Befehl im Terminal eingeben werden. wlp3s0 muss durch den Namen des Netzwerkgeräts ersetzt werden, das mit dem Internet verbunden ist.

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o wlp3s0 -j MASQUERADE
```

Kapitel 5

GPIO

5.1 Raspberry Pi

Die GPIO's der Raspberry Pi sind sehr gut dokumentiert und haben sich in der Lauf der Zeit bzw. bei den unterschiedlichen Typen nur wenig geändert. Die erste und zweite Revision des Raspberry Pi B's benutzte noch den kleineren 26-Pin GPIO Header. Bei der Revision 1 gibt es noch GPIO1 und GPIO2, die bei den späteren Versionen durch GPIO2 und GPIO3 ersetzt wurden. Die I2C-Bus Nummer wurde beim Wechsel vom Revision 1 auf Revision 2 von 1 auf 0 gewechselt.

Ab der Raspberry Pi B+ wurde dann ein 40-Pin GPIO Header eingeführt. Bei diesem sind die ersten 26-Pin kompatibel mit der Revision 2. Im neuen Bereich des Headers befindet sich nun wieder der I2C-Bus 0, der aber nicht verwendet werden soll. Dieser wird für die automatische Erkennung von Aufsteckboards verwendet.

Es gibt 3 große C-Librarys für die GPIO's. Diese Librarys sind allerdings auch für anderen Sprachen verfügbar.

Pigpio http://abyz.co.uk/rpi/pigpio, no Licence

WiringPi https://github.com/WiringPi/WiringPi, LGPLv3

BCM2835 http://www.airspayce.com/mikem/bcm2835, GPLv3

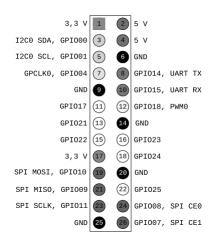


Abbildung 5.1: GPIO's Raspberry Pi Modell B $(\operatorname{Rev}\ 1)$

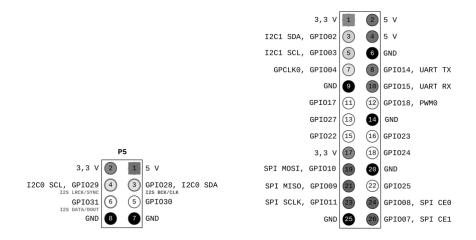


Abbildung 5.2: GPIO's Raspberry Pi Modell A und B (Rev2)

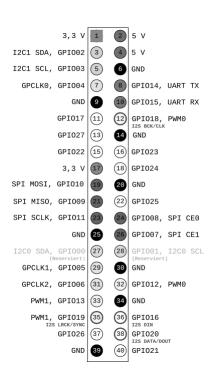


Abbildung 5.3: GPIO's Raspberry Pi $\mathrm{A}+,\,\mathrm{B}+,\,2,\,3$ und Zero (W)

5.2 Banana Pi/Pro

Die GPIO's des Banana Pi/Pro's sind auf der Seite http://wiki.lemaker.org/BananaPro/Pi: Pin_definition dokumentiert. Die GPIOs Pin können teilweise konfiguriert werden, sodass die Funktion auch von der verwendeten Distribution bzw. Einstellungen abhängig sind. Der verwendet Kernel hat auch einen Einfluss auf die Verfügbarkeit und Nummerierung von GPIO Funktionen. Die I2C-Bus Nummer des GPIO Headers wechselt z. B. zwischen Kernel 3.4 und 4.4

Die Parametrierung erfolgt über die binäre Datei "script.bin". Um sie editieren zu können, muss man die Datei mit dem Konverter "bin2fex" in eine Textdatei umwandeln. Darin enthalten sind dann viele Konfigurationsoptionen die als Name-Wert-Paare vorliegen. Nach erfolgter Änderung muss die Datei wieder mit dem Programm "fex2bin" in eine binäre Datei umgewandelt werden. Mehr zu den einzelnen Einstellungen kann man auf der Seite http://linux-sunxi.org/Fex_Guide erfahren.

Ein Beispiel für die Vorgangsweise bei der Parametrierung ist im Kapitel zur Orange Pi enthalten. Um zu den fex- bzw. bin-Dateien zu gelangen muss man die erste Partition der SD-Karte mounten.

```
mount /dev/mmcblk0p1 /mnt
ls -l /mnt/script.bin /mnt/fex/BananaPi
```

```
-rwxr-xr-x 1 root root 50936 Apr 7 2015 /mnt/script.bin
/mnt/fex/BananaPi:
total 78
-rwxr-xr-x 1 root root 50936 Apr 6 2015 script.bin
-rwxr-xr-x 1 root root 28427 Apr 6 2015 script.fex
```

Es gibt mehrere C-Librarys für die GPIO's bei denen es sich um angepasste Varianten der Raspberry Pi Librarys handelt. Die Library WiringLMK ist nur für den Banana Pro verfügbar. Die restlichen Librarys sind für beide Varianten gültig.

WiringLMK (Banana Pro) https://github.com/LeMaker/WiringLMK.git, LGPLv3

WiringBP https://github.com/LeMaker/WiringBP, LGPLv3

BCM2835 BP https://github.com/LeMaker/bcm2835_BP.git, GPLv2

RPi.GPIO BP https://github.com/LeMaker/RPi.GPIO_BP, GPLv2

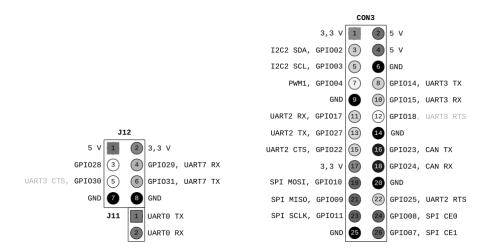


Abbildung 5.4: GPIO's Banana Pi

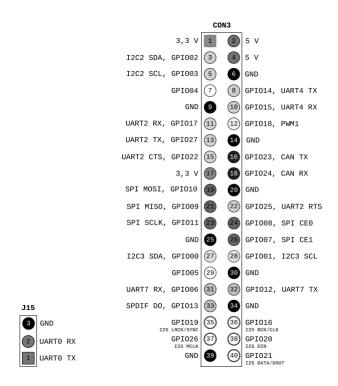


Abbildung 5.5: GPIO's Banana Pro

5.3 Orange Pi PC

Die GPIO's des Orange Pi's sind leider schlecht bzw. auch teilweise fehlerhaft dokumentiert. Die GPIOs Pin können teilweise konfiguriert werden, sodass die Funktion auch von der verwendeten Distribution bzw. Einstellungen abhängig ist.

Die Parametrierung erfolgt über die binäre Datei "script.bin". Um sie editieren zu können, muss man die Datei mit dem Konverter "bin2fex" in eine Textdatei umwandeln. Darin enthalten sind dann viele Konfigurationsoptionen die als Name-Wert-Paare vorliegen. Nach erfolgter Änderung muss die Datei wieder mit dem Programm "fex2bin" in eine binäre Datei umgewandelt werden. Mehr zu den einzelnen Einstellungen kann man auf der Seite http://linux-sunxi.org/Fex_Guide erfahren.

Beispiel:

```
cd /boot
bin2fex script.bin script.fex
cp script.fex script.fex.bkup

fexc-bin: script.bin: version: 0.1.2
fexc-bin: script.bin: size: 35916 (80 sections)
```

/boot/script.fex [-rw-r--r-- root root]

```
[uart3]
;uart_used = 0
uart_used = 1
uart_port = 3
;uart_type = 4
uart_type = 2
uart_tx = port:PA13<3><1><default><default>
uart_rx = port:PA14<3><1><default><default>
;uart_rts = port:PA16<3><1><default><default>
;uart_rts = port:PA16<3><1><default><default>
```

fex2bin script.fex script.bin

Bei Armbian sind alle UART GPIO's vom Pin Header deaktiviert, sodass nur die Ein/Ausgangs-Funktion der GPIO's zur Verfügung stehen. Die Änderung im Beispiel aktiviert RX und TX des UART3 bei Orange Pi PC. Dies entspricht dann der Konfiguration der seriellen Schnittstelle der Raspberry Pi. Allerdings wird die Funktion dort über /dev/ttyAMAO angesprochen. Bei dem Orange Pi PC steht die Schnittstelle unter /dev/ttyS3 zur Verfügung.

Den in manchen Dokumentationen angegebene GPIO PWM1 (Pin 7) gibt es nicht (zumindest nicht mehr im CPU-Datenblatt 1.2, denn dort wurde dieses Kapitel korrigiert). Es gibt nur einen dezidierten Pulsweitenmodulation Pin und dieser liegt auf UART0-RX, also am seriellen Debugeingang des Boards. Zur Nutzung müsste dieser Pin umkonfiguriert werden. Die UART0 Funktion wäre dann nicht mehr gegeben.

Der in manchen Dokumentationen angegebene GPIO CAN gibt es nicht. Die CPU bietet keine CAN-Bus Funktionalität. Der Pin hat somit nur eine normale Ein/Ausgangs-Funktion.

Im Vergleich zum Raspberry Pi 40-Pin Header sind die I2C-Bus Nummern vertauscht. Dort wo bei der Raspberry Pi der I2C-Bus Nummer 1 ist, liegt bei der Orange Pi der mit Nummer 0.

Der SPI-Bus des Orange Pi's bietet nur einen CE-Pin. Bei dem Raspberry Pi sind CE0 und CE1 vorhanden. Dadurch gibt es dort zwei SPI-Kanäle, nämlich /dev/spidev0.0 und /dev/spidev0.1. Bei dem Orange Pi gibt es nur /dev/spidev0.0 bzw. /dev/spidev1.0.

Der H3 Prozessor würde noch einige Funktionen bereitstellen können, diese sind aber durch das Boarddesign des Orange Pi PC/Lite nicht nutzbar. Z. B. sind wichtige Signale des zweiten SPI-Busses für die Status-LEDs verwendet worden, dadurch ist dieser nicht nutzbar. Der SPDIF-Out Ausgang wurde für das CSI-Kamerainterface verwendet.

Achtung: Beim Orange Pi Zero wurde eine abweichende Belegung des 26-Pin Header verwendet. Dies betrifft den I2C1-Bus und viele GPIOs. Beim Orange Pi PC ist nur SPI0 verfügbar, beim Orange Pi Zero ist an dieser Stelle SPI1 (/dev/spidev1.0) verfügbar.

Die offizielle GPIO C-Library "WiringOP" ist nicht verfügbar (https://github.com/orangepi-xunlong/WiringOP). Es gibt aber eine andere Portierung der Raspberry Pi WiringPi bzw. WiringBP Library unter https://github.com/zhaolei/WiringOP.git. Andere Librarys sind zur Zeit nicht verfügbar bzw. bekannt. Auchtung: Beim Orange Pi Zero wurde die abweichende Belegung nicht in der Library korrigiert. Es ergibt sich also eine andere Anordnung der GPIOs als bei der Ausgabe von "gpio readall" (Version Orange Pi PC). Das folgende GPIO Bild wurde dahingehend korrigiert und gibt die richtigen GPIO-Nummern an.

WiringOP https://github.com/zhaolei/WiringOP.git, LGPLv3

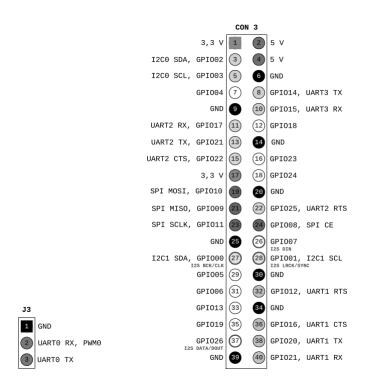


Abbildung 5.6: GPIO's Orange Pi PC/Lite

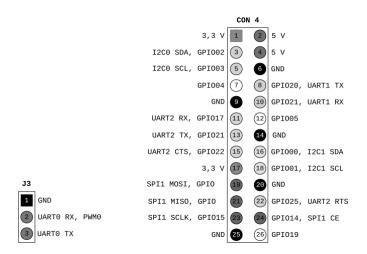


Abbildung 5.7: GPIO's Orange Pi Zero

Kapitel 6

Performance

CPU 6.1

Für die Ermittlung der CPU-Singlecore-Leistung wurde das alte Testprogramm "nbench" verwendet. Auf der Homepage des Projekts sind viele Messergebnisse von anderen Systemen als Referenz verfügbar.

6.1.1Installation

apt-get install build-essential mkdir ~/bench cd ~/bench wget http://fossies.org/linux/misc/nbench-byte-2.2.3.tar.gz tar xzvf nbench-byte-2.2.3.tar.gz cd nbench-byte-2.2.3 make

Ausführung 6.1.2

./nbench

C compiler

BYTEmark* Native Mode Benchmark ver. 2 (10/95) Index-split by Andrew D. Balsa (11/97) Linux/Unix* port by Uwe F. Mayer (12/96,11/97)

TEST :	Iterations/sec.				New Index AMD K6/233*
NUMERIC SORT :	584.94	-:	15.00	:-:-	4.93
STRING SORT :	68.82	:	30.75	:	4.76
BITFIELD :	1.703e+08	:	29.21	:	6.10
FP EMULATION :	84.893	:	40.74	:	9.40
FOURIER :	5776.3	:	6.57	:	3.69
ASSIGNMENT :	11.11	:	42.28	:	10.97
IDEA :	2278.6	:	34.85	:	10.35
HUFFMAN :	1001.3	:	27.77	:	8.87
NEURAL NET :	8.4073	:	13.51	:	5.68
LU DECOMPOSITION :	341.61	:	17.70	:	12.78
	====ORIGINAL BYTE	MARK	RESULTS==		
INTEGER INDEX :	30.114				
FLOATING-POINT INDEX:	11.623				
Baseline (MSDOS \star) :	Pentium* 90, 256	KB	L2-cache,	Wat	com* compiler 10.0
	======LINUX DAT	A BE	LOW=====		
CPU :	4 CPU				
L2 Cache :					
os :	Linux 3.4.112-su	n8i			

: Linux 3.4.112-sun8i : gcc version 4.9.2 (Debian 4.9.2-10)

```
libc : libc-2.19.so
MEMORY INDEX : 6.829
INTEGER INDEX : 8.073
FLOATING-POINT INDEX: 6.446
Baseline (LINUX) : AMD K6/233*, 512 KB L2-cache, gcc 2.7.2.3, libc-5.4.38
* Trademarks are property of their respective holder.
```

6.1.3 Berechnung

 $Index = (MEMORY\ INDEX + INTEGER\ INDEX + FLOATING - POINT\ INDEX) \times 0.33$

6.1.4 Ergebnisse

	Takt	nBench
	MHz	Index
Orange Pi PC	1536	8,436
Raspberry Pi 3	1200	7,891
Orange Pi PC	1296	7,116
Orange Pi Lite/Zero	1200	6,594
Banana Pi	1000	5,552
Raspberry Pi 2 (Turbo)	1000	5,448
Orange Pi Lite/Zero	912	5,006
Raspberry Pi 2	900	4,904
Raspberry Pi B+ (Turbo)	1000	3,855
Raspberry Pi Zero	1000	3,820
Raspberry Pi B+	700	2,622



Abbildung 6.1: CPU Singlecore Performance (nBench)

Der Raspberry B+ und Zero setzen noch auf den SoC (System-on-a-Chip) BCM2835 von Broadcom. Dieser enthält einen alten ARM11 Prozessorkern. Dadurch sind die Messergebnisse vom Raspberry Pi B+ mit Turbo Einstellung identisch zum Raspberry Pi Zero. Der Raspberry Pi 2 verwendet den SoC BCM2836 von Broadcom. Darin enthalten sind vier ARM Cortex-A7 Prozessorkerne. Der Raspberry Pi 3 verwendet den SoC BCM2837 von Broadcom. Darin enthalten sind vier ARM Cortex-A53 Prozessorkerne.

Der Banana Pi und Banana Pro setzen auf den SoC A20 von Allwinner. Dieser enthält zwei ARM Cortex-A7 Prozessorkerne. Der Orange Pi enthält den SoC H3 von Allwinner. Beim Orange Pi PC wird dieser mit 1,3 GHz getaktet. Beim Orange Pi Lite wird der SoC nur mit 1,2 GHz getaktet weil hier ein anderer Spannungsregler zum Einsatz kommt. Wie bei dem Raspberry Pi 2 sind darin vier ARM Cortex-A7 Prozessorkerne enthalten.

Beim Orange Pi Zero wird der SoC H2+ von Allwinner eingesetzt. Dieser entspricht einem H3

Prozessor der um die Funktionen 4K HDMI Support und Gigabit Ethernet MAC reduziert wurde. Die vier ARM Cortex-A7 Prozessorkerne sind identisch und werden mit maximal 1,2 GHz getaktet.

Der Raspberry Pi 2 (Turbo) und Banana Pi/Pro haben beinahe die gleiche Single-Core-Leistung. Allerdings hat der Raspberry Pi 2 vier Kerne im Gegensatz zum Banana Pi und Banana Pro die nur 2 Kerne haben. Gegenüber dem Raspberry Pi 1 (Turbo-Modus) mit dem ARM11 Prozessor sind die neueren Cortex-A7 Kerne um rund 40 % schneller. Bei der CPU-Geschwindigkeit kann der Orange Pi mit seinem etwas höheren Takt punkten. Er übertrifft vor allem den Raspberry Pi B+ (Turbo) bzw. Zero um fast das doppelte. Sie besitzen auch nur einen CPU-Kern. Man könnte also von fast der 8-fachen CPU-Performance sprechen. Gegenüber dem Raspberry Pi 2 sind es immerhin auch noch gut 30 % mehr.

Der Raspberry Pi 3 ist mehr als 40 % schneller als der Raspberry Pi 2 (1000 MHz). Der Prozessorkern ARM Cortex-A53 ist also bei gleichem Takt ca. 20 % schneller als der Cortex-A7. Im Vergleich zum Orange Pi PC (1300 MHz) ist der Raspberry Pi 3 nur um 10 % schneller. Der Orange Pi Lite mit 1200 MHz ist um 8 % niedriger getaktet als der Orange Pi PC, um diesen Prozentsatz ist auch die CPU-Leistung reduziert. Orange Pi Lite und Orange Pi Zero haben die gleiche Taktrate und CPU-Leistung. Übertaktet man den H3 Prozessor des Orange Pi PC auf 1536 MHz so ist er noch schneller als der Raspberry Pi 3. Die Kühlung wird dabei aber zum Problem, deshalb sollte man 2 Kerne deaktivieren um eine ähnliche Leistungsaufnahme wie beim 4 Kernen mit normalen Takt zu erreichen. Sinn macht dies nur wenn wenn eine höhere Singlecore Geschwindigkeit benötigt, wie z. B. bei Spiel-Emulatoren oder anderen Programmen die nur einen Thread benutzen.

Wenn man den günstigen Preis des Orange Pi in Relation setzt, hat man ein optimales Preis-Leistungs-Verhältnis beim Orange Pi. Allerdings sind CPU-Performance und Preis unter Umständen nicht die einzigen Kriterien für ein Projekt.

6.2 Netzwerk/USB-HDD

Für die folgenden Benchmark Tests wird das Programm iperf für die Ermittlung der Netzwerk Übertragungsgeschwindigkeit verwendet. Es benötigt einen Server-Dienst, der auf einem Windows PC läuft und ein Client-Programm das am Einplatinencomputer gestartet wird.

Die Festplatten (HDD) Übertragungsrate wird mit dem Programm dd ermittelt, es ermöglicht blockorientiertes kopieren.

Der SMB-Server Samba ermöglicht eine Windows Dateifreigabe bzw. einen Windows File-Server unter Linux zu betreiben. Die Übertragungsgeschwindigkeit wird mit dem Windows Programm robocopy anhand einer Kopieroperation gemessen.

Bei den Tests mit dem Banana Pi wurde das Betriebssystem Bananian mit Kernel 3.4 verwendet.

6.2.1 Installation

apt-get install iperf samba

/boot/config.txt [-rwxr-xr-x root root]

max_usb_current=1

sudo reboot

/etc/samba/smb.conf [-rw-r--r-- root root]

```
[HDD]
   comment = HDD
   browseable = yes
   path = /mnt
  guest ok = yes
read only = no
   create mask = 0700
```

```
sudo -i
smbpasswd -a user
service smbd restart
service nmbd restart
mount /dev/sda1 /mnt
echo 480 > /sys/block/sda/device/max_sectors
mkdir /mnt/all
chmod 777 /mnt/all
```

6.2.2Ausführung

Server (Einplatinencomputer):

```
dd if=/dev/zero of=/mnt/speedtest-1gb-1 count=1024 bs=1M conv=fdatasync,notrunc
dd if=/dev/zero of=/mnt/speedtest-lgb-2 count=1024 bs=1M conv=fdatasync,notrunc
echo 3 | tee /proc/sys/vm/drop_caches
dd if=/mnt/speedtest-1gb-1 of=/dev/null bs=1M count=1024
dd if=/mnt/speedtest-1gb-2 of=/dev/null bs=1M count=1024
mkdir /mnt2
mount /dev/sdb1 /mnt2
echo 480 > /sys/block/sdb/device/max_sectors
time cp /mnt/speedtest-1gb-1 /mnt2
time cp /mnt/speedtest-1gb-2 /mnt2
echo 3 | tee /proc/sys/vm/drop_caches
time cp /mnt2/speedtest-1gb-1 /mnt
time cp /mnt2/speedtest-1gb-2 /mnt
iperf -c 192.168.0.106
Client (Windows PC):
robocopy \\192.168.0.110\hdd d:\temp speedtest-1gb-1 /tee
robocopy \\192.168.0.110\hdd d:\temp speedtest-1gb-2 /tee
robocopy d:\temp \\192.168.0.110\hdd\all speedtest-1gb-1 /tee /COPY:DT robocopy d:\temp \\192.168.0.110\hdd\all speedtest-1gb-2 /tee /COPY:DT
```

6.2.3Ergebnisse Netzwerk

	Takt	IPerf
	MHz	${f Mbits/sec}$
Banana Pi	1000	492
Orange Pi	1200	94,6
Raspberry Pi 3	1000	94,2
Raspberry Pi 2 (Turbo)	1000	94,2
Raspberry Pi 2	900	94,2
Raspberry Pi B+ (Turbo)	1000	91,7
Raspberry Pi B+	700	52,9



Abbildung 6.2: Netzwerk Übertragungsgeschwindigkeit

Der Banana Pi ist als einziger mit einem Gigabit-Ethernet-Netzwerkkontroller ausgestattet. Dies führt dazu, dass die Netzwerkperformance mehr als 5-mal so schnell ist, wie bei den Einplatinencomputern mit 100-MBit/s-Ethernet. Die restlichen Messwerte liegen nahe bei der maximalen Performance des 100-MBit/s-Ethernet Netzwerks. Nur der Raspberry Pi B+ ohne Turbo kann nur ca. 50~% der möglichen Netzwerkbandbreite nutzen.

6.2.4 Ergebnisse USB-HDD

Für diesen Test wurde eine externe 2,5 Zoll Festplatte von Typ Western Digital AV (WD500BUDT) 500 GB und ein LogiLink Gehäuse (UA0041A) verwendet. Beim USB-HDD-Test wird die maximale Lese- und Schreibgeschwindigkeit der Festplatte ermittelt. Beim Testen der USB-HDD zu USB-HDD Übertragungsgeschwindigkeit wurde zusätzlich eine weitere 3,5 Zoll USB-Festplatte mit externer Stromversorgung verwendet.

Beim SMB-Test wird die maximale Lese- und Schreibgeschwindigkeit des Samba File-Servers (SMB-Protokoll) ermittelt. Zugegriffen wird dabei auf die 2,5 Zoll Festplatte vom ersten Test.

USB-HDD:

	Takt	HDD 1.	HDD s.
	MHz	${f MB/s}$	${ m MB/s}$
Orange Pi	1200	37,6	36,5
Banana Pi	1000	35,5	34,3
Raspberry Pi 3	1200	35,3	34,3
Raspberry Pi 2 (Turbo)	1000	34,0	33,9
Raspberry Pi 2	900	33,9	33,4
Raspberry Pi B+ (Turbo)	1000	32,5	31,6
Raspberry Pi B+	700	31,6	29,2

Die USB-HDD Lese- und Schreibgeschwindigkeit liegt durchwegs bei über 30 MB/s. Die Orange Pi kann hier noch am meisten herausholen. Der Raspberry Pi B+ bildet das Schlusslicht und liegt ca. 15-20 % hinter der Orange Pi.

USB-HDD zu USB-HDD:



Abbildung 6.3: USB-HDD zu USB-HDD Übertragungsgeschwindigkeit

	Takt	HDD 1.	HDD s.
	MHz	${f MB/s}$	${f MB/s}$
Orange Pi	1200	29,8	23,9
Banana Pi	1000	25,1	22,7
Raspberry Pi 3	1200	17,3	16,3
Raspberry Pi 2 (Turbo)	1000	18,3	17,9
Raspberry Pi 2	900	18,1	17,9
Raspberry Pi B+ (Turbo)	1000	16,0	16,0
Raspberry Pi B+	700	11,9	14,0

Beim Übertragen von Daten von einer USB-Festplatte zur anderen, wird bei den Raspberry Pi Einplatinencomputern klar, dass nur ein USB-Kanal zur Verfügung steht. Die maximale Übertragungsrate liegt in der Nähe des halben Werts der Lese- und Schreiboperation allein. Bei den Banana und Orange Pi gibt es offenbar zwei getrennte USB-Kanäle. Dadurch können auch höher Übertragungsraten erzielt werden. Diese sind aber von der maximalen USB-Geschwindigkeit (ca. 34-37 MB/s) doch recht weit entfernt. Die Orange Pi bietet dabei eine etwas bessere Performance als die Banana Pi.

	Takt	SMB 1.	SMB s.
	MHz	${f MB/s}$	${f MB/s}$
Banana Pi	1000	24,9	31,6
Orange Pi	1200	11,3	11,2
Raspberry Pi 2 (Turbo)	1000	9,6	10,8
Raspberry Pi 3	1200	9,3	11,0
Raspberry Pi 2	900	9,3	10,6
Raspberry Pi B+ (Turbo)	1000	8,2	10,9
Raspberry Pi B+	700	5,4	8,1



Abbildung 6.4: SMB (Samba) Übertragungsgeschwindigkeit

Bei der realen Anwendung als File-Server sieht das Bild nicht mehr so drastisch aus wie beim Netzwerk Test, da die Daten von einer USB-Festplatte geladen bzw. geschrieben werden müssen. Hier ist der Banana Pi wieder unangefochten vorn und kann die Dateien mehr als 3-mal so schnell übertragen. Die Übertragungsrate von 31,6 MB/s beim Schreiben liegt nahe an der maximalen USB 2.0 Geschwindigkeit. Die Messwerte der anderen Einplatinencomputer liegen nahe bei der maximalen Performance des 100-MBit/s-Ethernet Netzwerks. Der Orange Pi kann hier das Maximum von ca. 11 MB/s ausreizen. Die Raspberry Pi 2 und 3 liegen etwas dahin und können beim Schreiben ca. 9,6 MB/s übertragen. Nur der Raspberry Pi B+ ohne Turbo fällt negativ auf. Er kann beim Lesen sogar nur 5,4 MB/s übertragen, das entspricht also ca. 50 % der möglichen Netzwerkbandbreite.

Für den Einsatz als File-Server bzw. NAS ist also der Banana Pi die beste Wahl. Er bietet sogar noch die Option eine 2,5 Zoll Festplatte direkt an einem SATA Anschluss zu betreiben. Eine

Konfiguration die im nächsten Test noch genau analysiert wird.

6.2.5 Ergebnisse SATA- und USB-SSD mit Banana Pi

Für diesen Test wurde eine SSD von Typ Kingston SSDNow V300 240 GB verwendet. Die SSD wurde einmal am USB-Anschluss mit einem LogiLink Gehäuse (UA0041A) und einmal am SATA-Anschluss direkt betrieben. Am Windows Client-System wurde ebenfalls eine SSD verwendet.

	SSD 1.	SSD s.	SMB 1.	SMB s.
	${f MB/s}$	${f MB/s}$	${f MB/s}$	${f MB/s}$
SATA	152,0	36,0	36,1	33,9
USB	36,5	36,6	25,0	32,4



Abbildung 6.5: SATA- und USB-SSD Übertragungsgeschwindigkeiten mit Banana Pi

Die Übertragungsraten der SATA-SSD können leider nur beim Lesen mit über $150\,\mathrm{MB/s}$ überzeugen. Beim Schreiben liegt die Übertragungsrate mit $36\,\mathrm{MB/s}$ gleichauf mit dem USB-Anschluss. Bei den Übertragungsgeschwindigkeiten beim File-Server Samba können nur beim Lesen nennenswerte Verbesserungen erzielt werden. Hier beträgt die Steigerung immerhin mehr als $10\,\mathrm{MB/s}$. Dadurch erreicht man also $36\,\mathrm{MB/s}$ beim Lesen und $33\,\mathrm{MB/s}$ beim Schreiben.

Insgesamt ist der Test aber etwas ernüchternd, wenn man bedenkt, dass theoretisch ca. $50~\mathrm{MB/s}$ möglich sein sollten. Ein guter PC kann sogar mehr als $100~\mathrm{MB/s}$ über das $1000~\mathrm{MBit/s}$ -Ethernet Netzwerk übertragen.

6.3 I/O (WiringPi C-Bibliothek)

Für manche Elektronik-Projekte ist nicht die Prozessorleistung von Bedeutung sondern die maximale Geschwindigkeit der GPIO's (General Purpose Input/Output). Bei diesem Test wurden die verschiedenen Einplatinencomputer bei I/O-Operationen mit der WiringPi C-Bibliothek verglichen.

Interessant ist dieser Test vor allem auch bei Wechsel der Hardware. Die verwendeten selbst entwickelten Programme können ohne signifikanter Anpassung (nur I2C-Bus Nummer) auf allen Einplatinencomputer kompiliert und gestartet werden.

Zu beachten ist allerdings, dass die WiringPi C-Bibliothek für Orange Pi und Banana Pi veraltet sind.

6.3.1 GPIO

Für den GPIO-Test wird ein Ausgang so schnell wie möglich zwischen High- und Low-Pegel umgeschaltet. Dies wird 100 Million Mal gemacht und daraus die Zykluszeit ermittelt.

./gpio_test

```
WiringPi GPIO speed test program toggle 100 million times ... 100000000 toggle took 7.347 s, Time per toggle 0.073 us, Freq 13.610 MHz
```

Ergebnisse

	Takt	Zykluszeit
	MHz	μs
Raspberry Pi 3	1200	0,073
Raspberry Pi 2 (Turbo)	1000	0,094
Raspberry Pi 2	900	0,104
Raspberry Pi B+ (Turbo)	1000	0,160
Raspberry Pi B+	700	0,230
Orange Pi PC	1200	0,771
Banana Pi	1000	1,001



Abbildung 6.6: GPIO Zykluszeit (weniger ist besser)

Bei diesem Test können die Raspberry Pi Varianten signifikant hervorstechen. Der Raspberry Pi 3 ist unangefochten der schnellste in diesem Test. Der Raspberry Pi 2 ist mehr als 7-mal so schnell wie der Orange Pi und fast 10-mal so schnell wie der Banana Pi. Im Vergleich zum Raspberry Pi B+ läuft er rund doppelt so schnell. Aber selbst der Raspberry Pi B+ ist um ein vielfaches schneller als der Orange und Banana Pi.

6.3.2 SPI

Für den SPI-Test wird ein Programm verwendet, dass einen MCP4922 ADC IC mit Sollwerten auf 2 Kanälen versorgt. Zusätzlich wird ein GPIO-Ausgang zur Steuerung des Ausgangs-Latch gesetzt. Der Test ist also eine Kombination aus SPI- und GPIO-Befehlen. Die Zykluszeit (Ausgabe von je einem Wert auf den zwei ADC-Kanälen) wird dabei ermittelt. Die Schaltung des GPIO-Ausgangs hat auf die ermittelte Zykluszeit keine nennenswerte Auswirkung. Die SPI-Frequenz wurde auf 6 MHz festgelegt. Sie ist in einem Bereich der keinen Einfluss mehr auf die Zykluszeit hat (gleiche Messwerte bei höheren SPI-Frequenz). Der Test kann auch ohne angeschlossener Hardware durchgeführt werden, da keine SPI-Leseoperationen enthalten sind.

Ausführung

./spi_test

```
WiringPi SPI speed test program open device '/dev/spidev0.0' (6 MHz) ...
Pong test:
  1000 frames took 2.827 s, dots per frame 150, Time per dot 19 us, fps 354 Space Invaders test:
  300 frames took 3.614 s, dots per frame 650, Time per dot 19 us, fps 83
```

Ergebnisse

	Takt	Zykluszeit
	MHz	μs
Raspberry Pi 3	1200	19
Raspberry Pi 2 (Turbo)	1000	26
Raspberry Pi 2	900	30
Raspberry Pi B+ (Turbo)	1000	61
Banana Pi	1000	85
Raspberry Pi B+	700	94
Orange Pi PC	1200	98



Abbildung 6.7: SPI Zykluszeit (weniger ist besser)

Ähnlich wie beim GPIO-Test sieht es auch beim SPI-Test aus. Den ersten Platz erreicht der Raspberry Pi 3, er ist 25 % schneller als Raspberry Pi 2. Der Raspberry Pi B+ (Turbo) erreicht ca. die halbe Leistung der Raspberry Pi 2. Der Banana Pi liegt sogar noch weiter dahinter aber noch vor dem Orange Pi.

Der Orange Pi erreicht trotz der stärkeren CPU nur rund ein Drittel der Geschwindigkeit der Raspberry Pi 2. Er ist damit das Schlusslicht des Tests.

Die Vergleichsweise schlechte Performance kann ein wesentlicher Negativpunkt bei zeitkritischen Projekten sein. Für die Vorgabe von XY-Signalen für ein Vektorbildschirm (Oszilloskop) ist die Geschwindigkeit maßgeblich für eine flimmerfreie Ausgabe. Von diesem Projekt stammt dieser

Benchmark (siehe Youtube Video "Vector Display Gaming with Raspberry Pi" - https://www.youtube.com/watch?v=hxSL9cxeJXo).

6.3.3 I2C

Der I2C-Test erfolgt mit einem Game Boy Advance-Dumper (GBA-Dumper). Dieser liest mithilfe von zwei MCP23017 ICs den ROM Speicher von einem GBA-Spiel aus. Zusätzlich wird ein GPIO-Ausgang zur Steuerung des RD-Eigangs des GBA-Moduls verwendet. Der Test ist also eine Kombination aus I2C- und GPIO-Befehlen. Die Zykluszeit (Auslesen eine Daten-WORDs) wird dabei ermittelt.

Ausführung

./gba

```
loding gba release list (139 kB)...done
Init GPIO interface...
        Set RD GPIO to default... open device '/dev/i2c-1'...
set slave 0x22 for IC1 (AD0-AD15) ... open device '/dev/i2c-1'...
set slave 0x21 for IC2 (A16-A23, Control) ...
write direction IC1 Port A/B (AD0-AD16) to output, default ...
write direction IC2 Port A (A16-A23) to output, default ...
write direction IC2 Port B (Control) to output, default
Write dump to file 'game.gba', using auto address mode, RD via GPIO!
-> 0000 KB (8 KB per dot).
-> Game : WINNIE (Save: WINNIE.gba)
-> GameCode : BWHP
-> Header: invalid
-> Header Complement (checksum): invalid (calc 0xE0, found 0x80)
-> Gamecode gbalist: AGB-BWHP-EUR
-> ROM Size: 32 MBit (4 MByte, max adress 0x200000)
-> Complement: 0x80
-> CRC32: 0xE5A97A58
-> RAM type:
-> RAM size: 0 kBit (0 Byte)
-> 0000 KB (8 KB per dot)......
-> Speed: dumping 64 kB took 8.1 sec. (246 microsec. per operation), full dump will take 8.3 min
-> 0128 KB (8 KB per dot).....
```

Ergebnisse

I2C-Bus Frequenz 100 kHz:

	Takt	Zykluszeit
	MHz	μs
Raspberry Pi 2	900	570
Raspberry Pi B+	700	606
Banana Pi	1000	747
Raspberry Pi 3	1200	856
Raspberry Pi 2 (Turbo)	1000	1076
Raspberry Pi B+ (Turbo)	1000	1105



Abbildung 6.8: I2C (100 kHz) Zykluszeit (weniger ist besser)

I2C-Bus Frequenz 400 kHz:

	Takt	Zykluszeit
	MHz	μs
Raspberry Pi 2	900	185
Raspberry Pi B+	700	222
Raspberry Pi 3	1200	248
Raspberry Pi 2 (Turbo)	1000	313
Raspberry Pi B+ (Turbo)	1000	350
Orange Pi PC	1200	391



Abbildung 6.9: I2C (400 kHz) Zykluszeit (weniger ist besser)

Bei dem Banana Pi läuft der I2C-Bus 2 nur mit 100 kHz. Damit ist die Performance um einiges geringer, als bei den anderen Systemen. Eine Möglichkeit wie der Bustakt angepasst werden kann, liegt nicht vor. Die Geschwindigkeit ist direkt im Kernel-Source hinterlegt (https://github.com/Bananian/linux-bananapi/blob/bananapi-3.4/arch/arm/plat-sunxi/include/plat/i2c.h, Anweisung #define I2C2_TRANSFER_SPEED (100000)).

Bei den Raspberry Pi Varianten kann der Bustakt in der Konfigurationsdatei angepasst werden. Dies ermöglicht einen Vergleich mit der reduzierten Frequenz. Hier liegt der Banana Pi hinter dem Raspberry Pi 2 und B+, aber noch vor dem Raspberry Pi 3. Bemerkenswert ist, dass die Übertaktung des Raspberry Pi einen negativen Einfluss auf die I2C-Geschwindigkeit hat. Dieser Effekt konnte auch bei der Taktrate von 400 kHz festgestellt werden.

Bei dem Orange Pi kann der Takt des I2C-Busses auch nicht parametriert werden. Er verwendet als Standardeinstellung allerdings 400 kHz. Trotz der stärkeren CPU ist der Orange Pi um einiges langsamer als die Raspberry Pi Systeme. Bemerkenswert ist allerdings wieder der negative Einfluss der Übertaktung bei den Raspberry Pi Einplatinencomputern.

Eine schlechte I2C-Bus Geschwindigkeit ist ein wesentlicher Negativpunkt bei zeitkritischen Projekten. Das Auslesen eines kleinen GBA-Moduls mit 4 MB dauert also auf dem Orange Pi fast 11 Minuten. Auf dem Raspberry Pi 2 ohne Turbo Einstellung ist dieser Vorgang in etwas mehr als 6 Minuten erledigt. Der Raspberry Pi 3 ist sonderbarerweise langsamer als die Raspberry Pi 2 und B+ (ohne Turbo) und würde ca. 8 Minuten brauchen. Die Banana Pi mit 100 kHz Bustakt braucht für den Vorgang sogar über 25 Minuten.

Anhang

Eigenwerbung

Vielleicht hat nun der eine oder andere Leser Lust bekommen mein E-Book Raspberry Pi - Projekte: Raspberry Pi als HTPC, Retro-Spielkonsole und für Elektronikprojekte nutzen zu lesen. Es kann über

Amazon http://www.amazon.de/kindle/dp/B00JGEEZOS oder

Google https://play.google.com/store/books/details?id=d6KAAwAAQBAJ bezogen werden.

Ich bedanke mich bei allen Käuferinnen und Käufern für die Unterstützung!



Danksagung

Besonders bedanken möchte ich mich bei Andi für die kostengünstige Erstellung der Fotos und bei Johanna für das Korrekturlesen.

Weiters danke ich Christof, Manfred und Andi für Tipps zum Buch.

Rechtliches

Haftungsausschluss

Die Benutzung dieses Buches und die Umsetzung der darin enthaltenen Informationen erfolgt ausdrücklich auf eigenes Risiko. Haftungsansprüche gegen den Verlag und den Autor für Schäden materieller oder ideeller Art, die durch die Nutzung oder Nichtnutzung der Informationen bzw. durch die Nutzung fehlerhafter und/oder unvollständiger Informationen verursacht wurden, sind grundsätzlich ausgeschlossen. Rechts- und Schadensersatzansprüche sind daher ausgeschlossen. Das Werk inklusive aller Inhalte wurde unter größter Sorgfalt erarbeitet. Der Verlag und der Autor übernimmt jedoch keine Gewähr für die Aktualität, Korrektheit, Vollständigkeit und Qualität der bereitgestellten Informationen. Druckfehler und Falschinformationen können nicht vollständig ausgeschlossen werden. Für die Inhalte, der in diesem Buch abgedruckten Internetseiten, sind ausschließlich die Betreiber der jeweiligen Internetseiten verantwortlich. Der Verlag und der Autor haben keinen Einfluss auf Gestaltung und Inhalte fremder Internetseiten. Verlag und Autor distanzieren sich daher von allen fremden Inhalten.

Lizenz



Dieses Werk steht unter der Lizenz Creative Commons BY-NC-ND 3.0

(https://creativecommons.org/licenses/by-nc-nd/3.0/at/). Sie erlaubt ausdrücklich, das Werk zu vervielfältigen, zu verbreiten und öffentlich zugänglich machen. Es ist nicht erlaubt diese Werk zu verändern und darauf aufbauen zu erweitern. Es muss der Urheber genannt werden, die Weitergabe darf nicht kommerziell erfolgen.

Dieses Werk ist eine reduzierte kostenfreie Version des kommerziell erhältlichen E-Book's des Autors Martin Strohmayer, es kann über

Amazon https://www.amazon.de/-/e/B071HJ6GYJ oder

Google https://play.google.com/store/books/author?id=Martin+Strohmayer bezogen werden.

Wenn Sie die Arbeit des Autors unterstützen wollen oder die orignale Version (qualitative bessere Bilder und vollständiger Umfang) besitzen wollen, so erwerben Sie das E-Book, Danke!

Impressum

Autor, Eigenverlag: Martin Strohmayer (freeebook@strohmayers.com)

Herstellungsort: Graz, Österreich