

Inhalt

1	Programmierung Microcontroller ATtiny45.....	2
2	Konfiguration Touchscreen.....	5
2.1	microSD(HC) Karte formatieren.....	5
2.2	Konfiguration übertragen.....	5
2.3	Touchscreen in Betrieb nehmen.....	6
3	Software Installation	7
4	Anschlüsse der Platine	12
4.1	K1: Temperatursensoren.....	12
4.2	K2: Raspberry Pi GPIO	13
4.3	K3: LCD	13
4.4	K4: PWM Lüfter	13
4.5	K5: Servomotor	13
4.6	K6: Summer	14
4.7	K7: Taster	14
4.8	K8: Programmieranschluss für Microcontroller	14
4.9	K9: 12V Spannungsversorgung.....	14
4.10	K10: AddOn.....	14
4.11	K11: Lüfter 3-polig.....	14

1 Programmierung Microcontroller ATtiny45

Nachdem alle Bauteile auf die Platine gelötet wurden und die IC Sockel mit ihren jeweiligen ICs korrekt bestückt wurden, muss der ATtiny45 Microcontroller noch programmiert werden, denn dieser wird leer geliefert. Folgende Dinge werden dafür benötigt:

- Microcontroller Programmiergerät
- Verbindungskabel für Programmiergerät
- Programm: eXtreme Burner AVR
- Datei: Chips.xml http://wlanthermo.com/dl/ATtiny_V3/chips.xml
- Datei: V3_Attiny.hex http://wlanthermo.com/dl/ATtiny_V3/V3_Attiny.hex

Die Programmierung erfolgt mithilfe des Programmiergeräts, welches den ATtiny45 unterstützen muss. Ein günstiges Exemplar das diese Eigenschaften mitbringt, ist das USBasp Programmiergerät. Dieses ist für wenige Euro über diverse Verkaufsportale erwerbbar.

Um nun den ATtiny45 zu programmieren, muss die Platine des WLANThermos mit dem Programmiergerät verbunden werden. **Die Platine darf dabei nicht mit dem Raspberry Pi verbunden sein und an den Anschlüssen darf ebenfalls noch nichts angeschlossen sein.** Um die Platine zu verbinden wird die zweireihige Stiftleiste K8 verwendet. Abbildung 1 zeigt die entsprechende Belegung.

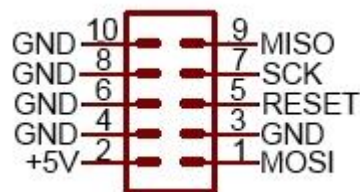


Abbildung 1 - Belegung K8

Wird das USBasp verwendet, so kann dieses einfach über das mitgelieferte Flachbandkabel mit der WLANThermo Platine verbunden werden. Hierbei ist darauf zu achten, dass der Stecker am Flachbandkabel so auf der Platine aufgesteckt wird, dass die in Abbildung 1 markierte Nase in Richtung des ATtiny45 (IC1) rechts von der Stiftleiste zeigt. Bevor das USBasp nun mit dem Computer verbunden wird, muss noch geprüft werden ob der Jumper auf dessen Platine auf 5V steht. Dieser muss ggf. erst umgesteckt werden. Danach dann der USBasp mit einem freien USB Port des Computers verbunden werden.

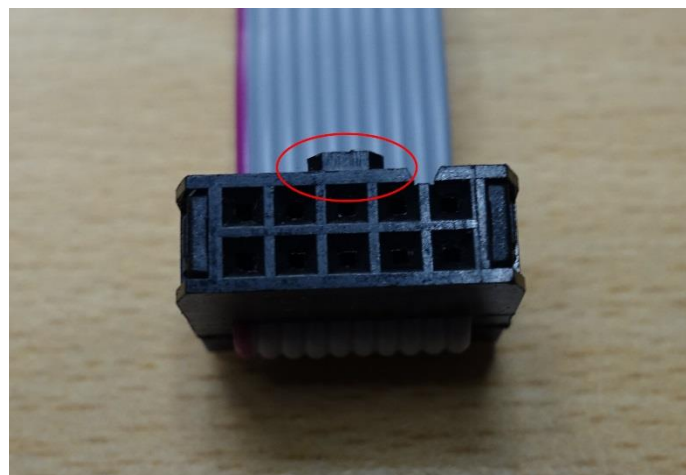


Abbildung 2 - ISP Stecker

Andere Programmiergeräte müssen entsprechend der in Abbildung 1 genannten Belegung eventuell anders angeschlossen werden.

Um nun mit der Programmierung zu beginnen wird das Programm eXtreme Burner AVR gestartet. Nun über die Schaltfläche „Open“ die zuvor heruntergeladen Datei „V3_Attiny.hex“ auswählen, oben in der ProgramMLEiste unter dem Punkt „Chip“ den „ATtiny45“ wählen, anschließend ebenfalls oben in der ProgramMLEiste unter dem Punkt „Settings“ den „ISP“ Mode auswählen und schließlich unter dem Reiter „Fuse Bits/Settings“ die Einstellung entsprechend Abbildung 3 vornehmen.

Abbildung 3 - eXtreme Burner AVR

Um die Programmierung nun zu starten oben auf die Schaltfläche „Write All“ mit dem roten Pfeil klicken. Es öffnet sich ein neues Fenster, welches die Programmierfortschritt anzeigt. Abbildung 4 zeigt wie dieses dann aussehen sollte nachdem die Programmierung beendet ist.

Abbildung 4 – Programmierung

2 Konfiguration Touchscreen

Sofern das WLANThermo mit einem Nextion 2,8" Touchscreen betrieben werden soll, muss dieser auf diesen vorher noch die Konfiguration geladen werden. Dafür braucht es lediglich eine microSD(HC) Karte mit maximal 8 GB Speicher sowie die Konfigurationsdatei.

2.1 microSD(HC) Karte formatieren

Die microSD(HC) Karte muss zunächst formatiert werden. Dazu wird die Speicherkarte mit dem Computer verbunden. Anschließend Computer (ehemals Arbeitsplatz) unter Windows öffnen, Rechtsklick auf die Speicherkarte dort und schließlich „Formatieren...“ anklicken. Dann öffnet sich ein Fenster wie in Abbildung 5 gezeigt.

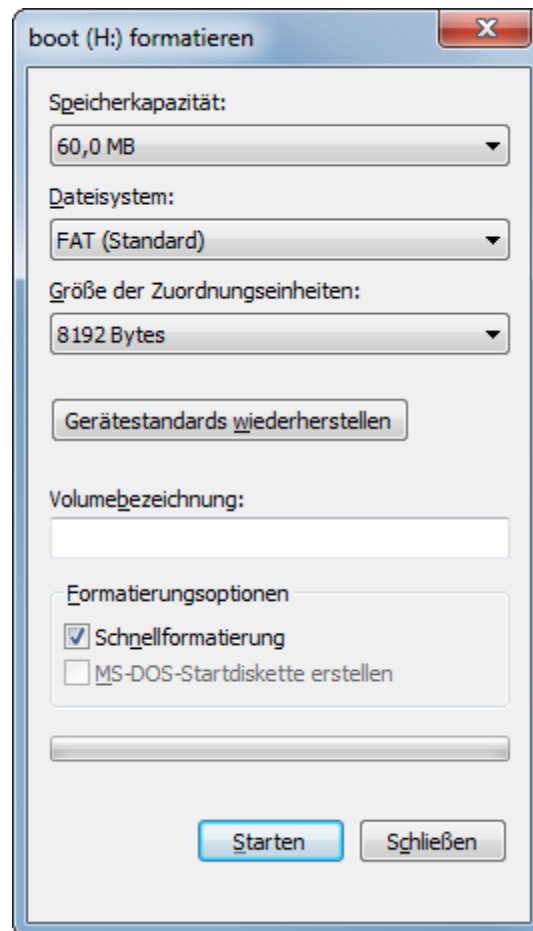


Abbildung 5 - microSD(HC) Karte formatieren

Hier ist wichtig, dass unter „Dateisystem“ die Auswahl auf „FAT“ steht. Danach auf „Starten“ klicken und die nun erscheinende Warnmeldung mit „OK“ bestätigen. Jetzt wird die Speicherkarte formatiert. Nachdem die Formatierung erfolgreich abgeschlossen wurde, kann mit dem nächsten Schritt fortgefahren werden. Für sämtliche andere Betriebssysteme sollten sich im Internet reichlich Anleitungen finden. Wichtig ist nur, dass als Dateisystem FAT(32) gewählt wird.

2.2 Konfiguration übertragen

Jetzt muss noch die Konfiguration auf die Speicherkarte übertragen werden. Dazu zunächst die aktuelle Datei unter <http://wlanthermo.com/index.php/bauanleitung/hardware/wlanthermo-mini> ganz unten über den Link „Nextion-Firmware“ herunterladen. Da diese in ein Archiv gepackt ist, muss es noch entpackt werden. Darin enthaltene Datei wird einfach auf die

Speicherkarte kopiert und diese kann nun vom Computer getrennt und in den microSD Slot auf der Platine des Touchscreens geschoben werden.

2.3 Touchscreen in Betrieb nehmen

Nachdem nun die Speicherkarte mit dem Touchscreen verbunden wurde, kann mit der Inbetriebnahme begonnen werden. Dazu zunächst das Touchscreen mit der WLANThermo Platine verbinden. Hierfür wird die zweireihe Stiftheiste K10 verwendet, siehe Abschnitt Anschlüsse der Platine. Die Leitungen des Touchscreens werden entsprechend dem Schema in Abbildung 6 mit der Platine des WLANThermos verbunden. Die Farbe der Pfeile steht dabei jeweils für die Aderfarbe.

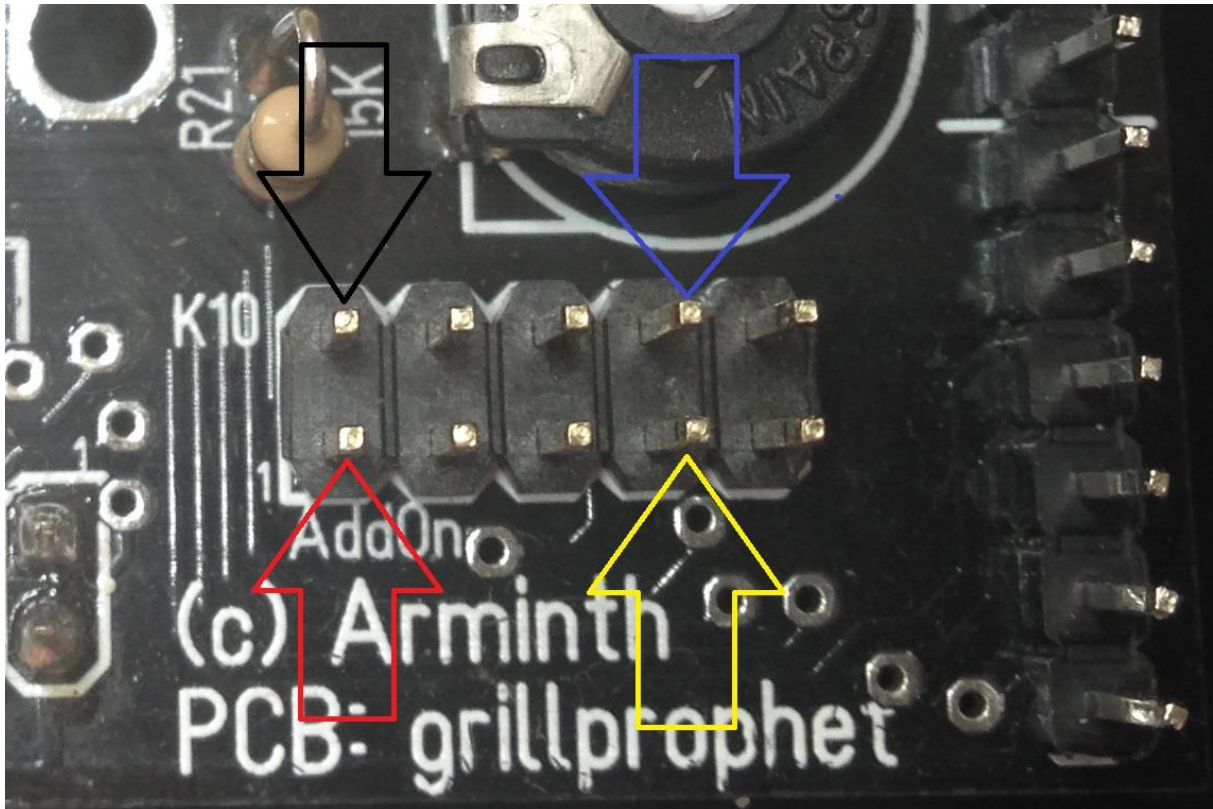


Abbildung 6 - Anschlussschema Touchscreen

Ist der Touchscreen nun angeschlossen, wird die (micro)SD(HC) Karte des Raspberry Pis entnommen und das WLANThermo mit Spannung versorgt. Nun sollte im Touchscreen automatisch der Kopiervorgang für die Konfiguration starten. Bei mir war dies nicht der Fall, daher habe ich die Spannungsversorgung getrennt, die microSD(HC) Karte aus dem Touchscreen genommen und nochmal mit dem Computer verbunden. Dort habe ich dann die Datei in NX3224T028_011.tft umbenannt. Sollte sie als Dateiname vorher nur „NX3224T028“ ohne die Endung .tft angezeigt werden, so ist die Dateiendung ausgeblendet. In diesem Fall dann die Datei in „NX3224T028_011“ umbenennen. Danach die Speicherkarte wieder mit dem Touchscreen verbinden und den WLANThermo mit Spannung versorgen. Nun sollte der Kopiervorgang starten. Wenn der Vorgang abgeschlossen ist, erscheint die Meldung „Update succeeded!“. Dann wird die Spannungsversorgung wieder vom WLANThermo getrennt, die (micro)SD(HC) Karte des Raspberry Pis wieder in dessen Speicherkarten Slot gesteckt und die Spannungsversorgung wieder verbunden. Nun muss nur noch in den Einstellungen des Webinterfaces unten ein Haken bei „LCD Anzeige:“ gesetzt werden und unter „LCD Type:“ der Eintrag „Nextion LCD“ gewählt werden. Danach unten auf „Speichern“ klicken und der Touchscreen verbindet sich. Dieser ist damit nun betriebsbereit.

3 Software Installation

Nachdem die Programmierung zuvor erfolgreich durchgeführt wurde, geht es nun an die Installation des Betriebssystems für den Raspberry Pi sowie der WLAN-Thermometer Software. Hierfür werden folgende Dinge benötigt:

- Programm: Win32 Disk Imager <https://sourceforge.net/projects/win32diskimager/>
- Programm: PuTTY <http://www.putty.org/>
- Raspbian Lite Image https://downloads.raspberrypi.org/raspbian_lite_latest

Zunächst müssen die genannten Dinge heruntergeladen und installiert bzw. entpackt werden. Ist dies erfolgt, kann die eigentliche Installation beginnen. Dazu wird zunächst Win32 Disk Imager geöffnet wie in Abbildung 7 gezeigt. Über das blaue Ordner Symbol wird nun das entpackte Raspbian Lite Image ausgewählt. Anschließend unter Device den Buchstaben der Speicherkarte wählen auf die das Image geschrieben werden soll. Achtung, das hier ausgewählte Gerät wird im Anschluss formiert. Es kann also zu erheblichem Datenverlust führen, wenn der falsche Buchstabe gewählt wurde!

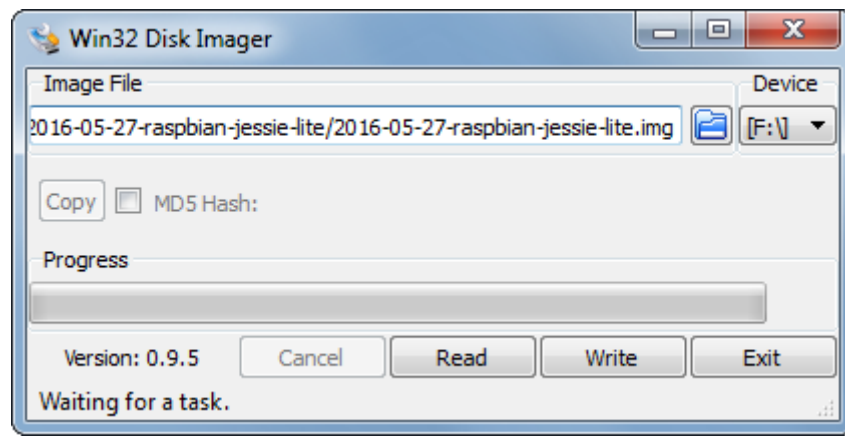


Abbildung 7 - Win32 Disk Imager

Danach auf „Write“ klicken und die Meldung mit Klick auf „Yes“ bestätigen. Nun wird das die Speicherkarte beschrieben. Der Vorgang dauert je nach Karte und Kartenleser einige Zeit. Auch einige der folgenden Aktionen werden nochmal ähnlich lange dauern und können unbeaufsichtigt laufen. Daher empfiehlt es sich die Wartezeit sinnvoll mit anderen Dingen zu nutzen.

Ist der Vorgang erfolgreich beendet, wird dies mit der Meldung „Write successful“ quittiert. Nun wird die Speicherkarte entnommen und in den Raspberry Pi gesteckt. Nun kann, sofern nicht bereits erfolgt, die WLANThermo Platine auf den Raspberry Pi gesteckt bzw. mit diesem verbunden werden. Dieser wird über ein Netzkabel mit dem Netzwerk verbunden und anschließend die WLANThermo Platine mit Strom versorgt. Der Raspberry Pi fährt nun hoch. Am Computer wird nun PuTTY gestartet um sich über das Netzwerk mit dem Raspberry Pi zu verbinden. Dazu wird im Feld „Host Name“ „raspberrypi“ eingetragen und anschließend auf „Open“ geklickt, Abbildung 8 zeigt das entsprechende Fenster.

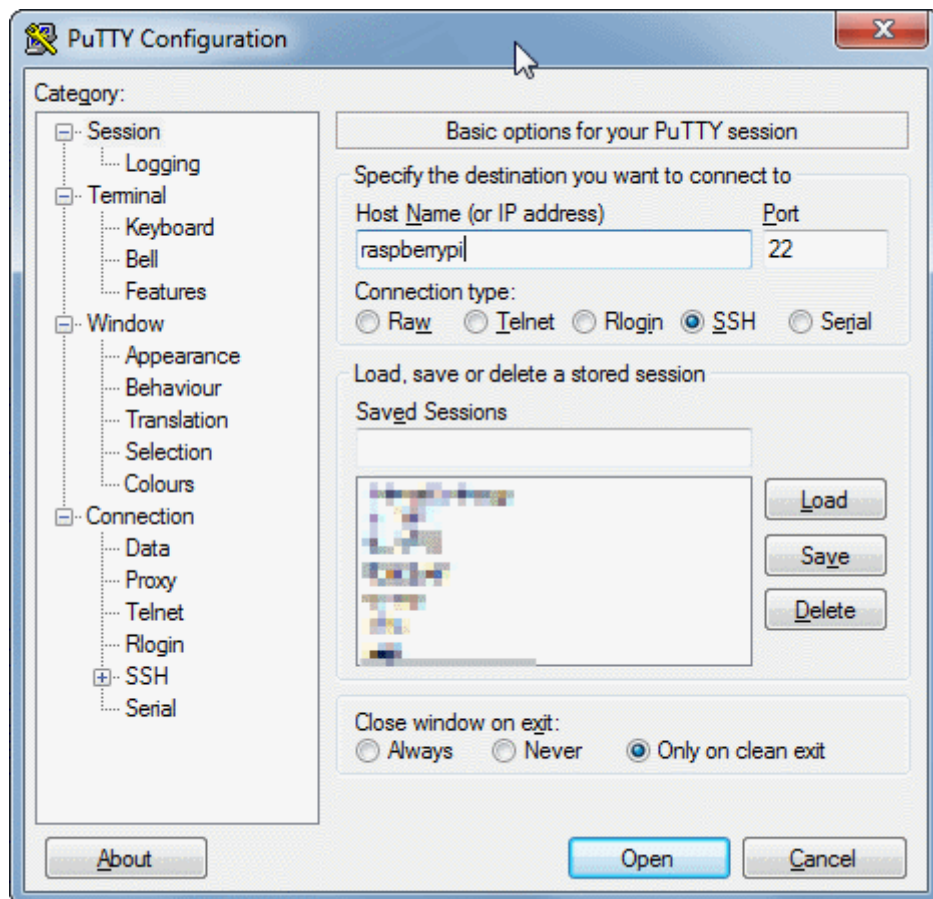


Abbildung 8 - PuTTY Konfiguration

Die nun erscheinende Meldung wird mit „Yes“ quittiert. Im Folgenden werden Befehle und Eingaben in jeweils eine eigene Zeile geschrieben damit diese besser kopiert werden können.

Als Login wird

pi

eingegeben. Das entsprechende Passwort lautet

raspberry

Dabei ist zu beachten, dass die Eingabe nicht durch erscheinen von Sternchen o.Ä. rückgemeldet wird. Ist der Login erfolgreich, sollte das Fenster wie in Abbildung 9 gezeigt aussehen.

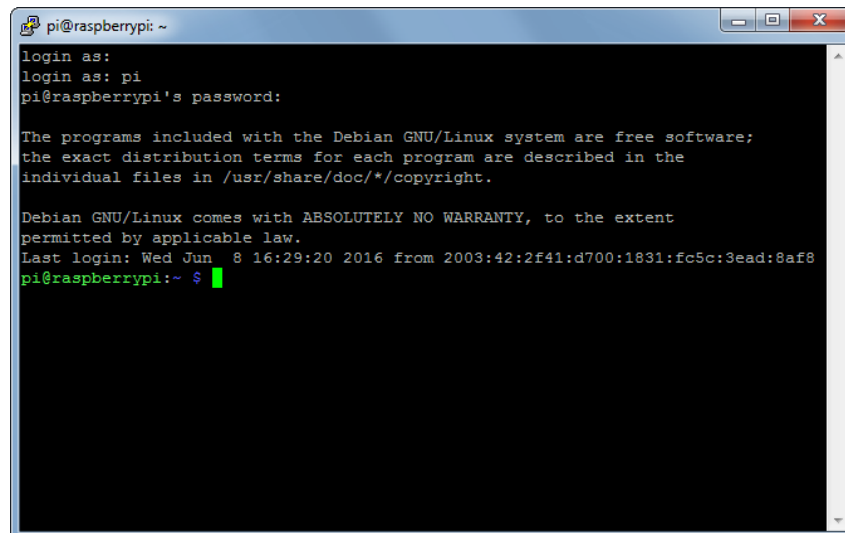


Abbildung 9 - PuTTY Login

Allgemein ist noch anzumerken, dass man unter Putty nicht den Nummernblock oder andere Sondertasten der Tastatur benutzen sollte. Die Eingaben sollten rein über den normalen „Hauptbereich“ der Tastatur erfolgen.

Nun beginnt die Konfiguration des Raspberry Pi. Dazu wird der Befehl

sudo raspi-config

einggegeben und mit Enter bestätigt. Im nun erscheinenden Menü wird über die Pfeiltasten nach oben/unten navigiert. So werden nun folgende Einstellungen vollzogen:

- **9 Advanced Options**
 - A0 Update this tool to the latest Version
- **1 Expand Filesystem** (falls dabei eine rote Zeile erscheint, einfach ignorieren)
- **2 Change User Password** (ändert das Passwort für den Login über Putty)
- **5 Internationalisation Options**
 - I2 Als Zeitzone zunächst „Europe“ und dann „Berlin“ auswählen
- Falls eine Raspi-Cam installiert ist: **6 Enable Camera enable**
- **9 Advanced Options**
 - A2 Hostname ändern auf z.B. „rpi-bbq“ unter diesem Namen ist der Raspberry Pi dann später im Netzwerk erreichbar
 - A3 Memory Split auf „8“ damit mehr RAM für die Ausführung der Dienste zur Verfügung steht. Dabei nicht den Nummernblock nutzen.
 - A7 Serial auf „No“ ändern
- **Finish** (Pfeiltasten nach links/rechts)

Nun wird der Raspberry Pi mit

sudo reboot

neugestartet. Dabei wird die Verbindung von Putty abbrechen. Entsprechend Putty schließen und erneut starten. Unter „Host Name“ kann diesmal „rpi-bbq“ eingegeben werden. Je nach Router kann es aber vorkommen, dass der Raspberry Pi noch nicht unter diesem neuen Hostnamen erreichbar ist. In diesem Fall dann stattdessen nochmal „raspberrypi“ eingeben. Nun erneut mit dem Benutzer „pi“ und dem neuen Passwort einloggen. Danach erneut mit folgendem Kommando Updates installieren

sudo aptitude update && sudo aptitude upgrade -y

Auch dieser Vorgang wird wieder einige Zeit in Anspruch nehmen. Danach ist die Konfiguration des Betriebssystems beendet und es geht weiter mit der Installation der WLAN Thermo Software. Dazu wird zunächst

cd /home

einggegeben. Danach wird das Softwarepaket mit dem Kommando

sudo wget http://www.wlanthermo.com/dl/WLANThermo_install.run

heruntergeladen und mit den Kommandos

sudo chmod +x WLANThermo_install.run

sowie

sudo ./WLANThermo_install.run

installiert. Auch dieser Vorgang erfordert wieder einige Geduld. Dabei wird man dazu aufgefordert das Passwort für die Benutzeroberfläche des WLAN Thermo festzulegen. Der entsprechende Username lautet „wlanthermo“. Diese Daten werden später im Webinterface benötigt. Das Webinterface sollte nun unter <http://raspberrypi> oder <http://rpi-bbq> erreichbar sein. Sollte ein anderer Hostname als rpi-bbq gewählt worden sein, so ist dies in dem vorigen Link entsprechend zu ersetzen.

Nun muss lediglich noch der USB WLAN Adapter konfiguriert werden. Sofern dieser noch nicht mit dem Raspberry Pi verbunden ist, zunächst mit

sudo shutdown -h now

herunterfahren, dann den Adapter verbinden und anschließend den Raspberry Pi wieder hochfahren. Nun wie gewohnt wieder per Putty verbinden und einloggen. Als nächstes wird mit dem Kommando

lsusb

die Liste der verbundenen USB Geräte angezeigt. Hier sollte der WLAN Adapter mit aufgelistet sein. Ist dies der Fall, kann seine Funktionalität über folgendes Kommando getestet werden.

iwconfig wlan0

Erscheint folgende Antwort, so arbeitet der Adapter korrekt:

```
wlan0      unassociated  Nickname:"<WIFI@REALTEK>"
           Mode:Managed  Frequency=2.412 GHz  Access Point: Not-Associated
           Sensitivity:0/0
           Retry:off      RTS thr:off   Fragment thr:off
           Power Management:off
           Link Quality:0  Signal level:0  Noise level:0
           Rx invalid nwid:0    Rx invalid crypt:0    Rx invalid frag:0
           Tx excessive retries:0      Invalid misc:0  Missed beacon:0
```

Nun kann mit diesem Kommando nach verfügbaren WLAN Netzen gesucht werden:

iwlist wlan0 scanning

Um den WLAN Schlüssel nun zu schreiben sind höhere Rechte erforderlich. Dazu folgendes Kommando eintippen:

sudo su -

Nun kann der WLAN Netzwerkname sowie dessen Schlüssel mit folgendem Kommando eingegeben werden:

wpa_passphrase ESSID PASSPHRASE >> /etc/wpa_supplicant/wpa_supplicant.conf

ESSID durch den Namen des WLAN Netzes und PASSPHRASE durch den zugehörigen Schlüssel ersetzen. Sollten Leerzeichen enthalten sein, so sind vor und nach dem Ersatz für ESSID und PASSPHRASE jeweils noch einfache Anführungszeichen zu setzen. Bsp. 'ESSID'. Danach werden durch Druck der Tasten Strg+D die höheren Rechte wieder abgegeben.

Mit dem Kommando

sudo cat /etc/wpa_supplicant/wpa_supplicant.conf

Kann nun geprüft werden ob die Angaben korrekt übernommen wurden. Anschließend wird mit den Kommandos

sudo ifdown wlan0

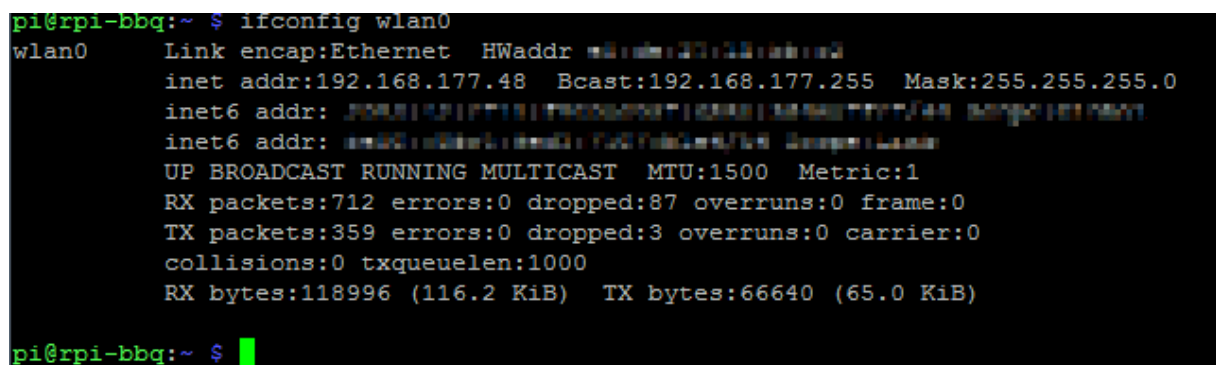
und

sudo ifup wlan0

der WLAN Adapter neugestartet.

Mit dem Kommando

kann nun noch die Konfiguration des WLAN Adapters angezeigt werden. Hier wird auch angezeigt unter welcher IP-Adresse der Raspberry Pi zu erreichen ist. Abbildung 10 zeigt wie die Antwort auf das Kommando aussehen kann.



```
pi@rpi-bbq:~$ ifconfig wlan0
wlan0      Link encap:Ethernet  HWaddr 88:7e:46:02:03:02
          inet addr:192.168.177.48  Bcast:192.168.177.255  Mask:255.255.255.0
          inet6 addr: fe80::887e:46ff:fe02:0302/64 Scope:Link
          inet6 addr: fe80::887e:46ff:fe02:0302/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:712 errors:0 dropped:87 overruns:0 frame:0
          TX packets:359 errors:0 dropped:3 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:118996 (116.2 KiB)  TX bytes:66640 (65.0 KiB)

pi@rpi-bbq:~$
```

Abbildung 10 - WLAN Konfiguration

Unter <http://IP-Adresse> (IP-Adresse entsprechend durch die Angabe unter „inet addr:“ ersetzen) sollte der Raspberry Pi nun definitiv erreichbar sein. Im Fall des Beispiels von Abbildung 10 wäre der Raspberry Pi nun unter <http://192.168.177.48> erreichbar.

4 Anschlüsse der Platine

Nachdem die Installation der Software nun erfolgreich erledigt wurde, geht es mit der Hardware weiter. Hier geht es nun um die Anschlüsse auf der Platine. Die fertig gelötete Platine des WLANThermo V3 ist in Abbildung 11 gezeigt. Die Farbe der Platine kann sich unterscheiden. Die im Folgenden gemachten Positionsangaben beziehen sich immer auf die Ausrichtung der Platine in der Abbildung.

Abbildung 11 - Platine

Die Bezeichnungen aller Anschlüsse sind auf der Platine aufgedruckt.

4.1 K1: Temperatursensoren

Die Temperatursensoren werden an der zweireihigen Stiftleiste K1, mittig auf der Platine, angeschlossen. Hierbei ist zu beachten, dass die **Messspannung für die Sensoren an der in Abbildung 11 unteren Stiftreihe anliegt. Die obere Reihe hat jeweils Masse (GND) Potential.** Die Buchsen für die Sensoren werden wie in Abbildung 12 gezeigt angeschlossen. Die dritte Lötfläche, welche in der Abbildung unbeschriftet ist, bleibt unbelegt. Zwischen die beiden belegten Lötflächen sollte ein 10 nF Kondensator gelötet werden um das Signal zu glätten.

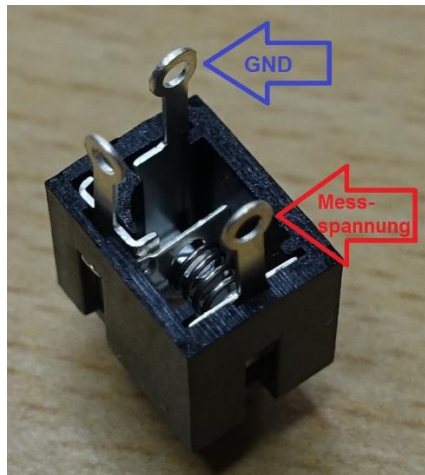


Abbildung 12 - Buchse Temperatursensor

4.2 K2: Raspberry Pi GPIO

Über die Buchsen- oder Stiftleiste (je nach Ausführung) K2 unten auf der Platine wird die Platine des WLANThermos mit den GPIO Pins des Raspberry Pis verbunden.

4.3 K3: LCD

Mithilfe der einreihigen Stiftleiste K3 am rechten Rand der Platine kann ein 4x20 LCD Display angeschlossen werden. Die Nummerierung der Pins ist von oben nach unten fortlaufend (1 bis 12). Der Kontrast des Displays kann mithilfe des schwarzen runden Potentiometers (R22) im unteren rechten Bereich der Platine eingestellt werden.

Das LCD wird nach folgendem Schema angeschlossen:

Pin an WLANThermo Platine	Anschlussbezeichnung am LCD
1	VSS
2	VDD
3	VEE oder V0
4	RS
5	RW
6	E
7	D4
8	D5
9	D6
10	D7
11	LED+ oder A
12	LED- oder K

Tabelle 1 - Anschlussschema LCD

4.4 K4: PWM Lüfter

Am 4-poligen Anschluss K4, welcher sich links oberhalb des Potentiometers (R22) befindet, kann ein sogenannter PWM Lüfter mit vierpoligem Stecker angeschlossen werden über dessen Luftstrom dann die Temperatur bei Holzkohlegrills reguliert werden kann, sofern dieser an deren Lüftungsöffnungen montiert ist.

4.5 K5: Servomotor

Über den dreipoligen Anschluss K3, rechts neben der Stiftleiste für die Temperatursensoren, kann ein Servomotor angeschlossen werden. Dieser kann dazu verwendet werden die

Lüftungsschieber eines Holzkohlegrills zu bedienen. **Belegung von oben nach unten: Masse, +5V, PWM.**

4.6 K6: Summer

Der Summer wird über die den Anschluss K6, links neben der Stiftleiste für die Temperatursensoren K1, verbunden. Oben Masse, unten Plus.

4.7 K7: Taster

Am Anschluss K7 oben links wird der Taster für die Bedienung des WLAN Thermos angeschlossen. Über ihn kann der Raspberry Pi oder die LCD Hintergrundbeleuchtung bspw. an- und ausgeschaltet werden. Als Taster ist ein sogenannter Schließer zu verwenden. Dabei sind die beiden Kontakte verbunden wenn der Taster betätigt wird und getrennt wenn er unbetätigt ist.

4.8 K8: Programmieranschluss für Microcontroller

Anschluss K8 oben rechts ist dafür da um den Microcontroller ATtiny45 zu programmieren. Das Vorgehen hierzu wurde im Abschnitt Programmierung Microcontroller ATtiny45 zuvor bereits beschrieben.

4.9 K9: 12 V Spannungsversorgung

Über die Klemme K9 unten links wird die Platine des WLAN Thermos und der Raspberry Pi mit Spannung versorgt. Hier ist eine 12 V Spannungsquelle anzuschließen. **Achtung, der Anschluss ist nicht verpolungssicher!** Also prüft die Polarität lieber zweimal bevor ihr Spannung anlegt. Die Polarität ist auch auf der Platine aufgedruckt. Links +, rechts -.

4.10 K10: AddOn

An der zweireihigen Stiftleiste K10 kann bspw. ein Nextion Touchscreen angeschlossen werden. Dessen Inbetriebnahme sowie das Anschlussschema sind unter dem Abschnitt Konfiguration Touchscreen nachzulesen.

4.11 K11: Lüfter 3-polig

Der Anschluss K11 ist dafür das WLANThermo mit einem 3-poligen Lüfter zu verwenden. Die Funktion ist identisch wie unter K4: PWM Lüfter beschrieben.