Tutotrial Raspberry pi B+ Setup:

Bei dem Kauf eines Raspberry Pi B+ besteht die Möglichkeit das Betriebssystem auf einer 8GB SD Card mit zubestellen. In diesem Fall ist das System einsatzbereit und der Raspberry Pi kann direkt von der mitgelieferten SD Card starten.

Die zweite Möglichkeit ist man bestellt sich nur die HW (Hardware) und ladet sich das Betriebssystem von www.raspberrypi.org herunter. Dies hat den Vorteil, dass verschiedene OS (Operating Systems) zur Verfügung stehen und je nach Anwendung ausgewählt werden können.

In diesem Tutorial wird die zweite Möglichkeit beschrieben und das OS per MAC auf die SD Card mit 32 GB geschrieben wird.

1. OS downloaden, wir verwenden 2014-09-09-wheezy-raspbian.img
2. Die SD Karte formatieren (FAT32) und Image (.img) auf die SD Card schreiben:

Mit dem Programm „Disk Utility“ (MAC OS) SD Card Partition neu formatieren und mit **UPPER CASE (Großbuchstaben!!)** für **FAT32** den Namen vergeben (wichtig UPPER CASE = FAT32).

3. Am MAC ein Terminal Fenster öffnen und das .img auf die SD Card schreiben.

sh-3.2# df-h

(folgendes erscheint man am Bildschirm:)

```
Filesystem     Size   Used  Avail Capacity  iused    ifree %iused  Mounted on
/dev/disk0s2   465Gi  133Gi  332Gi    29% 34900331 86986411   29%   /
devfs          185Ki  185Ki    0Bi   100%      640        0  100%   /dev
map -hosts       0Bi    0Bi    0Bi   100%        0        0  100%   /net
map auto_home    0Bi    0Bi    0Bi   100%        0        0  100%   /home
/dev/disk1s1    56Mi   12Mi   44Mi    22%      512        0  100%   /Volumes/bootFilesystem     Size   Used  Avail
Capacity  iused    ifree %iused  Mounted on
/dev/disk0s2   465Gi  133Gi  332Gi    29% 34900331 86986411   29%   /
devfs          185Ki  185Ki    0Bi   100%      640        0  100%   /dev
map -hosts       0Bi    0Bi    0Bi   100%        0        0  100%   /net
map auto_home    0Bi    0Bi    0Bi   100%        0        0  100%   /home
/dev/disk1s1    56Mi   12Mi   44Mi    22%      512        0  100%   /Volumes/boot
```

(folgendes Kommando eingeben:) sh-3.2# diskutil unmountDisk /dev/rdisk1

(folgendes erscheint am Bildschirm:) Unmount of all volumes on disk1 was successful

(folgendes Kommando eingeben:) sh-3.2# dd bs=1m if=2014-09-09-wheezy-raspbian.img of=/dev/rdisk1

(folgendes erscheint am Bildschirm:)
3125+0 records in
3125+0 records out
3276800000 bytes transferred in 312.989161 secs (10469372 bytes/sec)

(folgendes Kommando eingeben:)sh-3.2# diskutil unmountDisk /dev/rdisk1

11.01.15

(folgendes erscheint am Bildschirm:)
Unmount of all volumes on disk1 was successful

Jetzt kann man die SD-Karte herausnehmen.

.Die Schnittstellen des Raspberry Pi wie folgt verkabeln:

- **Schritt 1:** SD Card in den Slot schieben.
- **Schritt 2:** HDMI Kabel in die HDMI Buchse stecken und mit dem TV verbinden.
- **Schritt 3:** TV anschalten und auf den HDMI Port wechseln.
- **Schritt 4:** LAN-Kabel in die Ethernet RJ-45 stecken und mit dem Internet verbinden.
- **Schritt 5:** Sender zur Anbindung der Logitech K400r Tastatur an einen USB Port des Raspberry Pi stecken .
- **Schritt 6:** Stromversorgung an den Mini USB Port stecken, Raspberry Pi fährt hoch.

5. Falls sich der Raspberry Pi hinter einem Proxy-Server befindet, müssen dieser für ausgehende HTTP-, HTTPS- und möglicherweise auch für FTP-Verbindungen eintragen sein. Dazu erstellen man die Datei */etc/profile.d/proxy.sh* mit einem Editor (z.B. nano).
Alle Kommandos werden als SU (super user) eingegeben, wobei der TV als Monitor dient.

Login: pi
Password: raspberry

pi@raspberrypi ~ $ sudo su

root@raspberrypi:/home/pi# nano /etc/profile.d/proxy.sh

```
export http_proxy=http://192.168.10.1:3128
export https_proxy=http://192.168.10.1:3128
export ftp_proxy=http://192.168.10.1:3128

export HTTP_PROXY=$http_proxy
export HTTPS_PROXY=$https_proxy
export FTP_PROXY=$ftp_proxy
```

IP Adresse und Port sind nur ein Beispiel, bitte die eigenen Daten eingeben.
Aktuelle Terminal-Sitzung beenden und sich neu anmelden, dann werden die Einstellungen automatisch geladen.
Nun wird die eigene IP Adresse eingestellt, damit erhält der Raspberry Pi seine „feste“ IP Adresse.

root@raspberrypi:/home/pi# nano /etc/network/interfaces

```
auto eth0
iface eth0 inet static
    address 192.168.10.50
    netmask 255.255.255.0
    network 192.168.10.0
    broadcast 192.168.10.255
    gateway 192.168.10.1
    dns-nameservers 192.168.10.1 192.168.10.2
    dns-search home.lan
```

Nun werden wir den Service Networking neu starten und zwar mit der gewählten IP.

```
root@raspberrypi:/home/pi# service networking restart
```

6. Raspberry Pi über Config-Tool konfigurieren.

root@raspberrypi:/home/pi# raspi-config

- **Schritt 1: 1 Expand Filesystem** markieren und **Enter** drücken. Es sollte eine „erfolgreiche Meldung“ kommen die mit **Ok** bestätigt wir.
-
- **Schritt 2: 2 Change User Password** markieren und **Enter** drücken. Das sollte man generell immer nach dem Tastaturlayout machen, denn nutzt man z. B. Sonderzeichen sind die auf einer anderen Taste wenn man den englischen Standard hat. Daher aufpassen und erst nach dem Ändern des Tastaturlayout das PW ändern.

Die nächste Meldung mit **Ok** wieder bestätigen. Dann in der Konsole das neue Passwort eingeben, hier sollte ein starkes gewählt werden. Mit **Enter** bestätigen, nochmal eintippen und wieder mit **Enter** bestätigen, danach sollte eine „erfolgreiche Meldung“ kommen, die wieder mit **Ok bestätigt** wird.

- **Schritt 3: 3 Enable Boot to Desktop/Scratch** markieren und Enter drücken.

Da wir den Raspberry Pi ohne Monitor und Maus betreiben, muss der Raspberry Pi auch nicht mit einem Desktop starten. Daher hier **Console text console, requiring login (default)** auswählen und mit **Enter** bestätigen.

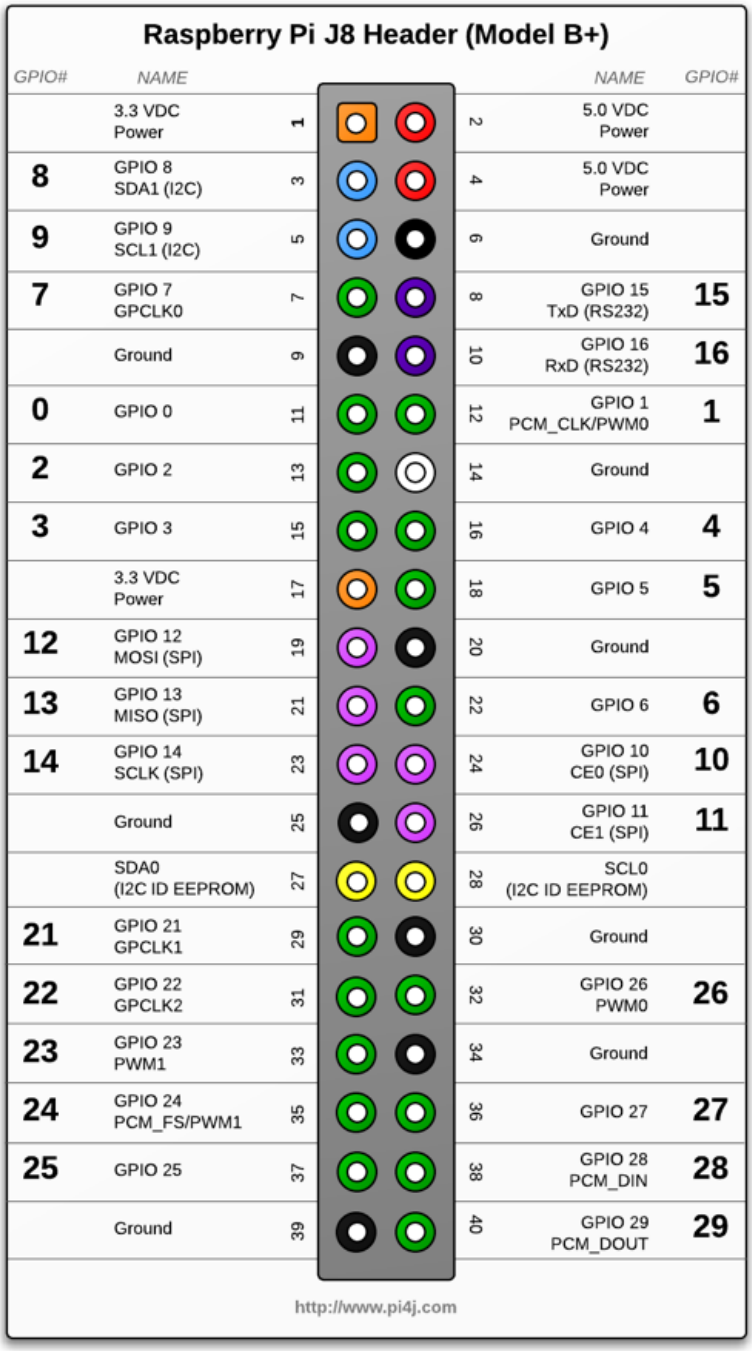
- **Schritt 4: 4 Internationalisation Options** markieren und mit **Enter** öffnen.
- **Schritt 4.1: I 1 Change Locale** auswählen und **Enter** drücken. Dort in der Liste **de_DE.UTF-8 UTF 8** mit Leertaste markieren. Jetzt **Enter** drücken und **de_DE.UTF-8** auswählen, nochmal **Enter** drücken.
- **Schritt 4.2: Wieder in Menüpunkt 4** und diesmal **I 2 Change Timezone** mit **Enter** auswählen. Jetzt **Europe** markieren und **Enter** auswählen, danach z. B. **Wien** mit **Enter** bestätigen.
- **Schritt 5: 7 Overclock** markieren und **Enter** drücken. Die Warnung mit **Enter** wegdrücken. Medium markieren und Enter drücken. Erfolgreich Meldung mit **Enter** wegdrücken. Das gibt dem Raspberry Pi etwas mehr Kraft, ohne ihn zu überlasten.
- **Schritt 6: 8 Advanced Options** markieren und mit **Enter** bestätigen. Jetzt **A3 Memory Split** auswählen und mit Enter öffnen. Dort **128** eintragen und mit **Enter** bestätigen.
- **Schritt 7: 8 Adcanced Options** markieren und mit **Enter** öffnen. Dort **A9 Update** markieren und **Enter** drücken.
- **Schritt 8: Finish** anwählen und mit **Enter** bestätigen. Danach im Terminal folgendes eintippen:

root@raspberrypi:/home/pi# reboot

Warten bis der Raspberry Pi neu gestartet wurde, danach neu anmelden mit **user** und neuem **Password**.
Danach sollte die Software aktualisiert werden:

```
root@raspberrypi:/home/pi# apt-get update
root@raspberrypi:/home/pi# apt-get upgrade
root@raspberrypi:/home/pi# halt –h
```

7. Verkabelung der LED um die Morsezeichen anzeigen zu lassen. Dazu benötigt man die Anschlußbelegung der GPIO Ports.



Die grüne LED U=2.1V wird über einen Widerstand an GPIO7 angeschlossen. Der Widerstand wird wie folgt berechnet:

R=U/I = 5V – 2,1V/10mA = 290Ω ≈ 270Ω

Anode**+**(langer Fuß) wird an GPIO7 und Kathode**-** (kurzer Fuß) wird mit GND9 verbunden.

8. Den Raspberry Pi neu gestartet und wieder User und Password eingeben.

Login: pi
Password: raspberry

pi@raspberrypi ~ $ sudo su

root@raspberrypi:/home/pi#

9. **Jetzt kommt die eigentliche Aufgabe, wir benutzen Python als Programmiersprache, welche bereits Bestandteil der Raspberry PI OS ist.**
   Wir brauchen aber noch die GPIO Library um die LED ansteuern zu können.

root@raspberrypi:/home/pi# apt-get install python-dev
root@raspberrypi:/home/pi# apt-get install python-rpi.gpio

10. Nun geht’s los mit dem Morsen:

root@raspberrypi:/home/pi# python morsen.py

Viel Spaß!

Anhang: Python Code „Morse Code LED“

```
import RPi.GPIO as GPIO
import time
   CODE = {' ': ' ',
         "'": '.----.',
         '(': '-.--.-',
         ')': '-.--.-',
         ',': '--..--',
         '-': '-....-',
         '.': '.-.-.-',
         '/': '-..-.',
         '0': '-----',
         '1': '.----',
         '2': '..---',
         '3': '...--',
         '4': '....-',
         '5': '.....',
         '6': '-....',
         '7': '--...',
         '8': '---..',
         '9': '----.',
         ':': '---...',
         ';': '-.-.-.',
         '?': '..--..',
         'A': '.-',
         'B': '-...',
         'C': '-.-.',
         'D': '-..',
         'E': '.',
         'F': '..-.',
         'G': '--.',
         'H': '....',
         'I': '..',
         'J': '.---',
         'K': '-.-',
         'L': '.-..',
         'M': '--',
         'N': '-.',
         'O': '---',
         'P': '.--.',
         'Q': '--.-',
         'R': '.-.',
         'S': '...',
         'T': '-',
         'U': '..-',
         'V': '...-',
         'W': '.--',
         'X': '-..-',
         'Y': '-.--',
         'Z': '--..',
         '_': '..--.-'}

 ledPin=7
 GPIO.setmode(GPIO.BCM)
 GPIO.setup(ledPin,GPIO.OUT)
   def dot():
        GPIO.output(ledPin,1)
        time.sleep(0.2)
        GPIO.output(ledPin,0)
        time.sleep(0.2)
  def dash():
        GPIO.output(ledPin,1)
        time.sleep(0.5)
        GPIO.output(ledPin,0)
        time.sleep(0.2)
  while True:
        input = raw_input('What would you like to send? ')
        for letter in input:
                      for symbol in CODE[letter.upper()]:
                             if symbol == '-':
                                    dash()
                             elif symbol == '.':
                                    dot()
                             else:
                                    time.sleep(0.5)
```

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