

# Das digitale Flugblatt – Installationsanleitung

## Benötigtes Technikzubehör

1. Raspberry Pi Zero WH (hat eine bereits angelötete GPIO-Stiftleiste; praktisch, falls das Gerät auch für andere Projekte bspw. mit Steckkabeln genutzt werden soll), oder Raspberry Pi Zero W (ohne angelötete GPIO-Stiftleiste), ca. 16,00 Euro
2. Powerbank mit microUSB-Kabel zum transportablen Betrieb der Raspberry Pis (z.B. 10000mAh), ca. 15,00 Euro
3. MicroSD-Karte mind. 8 GB, ca 4,00 Euro
4. (optional, empfiehlt sich aber, wenn man den Pi nicht völlig "blind" aufsetzen will) Micro-USB-Datenkabel, ca. 7,00 Euro
5. (optional) Mini-HDMI-zu-HDMI-Adapter für Anschluss Raspi an Monitor, 1,00 Euro
6. (optional) micro-B-USB-zu-2-Port-USB-A-female-Kabel für Anschluss Raspi an Tastatur und/oder Maus, 1,- bis 6,- Euro (je nach Port-Anzahl)
7. (optional) Gehäuse für den Raspberry Pi, ca. 6,00 Euro

Hinweis: Artikel 1. + 5.-7. gibt es bei Elektronikversandhäusern wie watterott, Pimoroni, buyzero, ThePiHut, ModMyPi oft in Kombi-Kits.

## Installationsanleitung

Die folgenden Schrittfolgen basieren auf den Empfehlungen folgender Artikel und beziehen sich zum einen auf Windows 10 und zum anderen auf MacOSX – insbesondere 1. bezieht sich auch auf Linux.

1. <https://www.heise.de/select/ct/2017/22/1508780300482172>
2. <http://www.circuitbasics.com/raspberry-pi-zero-ethernet-gadget/>
3. <https://www.raspberrypi.org/forums/viewtopic.php?t=171109>
4. <https://www.factoryforward.com/pi-zero-w-headless-setup-windows10-rndis-driver-issue-resolved>

### *erforderliche Software zum Installieren und Bespielen des Raspberry Pi*

- Raspbian (aktuelles Debian-Linux-Betriebssystem-Image für den Raspberry Pi)  
[https://downloads.raspberrypi.org/raspbian\\_latest](https://downloads.raspberrypi.org/raspbian_latest)
- Win32 Disk Imager Software (zum Aufspielen des Image auf die SD-Karte des Raspberry Pi)  
<https://sourceforge.net/projects/win32diskimager/>  
(auf dem Mac: Etcher <https://www.balena.io/etcher/> oder auf der Kommandozeile dd)
- für Windows 10: RNDIS-Treiber (damit Windows 10 den per USB-Kabel angeschlossenen Pi erkennt)  
<http://web1.moddevices.com/shared/mod-duo-rndis.zip>
- für Windows: Bonjour-Service von Apple (falls nicht schon iTunes installiert ist; ist auch für die Erkennung des Raspberry Pi am USB-Kabel nötig)  
[https://support.apple.com/kb/DL999?locale=en\\_US](https://support.apple.com/kb/DL999?locale=en_US)
- für Windows: SSH-Client PuTTY (zur Konfiguration des Raspberry Pi via Kommandozeile)  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- FileZilla (FTP-Client zum Aufspielen der Webseite auf den Raspberry Pi)  
<https://filezilla-project.org/download.php>
- Notepad++ (zum Editieren von Konfigurationsdateien vom Raspberry Pi  
(beim Mac geht zum Beispiel Textwrangler: <https://textwrangler.de.uptodown.com/mac>)  
<https://notepad-plus-plus.org/download/>
- (optional) VNCViewer (zum Bedienen des Pi via GUI-Desktop)  
<https://www.realvnc.com/de/connect/download/viewer/>

## Windows – Installationsschritte zum Aufsetzen des Raspberry Pis

1. Formatiere die SD-Karte über Datenträgerverwaltung von Windows.
2. Schreibe das heruntergeladene aktuelle Raspbian-Stretch-Image mit Win32 Disk Imager auf die Mikro-SD-Karte des Raspberry Pi.
3. Gehe nach dem Brennen mit dem Explorer in das Dateisystem der SD-Karte und öffne folgende Dateien im Editor notepad++: config.txt und cmdline.txt  
--> in der Datei config.txt füge diese Zeile als neue Zeile ans Ende der Datei:  
dtoverlay=dwc2
4. --> in der Datei cmdline.txt füge folgendes Kommando ein, das lediglich durch ein Leerzeichen getrennt hinter dem Kommando rootwait eingefügt wird:  
modules-load=dwc2,g\_ether  
Alle hiernach möglicherweise noch folgenden Kommandos folgen nach nur einem Leerzeichen hinter diesem neu eingefügten.
5. mit notepad++ die Datei wpa\_supplicant.conf mit korrekten SSID-/PWD-/KEY-Daten für ein funktionierendes WLAN auf dem Pi erstellen – Beispielinhalt ab nächster Zeile kopieren:  
ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev  
update\_config=1  
country=DE  
  
network={  
    ssid="the\_ssid\_of\_your\_wlan"  
    psk="the\_wpa2\_key\_of\_your\_wlan"  
    key\_mgmt=WPA-PSK  
}
6. SSH in Raspbian anschalten, indem man eine leere Datei mit dem Namen ssh und ohne Dateinamenerweiterung ebenfalls in den BOOT-Ordner des Dateisystems der SD-Karte kopiert.
7. SD-Karte auswerfen, in den Raspberry Pi stecken und das USB-Kabel zum Daten-USB-Port des Pis verbinden (das ist der innen liegende, nicht der äußere PWR-USB-Port)
8. Im besten Falle installiert sich der RNDIS-Treiber beim Anschluss des Raspberry Pi von selbst als USB Serial Device (im Gerätemanager überprüfbar). In Windows 10 installiert sich der RNDIS-Treiber als COM-Port. Um Ethernet over USB nutzen zu können, muss der RNDIS-Treiber per Hand installiert werden. BITTE HIER DIE LINKS IN DER ANLEITUNG <https://www.factoryforward.com/pi-zero-w-headless-setup-windows10-rndis-driver-issue-resolved> BEACHTEN.
9. Öffne Putty. In SSH connect schreibe raspberrypi.local. Klicke Open. Eine SSH-Verbindung sollte jetzt aufgebaut werden (beim ersten Mal muss noch der Key akzeptiert werden.)
10. Login mit Default-Username: pi  
Default-Passwort: raspberry
11. Eingabe der folgenden Befehle am Kommando-Prompt:  
pi@raspberrypi:~\$ sudo raspi-config  
--> Interfacing option, enable VNC  
--> change user password to something unique (turingbus)  
Nun kann der Raspberry Pi ggf. auch via VNCViewer und Desktop weiter konfiguriert/erreicht werden, wenn der vncserver gestartet wird.

## Abgekürzte Kommandozeilenbefehle für die weitere Netzwerkkonfiguration des Raspberry Pi als Hotspot aus dem Heise-Artikel:

```
pi@raspberrypi:~$ sudo apt-get install hostapd dnsmasq
```

### Webserver

```
pi@raspberrypi:~$ sudo apt-get install nginx
```

SITE liegt unter /var/www/html (am besten mit FileZilla befüllen)

### Konfigurationsdateien von heise laden/per Filezilla

```
pi@raspberrypi:~$ wget ftp://ftp.heise.de/pub/ct/listings/1722-144.zip
```

(stattdessen kann sie auch per FileZilla in den pi-Home-Ordner kopieren)

```

pi@raspberrypi:~$ unzip 1722-144.zip
pi@raspberrypi:~$ sudo cp 1722-144/default/* /etc/default
pi@raspberrypi:~$ sudo cp -a 1722-144/hostapd /etc
pi@raspberrypi:~$ sudo cp 1722-144/dnsmasq.conf /etc

pi@raspberrypi:~$ cd 1722-144
pi@raspberrypi:~$ sudo cp fakedns/*.py /usr/local/bin
pi@raspberrypi:~$ sudo cp fakedns/*.service /etc/systemd/system
pi@raspberrypi:~$ sudo systemctl enable fakedns.service

```

#### #Umleitungen bei HTTP-Errorcodes

```

pi@raspberrypi:~$ sudo pico /etc/nginx/sites-available/default
in diese Datei vor "Location" die Zeile einfügen:
error_page 404 =302 http://192.168.255.1/;
pi@raspberrypi:~$ service nginx restart

```

#### #use DNS-Masq als DNS-Umleiter

```

pi@raspberrypi:~$ sudo pico /etc/dnsmasq.conf
hier einfügen:
address=/#/192.168.255.1

```

#### Netzwerk

```

pi@raspberrypi:~$ sudo pico /etc/network/interfaces
hier einfügen:

```

```

auto wlan0
iface wlan0 inet static
address 192.168.255.1
netmask 255.255.255.0
wireless-mode Master
wireless-power off

```

```

auto usb0
iface usb0 inet static
address 192.168.3.11
netmask 255.255.255.0

```

--> Netzwerk-Interfaces neu starten (Achtung, immer Gefahr der Disconnection)

```

pi@raspberrypi:~$ ifup wlan0
pi@raspberrypi:~$ sudo /etc/init.d/networking restart
ODER
pi@raspberrypi:~$ sudo systemctl restart networking

```

#### Umbenennen des Hotspots (siehe Arbeitsblatt):

```

pi@raspberrypi:~$ sudo service hostapd restart

```

## MacOSX – Installationsschritte zum Aufsetzen des Raspberry Pis

1. Formatiere die SD-Karte über das Festplattendienstprogramm von MacOSX mit FAT32
2. Schreibe das heruntergeladene aktuelle Raspbian-Stretch-Image mit Etcher auf die Mikro-SD-Karte des Raspberry Pi.
3. Gehe danach mit dem Finder in das Dateisystem der SD-Karte und öffne folgende Dateien im Editor Textwrangler: config.txt und cmdline.txt  
--> in der Datei config.txt füge diese Zeile als neue Zeile ans Ende der Datei:  
dtoverlay=dwc2
4. --> in der Datei cmdline.txt füge folgendes Kommando ein, das lediglich durch ein Leerzeichen getrennt hinter dem Kommando rootwait eingefügt wird:  
modules-load=dwc2,g\_ether  
Alle hiernach möglicherweise noch folgenden Kommandos folgen nach nur einem Leerzeichen hinter diesem neu eingefügten.
5. mit Textwrangler die Datei wpa\_supplicant.conf mit korrekten SSID-/PWD-/KEY-Daten für ein funktionierendes WLAN auf dem Pi erstellen – Beispieldinhalt ab nächster Zeile kopieren:  
ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev  
update\_config=1  
country=DE  
  
network={  
    ssid="the\_ssid\_of\_your\_wlan"  
    psk="the\_wpa2\_key\_of\_your\_wlan"  
    key\_mgmt=WPA-PSK  
}
6. SSH in Raspbian anschalten, indem man eine leere Datei mit dem Namen ssh und ohne Dateinamenerweiterung ebenfalls in den BOOT-Ordner des Dateisystems der SD-Karte kopiert.
7. SD-Karte auswerfen, in den Raspberry Pi stecken und das USB-Kabel zum Daten-USB-Port des Pis verbinden (das ist der innen liegende, nicht der äußere PWR-USB-Port)
8. Öffne das Programm Terminal und verbinde dich via ssh mit dem Raspberry Pi. Hierzu schreibe folgendes auf die Kommandozeile:  
ssh pi@raspberrypi.local  
Beim ersten Mal wird noch abgefragt, ob du dich wirklich verbinden willst, weil der Key des neuen Hosts noch unbekannt ist. Hier yes eingeben.
9. Bei der Passwortabfrage  
pi@raspberrypi.local's password:  
ist das Default-Passwort: raspberry

Eingabe der folgenden Befehle am Kommando-Prompt:

```
pi@raspberrypi:~$ sudo raspi-config
```

--> Interfacing option, enable VNC

--> change user password to something unique (turingbus)

Nun kann der Raspberry Pi ggf. auch via VNCViewer und Desktop weiter konfiguriert/erreicht werden, wenn der vncserver gestartet wird.